Optimizing Task Offloading and Resource Allocation in Vehicular Edge Computing Based on Heterogeneous Cellular Networks

Xinggang Fan, *Member, IEEE,* Wenting Gu, *Student Member, IEEE,* Changqing Long, *Student Member, IEEE,* Chaojie Gu, *Member, IEEE,* and Shibo He, *Senior Member, IEEE,*

***Abstract*—5G is a promising technology for improving the Quality of Service (QoS) in Internet of Vehicles (IoV) applications, including Vehicular Edge Computing (VEC). However, 5G networks have a limited communication range due to their radio-frequency properties, which can be a challenge in dynamic IoV environments. To address this issue, we propose a VEC architecture based on heterogeneous cellular networks, in which vehicles can select the appropriate communication network by classifying tasks according to their maximum tolerable latency. In order to further enhance the overall performance of the VEC system, we developed an efficient scheme that optimizes task offloading decisions and computation resource allocation in the proposed architecture. We analyze and formulate the optimization problem and use the linear relaxation improved branch-and-bound algorithm to solve it. Through extensive simulations, we demonstrate that the proposed scheme is superior to other solutions in computing latency, energy consumption, and failure rate. *Index terms*— Vehicular Edge Computing (VEC), task offloading, computation resource allocation, task classification.**

I. Introduction

With the continuous increase of intelligent vehicles, Vehicular Edge Computing (VEC) is a promising technology supporting the Internet of Vehicles (IoV) applications in Intelligent Transport Systems (ITS) [1], such as navigation, traffic management, safety, and in-car entertainment. VEC is a type of edge computing that involves the utilization of computing resources and sensors in vehicles to process and analyze data, perform various tasks, and provide services to users. By bringing computing resources closer to terminal vehicles, VEC enables low-latency access to services while fulfilling the execution requirements of various service types [2]–[4].

Typically, VEC adopts wireless communication due to the fact that the vehicle environment is highly dynamic. The existing IoV communication mainly adopts Dedicated Short Range Communications (DSRC) method, which can realize vehicle identification, electronic deduction, and establishment of unattended vehicle channels [5]. As the IoV continues to evolve, the shortcomings of DSRC have become increasingly apparent, such as short transmission distance, signal blockage, and repeated construction of signal transmitters. Aiming at the limitations of DSRS, Cellular Vehicle to Everything (C-V2X) has been introduced to provide support for the complex services of the existing LTE-V2X and the developing NR-V2X [6]. Compared to DSRC, LTE-V2X technology is more advanced except for the latency. As the evolution of LTE, the fifth generation (5G) communication technology not only has higher bandwidth, supports a larger number of connections, but also supports higher mobile speed [7].

However, there are two major network problems if VEC is completely dependent on 5G services. The first is the availability of 5G services. In the current stage, 5G infrastructures have not been widely deployed due to construction costs and the availability of suitable deployment sites. 5G networks may not yet have widespread coverage, especially in rural or remote areas [8]. The second is the communication range of 5G networks. Compared to 4G networks, 5G networks have a shorter communication range because they operate at higher frequencies, which have shorter wavelengths. In other words, 5G signals can be more easily absorbed or blocked by obstacles on their propagation path. To this end, in this paper, we proposed to incorporate 5G base stations (gNBs) with the widely available 4G base stations (eNBs) to support VEC communication. In this way, the vehicle can actively select the communication network to improve its connectivity with the edge server.

In 5G communications, the Third Generation Partnership Project (3GPP) standardization proposes two network architectures, standalone (SA) and non-standalone (NSA). SA refers to the construction of a new 5G network, including new base stations, backhaul links, and a core network, which does not depend on the existing 4G network. NSA networking refers to the deployment of 5G networks using existing 4G infrastructure. 5G carriers based on the NSA architecture carry only user data, and their control signaling is still transmitted over the 4G network [9]. In the early stages of 5G deployment, 5G cells will not be widely covered and there will be 5G coverage gaps. Operators can seamlessly serve 5G users by interoperating with existing LTE networks. 5G interoperability with fully deployed LTE networks not only provides fast, seamless coverage, but also brings economic benefits to network operators [10]. Therefore, NSA is a more common network architecture today. However, the task offloading problem

X. Fan and W. Gu are with College of Computer Science and Technology, Zhejiang University of Technology, Hangzhou, Zhejiang, 310023, China. X. Fan and W. Gu are also with Zhijiang College, Zhejiang University of Technology, Shaoxing, Zhejiang, 312030, China. E-mail:{2112112060, xgfan}@zjut.edu.cn. C. Long, C. Gu, and S. He are with the College of Control Science and Engineering, Zhejiang University, Hangzhou, Zhejiang, 310027, China. E-mail: {lcqzju, gucj, s18he}@zju.edu.cn. *Corresponding author: Chaojie Gu*.

This research is supported by the National Natural Science Foundation of China under Grant U1909207, 62302439 and U23A20296.

in IoV has been extensively studied in SA networks [11], and the VEC problem in NSA networks has rarely been considered. The joint offloading of heterogeneous cellular networks based on the NSA network architecture proposed in this paper not only improves the computing efficiency of VEC, but also better meets the practical offloading scenarios in the current vehicular networking.

While utilizing the VEC architecture based on the heterogeneous cellular network to deal with the aforementioned network problems, task offloading and resource allocation should also be considered to improve computational efficiency and reduce energy consumption. Researchers have devoted efforts to designing various task offloading and resource allocation approaches. For more efficient utilization of edge resources, [12] introduced a 5G-enabled EC-IoV system architecture to improve the efficiency of the current EC-IoV system, and additionally provided a task offloading calculation method that is applicable under diverse scenarios. [13] designed an autonomous vehicular edge (AVE) system, which can effectively manage the idle computational resources of vehicles and leverage them to offer computing services in dynamic vehicular situations. However, existing approaches cannot be directly applied to the proposed VEC architecture because they do not consider the heterogeneity in base stations. Thus, we present a task offloading and resource allocation scheme for VEC based on heterogeneous cellular networks. In this scheme, tasks are classified according to their maximum tolerable delay and the communication range of the base stations. Based on this classification, the most suitable offloading method is selected. For the purpose of minimizing global task completion latency and energy consumption, we propose the use of a branch-and-bound algorithm to determine whether tasks should be executed locally or at the edge server. Our proposed scheme not only lowers the failure rate of task offloading in 5G-enabled edge computing but also minimizes the overall latency and energy consumption through weighted summation.

The contributions of this work can be summarized as follows:

- We consider the maximum tolerance delay of tasks and propose the VEC architecture based on heterogeneous cellular networks aimed at minimizing the weighted sum of total latency and energy consumption.
- We jointly formulate task offloading strategy and resource allocation as a mixed integer nonlinear programming (MINLP) problem. To address this issue, first, we introduce the task classification algorithm to determine the offloading method, and then we use the linear relaxation improved branch-and-bound algorithm to find the desirable solution.
- We simulate realistic vehicle tasks offloading with MATLAB to examine the effectiveness of our proposed scheme under various key parameters. In comparison with other offloading schemes, the proposed system exhibits lower latency, energy consumption, and failure rate, indicating superior performance.

Paper Organization. Section II reviews related work. Section III presents the proposed system model and problem formulation. Section IV proposes the linear relaxation improved branch-and-bound algorithm to resolve the formulated problem. Section V presents the evaluation results and Section VI concludes this paper.

II. Related Work

In recent years, the increased number of vehicle applications, which require high computation and consume a large amount of energy, has made VEC a research hotspot [14]. Numerous researchers have made significant contributions towards reducing the execution delay of tasks [15]–[18], as well as the energy consumption of task offloading [19]–[21]. Furthermore, several studies [11], [22], [23] have taken into account both user experience quality and limited resources, suggesting that leveraging the idle resources of communication vehicles can effectively address the task load on VEC servers.

Choo *et al*. [15] proposed an architecture called software-defined vehicles edge computing (SD-VEC), which allows the controller to manage both task offloading and resource allocation among edge servers simultaneously. They formulated a problem related to the selection of an edge server and allocation of resources with the goal of maximizing the probability of successfully completing tasks within specified time constraints. Zhu *et al*. [16] proposed a novel solution that considers the limitation of fog capacity, service latency, and quality loss to optimize resource allocation in Vehicular Fog Computing (VFC). Qiao *et al*. [17] designed the task migration computation offloading (TMCO) algorithm for VEC considering vehicle mobility and the strict delay deadline. The TMCO algorithm can dynamically select the appropriate edge server for offloading according to the moving route of the vehicles. Zhang *et al*. [18] proposed a MEC-enabled IoV architecture that allows both MEC servers and vehicles to act as offloading nodes, and jointly roadside units to enable the provision of low-latency offloading services. Additionally, they presented a task offloading strategy named TO-TCONS, which considers the selection of offloading nodes and task classification to minimize overall completion delay. These studies mainly take into account the high reliability and low latency of the task but ignore the energy efficiency problem. In fact, energy consumption is also a crucial factor to consider in the task offloading process.

Because of the mobility of vehicles, the communication environment is constantly changing. Jang *et al*. [19] jointly optimized the bit allocation and offloading percentage to reduce the total energy consumption of vehicles under the delay constraint. Li *et al*. [20] proposed the JTORAEH algorithm combining MEC and energy harvesting to decrease overall energy consumption while meeting the task latency requirement. Lu *et al*. [21] considered a large-scale MEC network that included multiple MEC servers and users. The overall strategy of joint task computing delay and energy efficiency is proposed to maximize the average user offloading utility.

However, the above work is mainly to enhance the user experience quality and cannot solve the problem of limited resources. Zeng *et al*. [22] conducted a study on the efficient

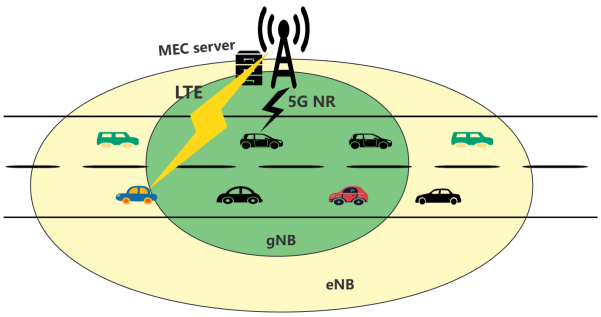


Fig. 1: The dual base station collaborative IoV architecture.

utilization of volunteer vehicle resources to manage the workload on VEC servers. Similarly, Peng *et al*. [23] introduced a novel paradigm known as parked-vehicle assisted edge computing (PVEC), which leverages idle computational resources of parked vehicles as a supplement for VEC. Utilizing the Stackelberg game, this study analyzes the mutual communication between VEC servers and requesting vehicles to identify the most effective offloading strategies. In addition to the availability of vehicle idle resources, gNBs can also provide task offloading services. Based on the cellular network, Raza *et al*. [11] further take into account the fifth-generation new-radio vehicle-to-everything communication model to enhance the overall system performance. However, the authors only analyzed resource allocation and did not consider the cooperation of base stations.

This paper proposes a VEC architecture based on heterogeneous cellular networks that differ from prior research works by offering more computing resources and decreasing the failure rate of low-latency tasks via base station cooperation. We jointly optimized task offloading and resource allocation in order to achieve optimal computational efficiency while minimizing total completion delays and energy consumption.

In addition to the studies addressing latency, energy consumption, and resource allocation challenges in task offloading, there has been extensive research on the selection of base stations when multiple base stations across different regions collaboratively provide wireless communication and task offloading services for vehicles. These studies can be broadly summarized into three categories. The first is based on the signal strength of the base station to select the appropriate offloading node for the task [24]–[26]. The second is based on load balancing between base stations to develop a global task offloading scheme [27]–[29]. The last is based on the total delay or total energy consumption of task offloading to improve the user experience quality [30], [31]. However, they did not consider the situation of collaborative communication assistance for task offloading using heterogeneous cellular networks within the same region. Differently, the proposed collaborative task offloading of heterogeneous cellular networks within the same region in this paper can further expand the application scope and scenarios of vehicle edge computing.

TABLE I: Notations

Symbol	Description
N	Vehicle set
P^{bs}	Position of the eNB-gNB dual base station
P_i^{veh}	Position of the vehicle i
g_i	Distance between vehicle i and base station j
Φ_i	Task Φ_i
d_i	Size of data to be offloaded
c_i	Computing resources required for the task Φ_i
T_i	Delay constraint of Φ_i
μ_i	Task offloading decision
η_i	Task offloading node selection
$R_{i,j}$	Uplink transmission rate
D_i^l	Local computing time
E_i^l	Local energy consumption
$D_{i,j}^t$	Task transmission time
$D_{i,j}^c$	Edge computing time
$D_{i,j}$	Total edge processing time
$E_{i,j}$	Vehicle energy consumption when offloading task
$B_{i,j}$	Uplink channel bandwidth
P_i^T	Transmission power when vehicle i is busy
P_i^I	Power consumption when vehicle i is idle
α	Path loss exponent
N_0	Noise power spectral density
h_0	Channel attenuation coefficient

III. System Model and Problem Formulation

In this section, we will introduce the VEC architecture based on heterogeneous cellular networks and discuss the task classification method, followed by the presentation of the system model. We will then provide a detailed discussion of the problem description and optimization function. Relevant symbols used in this section are listed in Tab. I.

A. VEC Architecture

We consider a VEC architecture based on heterogeneous cellular networks, where the vehicles that travel on the city road can receive LTE and 5G NR services. The offloading process of the task can be completed through the cooperation of eNB and gNB. Since the scenario modeled in this paper is aimed at dense urban areas, there is no significant difference in the density of eNBs and gNBs. To better describe the process of eNB and gNB collaborative task offloading, this paper takes the example of a dual base station model which is a co-sited base station with edge computing servers and vehicles [32], as shown in Fig. 1. In a practical environment, we can divide multiple eNBs and gNBs on the road into multiple dual base station models for collaborative task offloading [18]. The position of the base station is denoted as $P^{bs} = (x^{bs}, y^{bs})$. Let N ={1, 2,..., N} denote N vehicles in this system. The position of vehicle i is indicated by $P_i^{veh} = (x_i^{veh}, y_i^{veh}), i = 1, ...N$.

B. Task Classification

We make the assumption that there are N vehicles on the road with each vehicle i responsible for performing a periodic

computation-intensive task, which is modeled as a ternary $\Phi_i = \{c_i, d_i, T_i\}$, where c_i denotes the total number of CPU cycles necessary to complete the task Φ_i, d_i denotes the size of the input data required to process, T_i denotes the delay constraint of Φ_i. We use $\mu = \{\mu_1, \mu_2, ..., \mu_n\}$ to represent the task offloading selection. Note that when μ_i equals 0, it means that the task Φ_i will be executed locally, otherwise, it indicates that task Φ_i should be performed at the edge server. In addition, we also need to determine the offloading node of task Φ_i, therefore, we introduce a variable $\eta_i \in \{0, 1\}$. If η_i equals 0, it means that the vehicle i is in the coverage of eNB when it sends the request, so the task needs to be offloaded to eNB. Otherwise, the vehicle i will transmit the task to gNB.

Assume that the position where the vehicle sends the request is x_o, and the destination position to receive responses is denoted as x_d. According to the delay constraint and the coverage of the base station, the offloading modes of tasks can be further divided into two types.

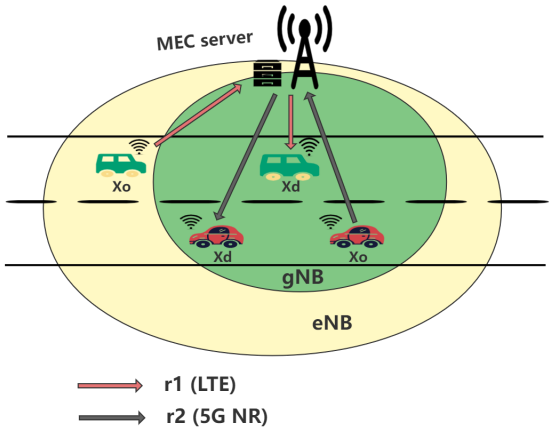


Fig. 2: Single base station offloading mode.

1) Single base station offloading mode: The first offloading mode pertains to a scenario where both the original and destination vehicles are located within the same coverage area of a gNB or eNB. As illustrated in Fig. 2, there are two methods for carrying out the offloading task. The first method, labeled as $r1$, involves x_o and x_d being situated within the same eNB's coverage zone. In this case, the vehicle offloads the task to the eNB for execution. After the eNB finishes executing the task, the result is returned from the eNB to the vehicle. Similarly, the other way, labeled as $r2$, is that both the original and destination vehicles are covered by gNB. In this case, the vehicle can directly send the task to the gNB, and the gNB will execute it and return the result. Under the first offloading mode, the offloading process of these tasks consists of three parts: task uplink transmission, edge computation, and downlink transmission.

2) Dual base station offloading mode: Unlike the first offloading mode, the second mode involves the original and destination vehicles that are not within the same base station coverage area, as depicted in Fig. 3. In particular, if the vehicle sends a task processing request to the eNB, we do not consider the dual base station offloading mode, because

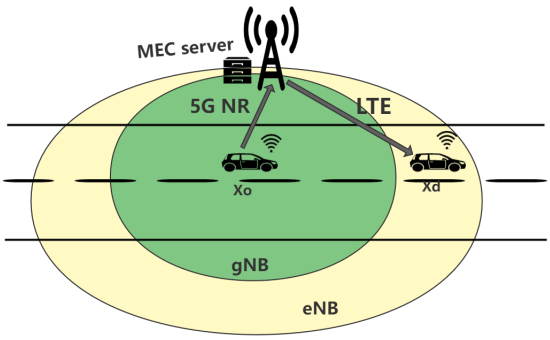


Fig. 3: Dual base station offloading mode.

the eNB has a larger coverage area and almost covers the gNB. Specifically, the x_o is situated within the gNB's coverage area, while the x_d is located in an area covered by the eNB. Therefore, there is a high probability that the vehicle will not be able to receive processing feedback before driving away from gNB's coverage. Based on the dual base station collaborative architecture, the following method can solve this problem. First, the vehicle offloads the task to the gNB for execution, then the gNB transmits the result to the eNB that can cover the destination vehicle. If the destination vehicle is located within the overlapping coverage area of multiple eNBs, then we determine which eNB will serve as the relay node according to the optimal selection of base stations [25], [33]. Finally, the eNB returns the calculation result to the requesting vehicle. These tasks require migration between base stations during the offloading process. In other words, eNB can serve as a relay node in the process of result transmission [34].

C. System Model

1) Communication model: We take into account the impact of distance on transmission rate in our offloading scheme. It is assumed that the base station has coordinate (x^{bs}, y^{bs}) and the target vehicle i has coordinate (x_i^{veh}, y_i^{veh}). Thus, we can express the distance between vehicle i and base station j as follows:

$$g_i = \sqrt{\left(x_i^{veh} - x^{bs}\right)^2 + \left(y_i^{veh} - y^{bs}\right)^2}. \tag{1}$$

In addition, based on Shannon's theory, the uplink transmission rate between vehicle i and the base station j is represented as

$$R_{i,j} = B_{i,j} \log_2 \left(1 + \frac{P_i \left|h_0\right|^2 (g_i)^{-\alpha}}{N_0 B_{i,j}}\right), \tag{2}$$

where $B_{i,j}$ represents the uplink channel bandwidth between vehicle i and base station j, while P_i refers to the uplink transmission power. The channel attenuation coefficient is denoted as h_0, and follows the complex normal distribution *CN* (0, 1). N_0 is the noise power spectral density [35], and the path loss exponent is represented by α.

2) Computation model: In what follows, we will describe two ways of calculating the task to illustrate the computation model: a) Local computing; b) Edge computing. In many

computation-intensive applications, the output data of the computational results is often significantly smaller relative to the input data. Therefore, the amount of time it takes to return the calculation results to the vehicle can be disregarded.

a) Local computing: When the vehicle opts to perform task Φ_i locally, we define D_i^l as the local processing delay, which accounts solely for the computing capacity of the local CPUs. Similarly, we introduce f_i^l as the available CPU cycles allocated for processing task Φ_i. Consequently, we can define the calculation delay of task Φ_i as

$$D_i^l = \frac{c_i}{f_i^l}. \tag{3}$$

The energy consumption of task Φ_i is represented by E_i^l, as expressed in the following way:

$$E_i^l = k \left(f_i^l\right)^2 c_i. \tag{4}$$

In Eq. 4, k represents effective switched capacitance, determined solely by chip architecture [36]. In this paper, we set $k = 10^{-27}$ [37], [38].

b) Edge computing: When local computing cannot meet latency requirements, tasks need to be executed at the edge server. Therefore, tasks need to be transmitted from the local to the corresponding base station before they can be executed at the edge. The delay in transmitting task Φ_i from vehicle i to base station j is determined by dividing the size of input data by the uplink transmission rate

$$D_{i,j}^t = \frac{d_i}{R_{i,j}}. \tag{5}$$

The base station can start the computing procedure after it has received the vehicle's offloaded task data. Hence, the base station's computation time to complete the offloaded task is

$$D_{i,j}^c = \begin{cases} \frac{c_i}{f_{i,j}}, & f_{i,j} \neq 0 \\ 0, & f_{i,j} = 0. \end{cases} \tag{6}$$

The computation resource allocated by base station j to task Φ_i is represented by $f_{i,j}$ (in CPU cycles/s). Hence, the overall latency of the task Φ_i calculated at the base station j is

$$D_{i,j} = D_{i,j}^t + D_{i,j}^c. \tag{7}$$

The corresponding energy consumption is

$$E_{i,j} = P_i^T D_{i,j}^t + P_i^I D_{i,j}^c, \tag{8}$$

where P_i^T is the power used for transmission when vehicle i is active, while P_i^I represents the power consumption when vehicle i is in an idle state.

D. Problem Formulation

The quality of user experience and the computational efficiency of the overall VEC network are closely related to two primary elements: 1) latency; 2) energy consumption. It is possible to represent the overall latency as

$$T = \sum_{i=1}^{N} \left[(1-\mu_i) D_i^l + \mu_i \Big[(1-\eta_i) D_{i,1} + \eta_i D_{i,2} \Big] \right]. \tag{9}$$

The complete latency is comprised of two components, namely, the local processing duration and the offloading latency.

The total energy consumption for a task also consists of two parts: energy used for local execution and energy used for offloading to the edge server. Therefore, to determine the energy consumption of performing tasks, the following definition can be used:

$$E = \sum_{i=1}^{N} \left[(1-\mu_i) E_i^l + \mu_i \Big[(1-\eta_i) E_{i,1} + \eta_i E_{i,2} \Big] \right]. \tag{10}$$

The paper aims to improve user experience by decreasing the global latency of task execution and reducing vehicle energy consumption. To achieve this goal, we propose the objective function of the joint task offloading and resource allocation. Essentially, the problem is an optimization one, as the aim is to minimize the total delay and energy consumption by optimizing the offloading strategy and resource allocation.

The following is an expression for the objective function:

$$\begin{aligned} Q =& \gamma T + (1-\gamma) E \\ =& \gamma \sum_{i=1}^{N} \left[(1-\mu_i) D_i^l + \mu_i \Big[(1-\eta_i) D_{i,1} + \eta_i D_{i,2} \Big] \right] \\ &+ (1-\gamma) \sum_{i=1}^{N} \left[(1-\mu_i) E_i^l + \mu_i \Big[(1-\eta_i) E_{i,1} + \eta_i E_{i,2} \Big] \right]. \end{aligned} \tag{11}$$

The problem of optimization we aim to resolve is expressed as

$$\min_{\eta_i, \mu_i, f_{i,j}} Q \tag{12}$$

$$s.t. \qquad \eta_i, \mu_i \in \{0,1\} \qquad \forall i \in N, \tag{13}$$

$$f_{i,j} \geq 0 \qquad \forall i \in N, \forall j \in \{1,2\}, \tag{14}$$

$$\sum_{i=1}^{N} f_{i,j} \leq F_j \qquad \forall i \in N, \forall j \in \{1,2\}. \tag{15}$$

The weighting factor γ can be used to adjust the tradeoff between computing latency and energy consumption in the objective function, which is defined as the weighted sum of the overall latency and energy consumption. The value of the weighting factor can vary with the situation. If the emphasis is on energy saving, the value is close to 0, and when the focus is on latency performance, the value is close to 1. In the optimization problem, Eq. 13 represents that each task is indivisible and can only be processed one way at a time; Eq. 14 represents the constraint on the amount of computation resource allocated to vehicle i; Eq. 15 indicates that the edge server's allocation of computing resources to total tasks must not surpass the maximum CPU frequency.

IV. Task Classification and Offloading

This section intends to minimize the objective function. Specifically, the optimization problem addressed in this paper falls under the category of mixed integer nonlinear programming (MINLP). It involves integer and continuous variables

Algorithm 1: Task Classification Algorithm

input : $\Phi_i = \{c_i, d_i, T_i\}$, i=1,...N, D_i, v_i, P_i^{veh}
output: Ω_i

for $i \leftarrow 1$ **to** N **do**
  $t_i = \frac{D_i}{v_i}$;
  if P_i^{veh} *is within eNB coverage, not gNB* **then**
    select single base station offloading mode;
    $\Omega_i \leftarrow \Phi_i = \{c_i, d_i, T_i\}, F_l, F_{eNB}, R_{eNB}$;
  end
  if P_i^{veh} *is within gNB coverage* **then**
    if $t_s < T_s$ **then** select single base station offloading mode;
    else select dual base station offloading mode;
    $\Omega_i \leftarrow \Phi_i = \{c_i, d_i, T_i\}, F_l, F_{gNB}, R_{gNB}$;
  end
  return Ω_i
end

as well as nonlinear feasible regions, etc [39]. To begin, we introduce the task classification algorithm, which helps determine the offloading node and method for each task, that is, if the task chooses to offload, whether it is performed by a single base station or collaboratively by two base stations. Then we use the linear relaxation improved branch-and-bound algorithm (BBA) to find the optimal solution.

A. Task Classification Algorithm

After receiving a task request from target vehicle i, the MEC server retrieves the vehicle's speed and location. If the vehicle i is in the coverage area of the eNB when sending the request, the task can only be executed by the eNB alone if it is selected for edge execution. If the vehicle i is located within the coverage area of the gNB, it is necessary to determine whether the task is performed by the gNB alone or collaboratively by the dual base stations. First, we calculate the driving time t_i of vehicle i in the range of communication, which is determined by dividing the driving distance D_i within the range by its speed v_i. Second, we evaluate the relationship between driving time t_i and the delay limitation T_i. If $t_i > T_i$, the task Φ_i would be calculated by a single base station. Otherwise, the task Φ_i would be calculated collaboratively by gNB and eNB. The specific method of task classification is presented in Algorithm 1.

B. Branch-and-Bound Algorithm

We use the BBA algorithm to solve the non-convex MINLP problem. Firstly, we transform it into a convex optimization problem. Then, we combine this convex optimization problem with a branch-and-bound constraint framework as a solution to the original problem. The specific calculation steps are as follows.

According to Algorithm 1, our decision set $\eta = \{\eta_1, \eta_2, ..., \eta_n\}$ can be determined, so we only need to relax the 0-1 variable μ_i to $0 \le \mu_i \le 1$. To prevent errors caused by a denominator of 0 (i.e., when $f_{i,j}$ = 0), a new variable δ will be defined. Therefore, the original objective function Q is transformed into a new function $Q1$.

$Q1$:

$$\begin{aligned}\min_{\mu_i, f_i^j} \gamma \sum_{i=1}^{N} \Bigg[& (1-\mu_i)\frac{c_i}{f_i^l} + \mu_i \Big[(1-\eta_i)\Big(\frac{d_i}{R_{i,1}} + \frac{c_i}{f_{i,1}+\delta}\Big) \\ & + \eta_i \Big(\frac{d_i}{R_{i,2}} + \frac{c_i}{f_{i,2}+\delta}\Big)\Big]\Bigg] \\ & + (1-\gamma)\sum_{i=1}^{N}\Bigg[(1-\mu_i)\,\kappa (f_i^l)^2 c_i \\ & + \mu_i\Big[(1-\eta_i)\Big(P_i^T \frac{d_i}{R_{i,1}} + P_i^I \frac{c_i}{f_{i,1}+\delta}\Big) \\ & + \eta_i\Big(P_i^T \frac{d_i}{R_{i,2}} + P_i^I \frac{c_i}{f_{i,2}+\delta}\Big)\Big]\Bigg] \qquad (16)\end{aligned}$$

$$s.t. \quad 0 \le \mu_i \le 1 \qquad \forall i \in N. \qquad (17)$$

Solving problem $Q1$ yields both the upper and lower bounds of the objective function Q. Next, we proceed to reformulate the problem as $Q2$ by introducing an auxiliary variable, in which $\alpha_{i,j} = (f_{i,j} + \delta)^{-1}$.

$Q2$:

$$\begin{aligned}\min_{\mu_i, \alpha_{i,j}} \gamma \sum_{i=1}^{N} \Bigg[& (1-\mu_i)\frac{c_i}{f_i^l} + \mu_i \Big[(1-\eta_i)\Big(\frac{d_i}{R_{i,1}} + c_i\alpha_{i,1}\Big) \\ & + \eta_i \Big(\frac{d_i}{R_{i,2}} + c_i\alpha_{i,2}\Big)\Big]\Bigg] \\ & + (1-\gamma)\sum_{i=1}^{N}\Bigg[(1-\mu_i)\kappa (f_i^l)^2 c_i \\ & + \mu_i\Big[(1-\eta_i)\Big(P_i^T \frac{d_i}{R_{i,1}} + P_i^I c_i\alpha_{i,1}\Big) \\ & + \eta_i\Big(P_i^T \frac{d_i}{R_{i,2}} + P_i^I c_i\alpha_{i,2}\Big)\Big]\Bigg] \qquad (18)\end{aligned}$$

$$s.t. \quad 0 \le \mu_i \le 1 \qquad \forall i \in N \qquad (19)$$

$$\frac{1}{\delta + \mu_i F_j} \le \alpha_{i,j} \le \frac{1}{\delta} \qquad \forall i \in N, \forall j \in \{1,2\} \qquad (20)$$

$$\sum_{i=1}^{N}\left(\frac{1}{\alpha_{i,j}} - \delta\right) \le F_j \qquad \forall i \in N, \forall j \in \{1,2\}. \qquad (21)$$

The objective function in problem $Q2$ contains a quadratic form with discrete variables, rendering it nonconvex. To resolve this, a new variable will be defined to replace the quadratic form, enabling the transformation of problem $Q2$ into an optimization problem with convex constraints. We define $\xi_{i,j} = \mu_i \cdot \alpha_{i,j}$. The range of μ_i is $0 \le \mu_i \le 1$ and the range of $\alpha_{i,j}$ is $\frac{1}{\delta+\mu_i F_j} \le \alpha_{i,j} \le \frac{1}{\delta}$. So, there are some constraints on the new variable $\xi_{i,j}$

$$\begin{cases} (\mu_i - 0) * (\alpha_{i,j} - \frac{1}{\delta+\mu_i F_j}) \ge 0 \\ (1-\mu_i) * (\alpha_{i,j} - \frac{1}{\delta+\mu_i F_j}) \ge 0 \\ (\mu_i - 0) * (\frac{1}{\delta} - \alpha_{i,j}) \ge 0 \\ (1-\mu_i) * (\frac{1}{\delta} - \alpha_{i,j}) \ge 0. \end{cases} \qquad (22)$$

Then, replacing $\xi_{i,j} = \mu_i \cdot \alpha_{i,j}$

$$\begin{cases} \xi_{i,j} - \mu_i \frac{1}{\delta + \mu_i F_j} \geq 0 \\ \alpha_{i,j} - \frac{1}{\delta + \mu_i F_j} - \xi_{i,j} + \mu_i \frac{1}{\delta + \mu_i F_j} \geq 0 \\ \mu_i \frac{1}{\delta} - \xi_{i,j} \geq 0 \\ \frac{1}{\delta} - \alpha_{i,j} - \mu_i \frac{1}{\delta} + \xi_{i.j} \geq 0. \end{cases} \tag{23}$$

The convex optimization problem $Q3$ is obtained by replacing the objective function of $Q2$ with the new variable.

$Q3$:

$$\begin{aligned} \min_{\mu_i, \xi_{i,j}} \gamma \sum_{i=1}^{N} \Bigg[& (1-\mu_i) \frac{c_i}{f_i^l} + (1-\eta_i) \left(\mu_i \frac{d_i}{R_{i,1}} + c_i \xi_{i,1} \right) \\ & + \eta_i \left(\mu_i \frac{d_i}{R_{i,2}} + c_i \xi_{i,2} \right) \Bigg] \\ & + (1-\gamma) \sum_{i=1}^{N} \Bigg[(1-\mu_i) \kappa \left(f_i^l \right)^2 c_i \\ & + (1-\eta_i) \left(\mu_i P_i^T \frac{d_i}{R_{i,1}} + P_i^I c_i \xi_{i,1} \right) \\ & + \eta_i \left(\mu_i P_i^T \frac{d_i}{R_{i,2}} + P_i^I c_i \xi_{i,2} \right) \Bigg] \\ s.t. \quad & (19), (20), (21), (23). \end{aligned} \tag{24}$$

Through the solution of the convex optimization problem, we can readily obtain the best solution for problem $Q3$ and consequently derive the lower bound of the objective function Q. Then, the following method enables us to obtain the value of μ_i:

$$\mu_i = \begin{cases} 0, & \mu_i \geq 0.5 \\ 1, & \mu_i < 0.5 \end{cases} \qquad \forall i \in N. \tag{25}$$

When the value of μ_i is determined, we assume that m tasks are processed locally and $N - m$ tasks are offloaded to edge computing. Therefore, we can rephrase problem $Q1$ as

$Q4$:

$$\begin{aligned} \min_{f_i^j} \sum_{i=1}^{m} & \left[\gamma \frac{c_i}{f_i^l} + (1-\gamma) \kappa (f_i^l)^2 c_i \right] \\ & + \sum_{i=m+1}^{N} \Bigg[(1-\eta_i) \Big[(\gamma + (1-\gamma) P_i^T) \frac{d_i}{R_{i,1}} \\ & + (\gamma + (1-\gamma) P_i^I) \frac{c_i}{f_{i,1} + \delta} \Big] \\ & + \eta_i \Big[(\gamma + (1-\gamma) P_i^T) \frac{d_i}{R_{i,2}} + (\gamma + (1-\gamma) P_i^I) \frac{c_i}{f_{i,2} + \delta} \Big] \Bigg] \end{aligned} \tag{26}$$

$$s.t. \quad (14), (15). \tag{27}$$

By finding the optimal solution for $Q4$, we can determine the upper limit of objective function Q. $Q3$ and $Q4$ problems help us find the lower and upper limits of the objective function. Subsequently, we utilize the branch-and-bound algorithm, described in Algorithm 2, to discover the desirable solution for the initial problem.

Algorithm 2: BBA Algorithm

Initialization:
set the upper bound $UB = +\infty$ and $Q^* = \emptyset$
initialize the original problem Q
Linear Relaxation:
obtain problem $Q1$ from Q
Convex optimization:
obtain problem $Q3$ from Q
obtain the lower bound LB^* of the Q by solving $Q3$
Iteration:
compare LB_{n_1} and LB_{n_2} and select the lowest value LB_n in Q
set $LB = LB_n$
obtain suitable value of μ according to Eq. 25.
obtain the upper bound UB_n of the Q by solving $Q4$ and get the solution $OPT_n = (\mu, f)$
If $UB_n < UB$
update $UB = UB_n$ and $OPT = OPT_n$
If $UB == LB$, output the optimal solution OPT
otherwise, remove $LB_n > UB$ branches
Branch:
split the current branch into n_1 and n_2
Bounding:
obtain the lower bound LB_{n_1} and LB_{n_2} by solving n_1 and n_2
If $LB_{n_1} < UB$, insert n_1 into Q
If $LB_{n_2} < UB$, insert n_2 into Q
If $Q == \emptyset$, finish iteration and output the current OPT
otherwise, proceed to the next iteration

Output: the optimal solution UB and OPT

TABLE II: Execution time of the BBA

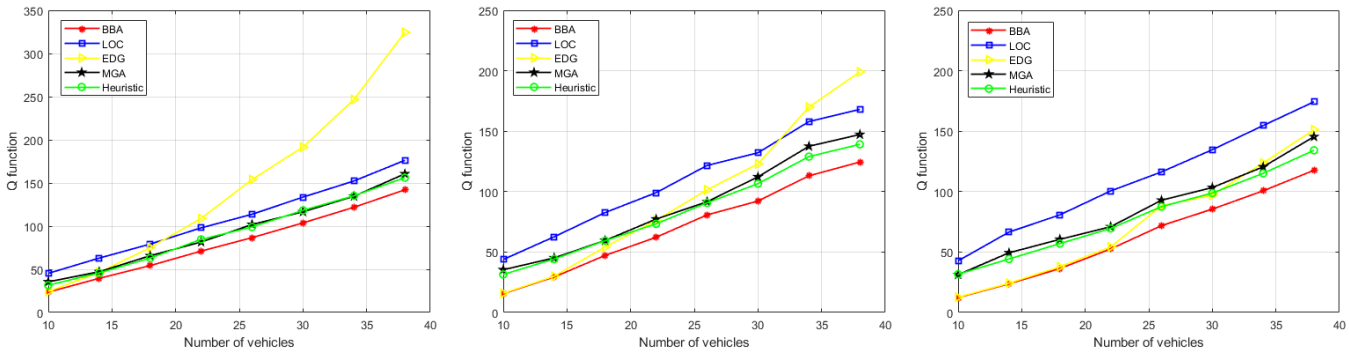
Number of vehicles	Execution time (s)
10	4.98
20	11.27
30	21.59
40	42.70

C. Computational Complexity

In this subsection, we analyze the complexity of the algorithms proposed in this paper and the execution time in different task scenarios.

Complexity analysis: For Algorithm 1, it is necessary to traverse each task to determine the type of task that belongs to a single base station execution or dual base station execution, so the time complexity is $O(n)$. For Algorithm 2, the branch-and-bound algorithm is an iterative algorithm, each iteration solving the corresponding relaxation of the linear programming problem, and the iteration is ended when the upper and lower bounds are very close to each other. Thus when determining the optimal offloading decision for n tasks, the time complexity of the algorithm is $O(2^n)$. This results in a final time complexity of $O(n + 2^n)$.

Execution time analysis: We tested the execution time of the algorithm for different numbers of vehicles on a laptop computer equipped with an Intel(R) Core(TM) i5-13500H CPU at 2.90 GHz and 32 GB of RAM. The development tool



(a) Edge server CPU of 5 GHz (b) Edge server CPU of 8 GHz (c) Edge server CPU of 11 GHz

Fig. 4: Q function vs. edge server CPU size.

was MATLAB R2022b installed on a Windows 11 Home 64-bit platform. The specific experimental results are shown in Tab. II. It can be seen that the execution time of the algorithm increases according to the trend of 2^n, and it can show better performance when the number of vehicles is within a reasonable range.

Considering the complexity and longer computational time of such algorithms, there has been a lot of research on using machine learning assisted branch and bound algorithms to solve combinatorial optimization problems with significant results [40], [41], making the branch-and-bound algorithm better in terms of both solution accuracy and computational time.

V. Simulation Results

In this section, we perform a comprehensive analysis of simulation results aimed at evaluating the performance of the proposed offloading scheme.

A. Setup

We consider a bidirectional road that spans 3000 meters and has base stations situated along it. Each of these base stations is equipped with a VEC server. Edge servers have a computational capacity of at least 5 GHz, while the CPU frequency of the vehicles is 0.8 GHz. According to the system model, in a certain time slot, there are N vehicles that simultaneously send task processing requests to the edge servers. Additionally, we assume that the vehicle speed belongs to [30,80] km/h and the vehicle is traveling at a constant speed on the road. The size of the task and required computation resources follow Gaussian distributions: $d_i \sim N$ (900, 300) KB and $c_i \sim N$ (2000, 200) MHz, respectively. The maximum latency constraint for each computation task is randomly generated from a uniform distribution, $T_i \sim$ U [1, 5] s. The simulations are performed in MATLAB. The communication parameters used in our simulations are summarized in Tab. III [34].

To conduct a comparative analysis, this paper explores four methods through simulation experiments.

1) **The heuristic scheme (Heuristic)** [2]. Within the VEC architecture based on heterogeneous cellular networks, the heuristic scheme specifies that only tasks whose local computation cannot meet the maximum tolerated latency are offloaded to the VEC server for execution.
2) **Mobility-aware greedy algorithm (MGA)** [15]. According to the communication delay of vehicles within the range of the base station, the algorithm prioritizes the computing resources to the vehicles with short communication delay, which can reduce the offloading failure rate.
3) **Local computing algorithm (LOC)**. This algorithm involves performing all tasks on the local vehicles without any task offloading, where $\mu_i = 0, \forall i \in N$.
4) **Edge computing algorithm (EDG)**. Unlike the LOC algorithm, this algorithm doesn't require local computing. Instead, it offloads all tasks to the edge server for execution, where $\mu_i = 1, \forall i \in N$.

TABLE III: Main Parameters Setting

Parameter	Value
L_{eNB}	1000 m
L_{gNB}	300 m
d_i	N (900, 300) KB
c_i	N (2000, 200) MHz
T_i	U [1, 5] s
$B_{i,eNB}$	1 MHz
$B_{i,gNB}$	10 MHz
P_i^T	100 mW
P_i^I	10 mW
α	2
N_0	5×10^{-5} W
γ	0.8

B. Q Function

Fig. 4 shows the objective function Q value versus the varying number of vehicles with different edge server CPU sizes. In order to demonstrate the distinction, we have set the CPU size of each base station to 5, 8, and 11 GHz, as illustrated in Fig. 4 (a)–(c), respectively. To ensure a fair comparison,

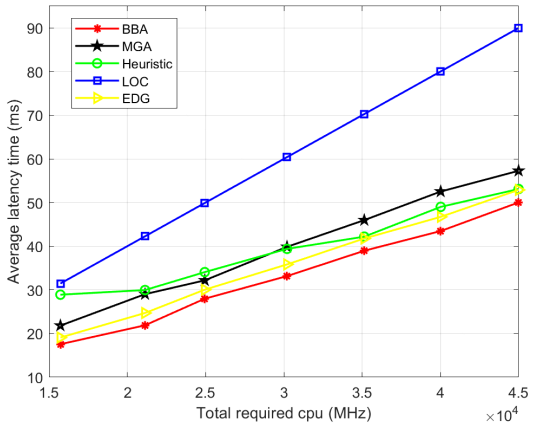


Fig. 5: Average latency vs. total required CPU.

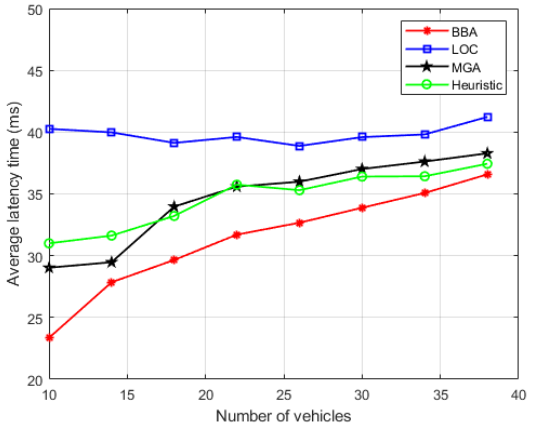


Fig. 6: Average latency vs. number of vehicles.

we have established an equal task size range for each vehicle. We can notice that the Q value of all schemes grows as the CPU size increases except for the local computing algorithm, and especially the edge computing algorithm changes most obviously. Moreover, the MGA algorithm and the heuristic scheme have similar Q values, however, as the edge server CPU size increases, the performance of the heuristic scheme is obviously better than MGA, that is because, in the case of sufficient edge computing resources, the Heuristic can basically guarantee that all tasks are completed within the maximum tolerable delay while MGA can only ensure the success rate of offloading tasks with high priority. After comparing the behaviors of various schemes, it becomes evident that the proposed scheme achieves the lowest Q value when tested on CPUs of all sizes. The reason is that the proposed BBA algorithm takes into consideration both the benefits of edge computing and local processing capabilities. Balancing these factors can greatly diminish the overall latency and energy consumption. In addition, dual base station cooperative offloading allows more tasks to be performed at the edge. Therefore, the BBA algorithm always has the lowest value.

C. Latency

The effect of the total required CPU on the average latency is depicted in Fig. 5. We assume that the CPU of the server is 8 GHz and the number of vehicles is 20. It can be observed that the average latency of all schemes increases rapidly as the total required CPU increases, especially for the LOC algorithm. This implies that the computing resource required to complete the task is one of the most critical factors affecting the completion delay. Furthermore, we can note that when the total required CPU is lower, the performance of MGA is better than that of Heuristic. As the total required CPU increases, the average latency of the Heuristic is gradually lower than that of MGA. That is because when the task requires a small number of computing resources, as long as the maximum tolerated delay is not exceeded by the local computing latency, the Heuristic will give priority to the task to be executed locally, and MGA will give priority to the edge offloading regardless of the size of the required resources. Therefore, as the required computing resources increase, due to the limited edge resources, low-priority tasks can only be executed locally in MGA, thereby increasing the average latency. BBA jointly optimizes local and edge computing resources to give the most reasonable offloading decision to ensure the lowest global delay and energy consumption. As a result, the BBA algorithm outperforms other schemes.

Fig. 6 illustrates the influence of the varying number of vehicles on the average latency of total tasks. Here, we fixed the maximum tolerable delay of tasks as 3 seconds, and the computing resources required by each task are in the same value space. Due to the limitation of edge resources, the more vehicles there are, the less effective resources are allocated to each task, so the average latency of the EDG algorithm grows rapidly as the number of vehicles increases. For the other algorithms, we can see from Fig. 6 that except the local computing algorithm remains stable regardless of the number of vehicles, the average latency of other algorithms increases slowly with the increase of vehicles and gradually approaches the local computing. This result is interpreted by the fact that the more vehicles, the greater the competition for edge resources, which will lead to some vehicles being forced to compute locally due to insufficient resource allocation, so the average latency gradually increases. Since local computing without resource competition itself, it can maintain a relatively stable trend, but it also has the largest delay compared to other algorithms. As a result, it can be demonstrated that a task offloading and resource allocation strategy is required in a situation where many vehicles compete for limited resources.

D. Energy Consumption

As displayed in Fig. 7, with the data size increases, it becomes apparent that more energy is needed for computation. In addition, we can also notice that MGA has the lowest energy consumption when the task is smaller. However, its energy consumption exceeds that of the other two algorithms with the task increases. It is because when the tasks generated by the vehicles are small, tasks are more likely to be processed

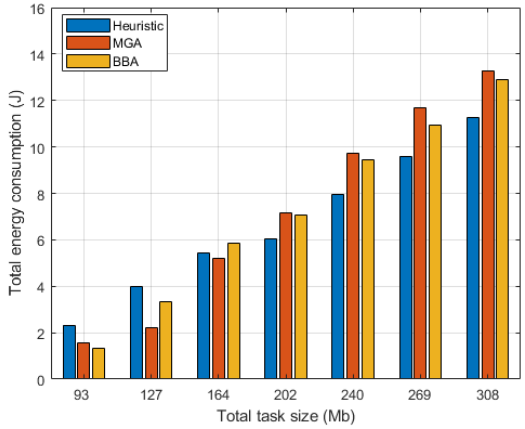


Fig. 7: Total energy consumption vs. total task size.

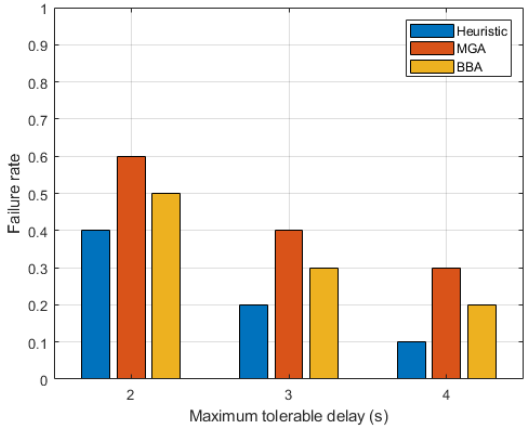


Fig. 9: Failure rate vs. maximum tolerable delay.

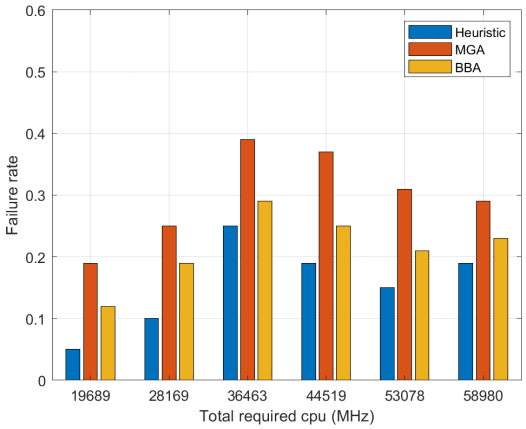


Fig. 8: Failure rate vs. total required CPU.

locally according to the heuristic scheme, but in MGA the tasks are prioritized to be computed at the edge. When the task size increases, the MGA does not have enough resources for low-priority tasks, so they can only process locally. The BBA algorithm proposed in this paper needs to consider both global delay and energy consumption to make the best offloading strategy, while the heuristic scheme only needs to ensure that the task is finished within the maximum tolerated delay. In conclusion, the heuristic algorithm outperforms other algorithms in terms of energy consumption when the total task size is larger.

E. Failure Rate

The failure rate of the execution for the tasks offloading is another key metric to assess the effectiveness of the vehicular edge computing system. Since the LOC and EDG schemes are not part of the optimization algorithm, the task failure rate is always the largest for LOC and the second highest for the EDG scheme under the same comparison conditions. In this subsection, we mainly compare the performance of the other three optimization algorithms.

As intuitively shown in Fig. 8, with the increase of total required CPU, the task completion failure rate of each scheme does not display a gradual upward trend. This also demonstrates that the success rate of task execution depends on various factors, including the size of the task data, the location of the vehicle when sending the request, and the number of vehicles requesting access to the same base station. Most significantly, it can also be observed that the performance of the Heuristic in the failure rate is better than that of BBA and MGA. Compared to BBA, the heuristic scheme determines the offloading strategy by comparing the local computing delay with the maximum tolerable delay regardless of the overall latency and energy consumption, so it can ensure that most tasks are processed with the maximum tolerable delay. MGA prioritizes the allocation of computing resources to tasks with short communication delays. Although the success rate of high-priority tasks is guaranteed, it leads to unreasonable utilization of resources, and low-priority tasks can only be forced to calculate locally. Therefore, the way of dual base station cooperative offloading can not only ensure the success rate of tasks with short communication delays but also make rational use of edge resources.

Fig. 9 shows the failure rate of total task executions by changing the fixed maximum tolerable delay. It can be seen that with the increase of the maximum tolerable delay, the failure rate of the three algorithms is all reduced, and in any case, the heuristic scheme has the lowest value and the BBA is second only to it. The reason is that the Heuristic allocates edge computing resources to tasks with a time limit of maximum tolerable latency. Based on the same edge resources, Heuristic allows more tasks to be executed at the edge server than BBA, so Heuristic actually guarantees the success rate by extending the completion latency of tasks. In addition, BBA and Heuristic both promise the successful offloading of short communication delay tasks by collaborating with the dual base station, while MGA allocates edge resources by priority. Although tasks with high priority can successfully complete offloading, this allocation strategy leads to a higher global failure rate than other algorithms. Therefore, it can be seen that dual base station collaboration to complete task offloading is necessary to increase the success rate of IoV applications.

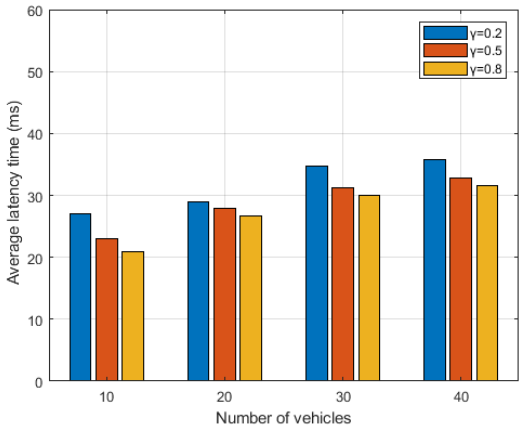


Fig. 10: Average latency vs. number of vehicles.

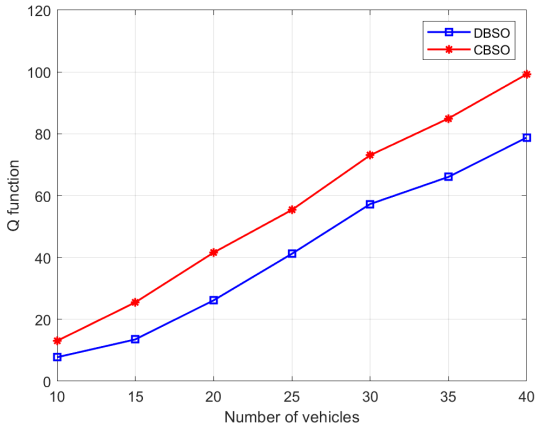


Fig. 12: Q function vs. number of vehicles.

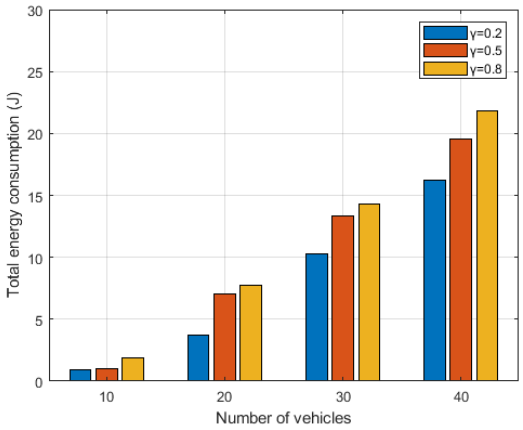


Fig. 11: Total energy consumption vs. number of vehicles.

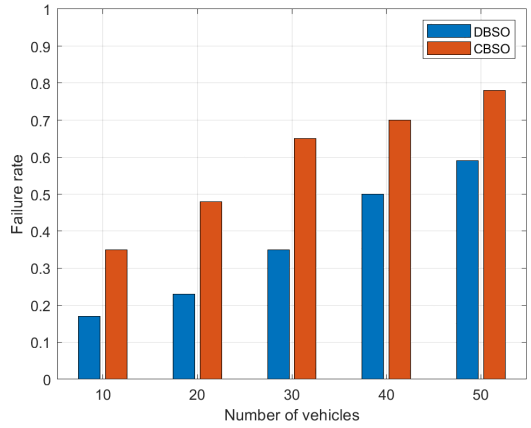


Fig. 13: Failure rate vs. number of vehicles.

F. Impact of Weighting Parameter γ

In order to explore the effect of the weighting parameter γ on the proposed objective function. We analyzed the average latency and total energy consumption of all tasks when γ was set to 0.2, 0.5, and 0.8 for scenarios of different numbers of vehicles (i.e., 10, 20, 30, and 40), respectively.

It can be seen in Fig. 10, the average latency of the task gradually decreases as γ increases. This shows that the larger the value of γ, the more the algorithm is optimized for latency performance. Similarly, Fig. 11 indicates that when γ is smaller, the algorithm focuses more on optimizing the energy consumption of the task.

Therefore, we can conclude that the optimization focus of the algorithm can be effectively controlled by adjusting the trade-off parameter γ. Adapting γ to the actual needs of the task is essential to improve the quality of user experience. If the task is latency intensive, then γ can be set larger to reduce the completion delay of the task. Conversely, setting γ smaller reduces the energy consumption of the task, which can result in longer endurance for the base station and the vehicle. Combined with Fig. 4, we can demonstrate that the BBA algorithm can achieve a balance between task latency and energy consumption, and maintain both latency and energy consumption at a low level.

G. Effectiveness and Compatibility

In order to explore the effectiveness and compatibility of collaborative task offloading for heterogeneous cellular networks in multiple-base station scenarios, we present the following experimental results in combination with the cross-region base station selection offloading scheme (CBSO) [12]. Specifically, CBSO addresses three different scenarios of computation offloading in 5G-enabled EC-IoV systems and gives the corresponding resource allocation strategy, which effectively manages the edge computation resources while reducing the system energy consumption and time cost. We use dual base station offloading scheme (DBSO) to demonstrate the effectiveness and compatibility of our approach. Specifically, DBSO adopts CBSO to solve the problem of task offloading node selection across regions, and uses our proposed collaborative offloading in heterogeneous cellular networks to solve the task offloading and resource allocation under the same region.

It can be seen from Fig. 12 and Fig. 13 that both Q-value and success rate of heterogeneous cellular network collaborative offloading are better than CBSO. Although CBSO can

effectively improve the load balancing between cross-region base stations, a single base station in the same region is still unable to support a large number of concurrent task requests. In the non-standalone networking period of 5G, the use of heterogeneous cellular networks for cooperative offloading can further divide the tasks under the same region effectively, and improve the success rate of task offloading while reducing the total latency and energy consumption. Based on the experimental data in Fig. 12 and Fig. 13. We demonstrate that the use of heterogeneous cellular network cooperative offloading can further reduce the task completion delay compared to the single base station, and the heterogeneous cellular network cooperative task offloading algorithm also has good compatibility with the existing base station selection algorithms in multiple base station offloading scenarios.

VI. Conclusion

In this paper, we introduced a VEC architecture based on the heterogeneous cellular network and presented a joint optimization problem for task offloading and resource allocation, which aims to maximize computational efficiency and minimize total completion latency and energy consumption. The problem was formulated as an MINLP problem, which could be resolved through a joint solution involving the task classification algorithm and the BBA algorithm. Simulation results demonstrated that the system proposed in this paper performed better than existing methods in computing latency, energy consumption, and failure rate. For future work, we would like to classify tasks according to their criticality in order to select an offloading strategy that better meets the quality of user experience as well as the rational use of edge resources.

References

[1] R. Meneguette, R. De Grande, J. Ueyama, G. P. R. Filho, and E. Madeira, "Vehicular edge computing: Architecture, resource management, security, and challenges," *ACM Comput. Surv.*, vol. 55, no. 1, nov 2021. [Online]. Available: https://doi.org/10.1145/3485129

[2] K. Zhang, Y. Mao, S. Leng, Y. He, and Y. ZHANG, "Mobile-edge computing for vehicular networks: A promising network paradigm with predictive off-loading," *IEEE Vehicular Technology Magazine*, vol. 12, no. 2, pp. 36–44, 2017.

[3] X. Hu, K.-K. Wong, and K. Yang, "Wireless powered cooperation-assisted mobile edge computing," *IEEE Transactions on Wireless Communications*, vol. 17, no. 4, pp. 2375–2388, 2018.

[4] S. He, K. Shi, C. Liu, B. Guo, J. Chen, and Z. Shi, "Collaborative sensing in internet of things: A comprehensive survey," *IEEE Communications Surveys & Tutorials*, vol. 24, no. 3, pp. 1435–1474, 2022.

[5] B. Cao, Z. Sun, J. Zhang, and Y. Gu, "Resource allocation in 5g iov architecture based on sdn and fog-cloud computing," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 6, pp. 3832–3840, 2021.

[6] M. Gonzalez-Martín, M. Sepulcre, R. Molina-Masegosa, and J. Gozalvez, "Analytical models of the performance of c-v2x mode 4 vehicular communications," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 2, pp. 1155–1166, 2019.

[7] D. Kombate and Wanglina, "The internet of vehicles based on 5g communications," in *2016 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, 2016, pp. 445–448.

[8] X. Yin, J. Liu, X. Cheng, and X. Xiong, "Large-size data distribution in iov based on 5g/6g compatible heterogeneous network," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 7, pp. 9840–9852, 2022.

[9] M. Agiwal, H. Kwon, S. Park, and H. Jin, "A survey on 4g-5g dual connectivity: Road to 5g implementation," *IEEE Access*, vol. 9, pp. 16 193–16 210, 2021.

[10] N. An, C. Wang, and W. Wang, "Interference coexistence of 5g nr and lte system based on 2.1ghz," in *2020 12th International Conference on Communication Software and Networks (ICCSN)*, 2020, pp. 90–94.

[11] S. Raza, S. Wang, M. Ahmed, M. R. Anwar, M. A. Mirza, and W. U. Khan, "Task offloading and resource allocation for iov using 5g nr-v2x communication," *IEEE Internet of Things Journal*, 2021.

[12] S. Wan, R. Gu, T. Umer, K. Salah, and X. Xu, "Toward offloading internet of vehicles applications in 5g networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 7, pp. 4151–4159, 2021.

[13] J. Feng, Z. Liu, C. Wu, and Y. Ji, "Ave: Autonomous vehicular edge computing framework with aco-based scheduling," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 12, pp. 10 660–10 675, 2017.

[14] Y. Jang, J. Na, S. Jeong, and J. Kang, "Energy-efficient task offloading for vehicular edge computing: Joint optimization of offloading and bit allocation," in *2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*, 2020, pp. 1–5.

[15] S. Choo, J. Kim, and S. Pack, "Optimal task offloading and resource allocation in software-defined vehicular edge computing," in *2018 International conference on information and communication technology convergence (ICTC)*. IEEE, 2018, pp. 251–256.

[16] C. Zhu, J. Tao, G. Pastor, Y. Xiao, Y. Ji, Q. Zhou, Y. Li, and A. Ylä-Jääski, "Folo: Latency and quality optimized task allocation in vehicular fog computing," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4150–4161, 2018.

[17] B. Qiao, C. Liu, J. Liu, Y. Hu, K. Li, and K. Li, "Task migration computation offloading with low delay for mobile edge computing in vehicular networks," *Concurrency and Computation: Practice and Experience*, vol. 34, no. 1, p. e6494, 2022.

[18] R. Zhang, L. Wu, S. Cao, X. Hu, S. Xue, D. Wu, and Q. Li, "Task offloading with task classification and offloading nodes selection for mec-enabled iov," *ACM Transactions on Internet Technology (TOIT)*, vol. 22, no. 2, pp. 1–24, 2021.

[19] Y. Jang, J. Na, S. Jeong, and J. Kang, "Energy-efficient task offloading for vehicular edge computing: Joint optimization of offloading and bit allocation," in *2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*. IEEE, 2020, pp. 1–5.

[20] S. Li, N. Zhang, R. Jiang, Z. Zhou, F. Zheng, and G. Yang, "Joint task offloading and resource allocation in mobile edge computing with energy harvesting," *Journal of Cloud Computing*, vol. 11, no. 1, pp. 1–14, 2022.

[21] Y. Lu, M.-X. Luo, and X. Wang, "Large-scale mobile edge computing with joint offloading decision and resource allocation," in *International Conference on Artificial Intelligence and Security*. Springer, 2022, pp. 271–286.

[22] F. Zeng, Q. Chen, L. Meng, and J. Wu, "Volunteer assisted collaborative offloading and resource allocation in vehicular edge computing," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 6, pp. 3247–3257, 2020.

[23] Q. Peng, Y. Jia, L. Liang, and Z. Chen, "A task assignment scheme for parked-vehicle assisted edge computing in iov," in *2021 IEEE 93rd Vehicular Technology Conference (VTC2021-Spring)*. IEEE, 2021, pp. 1–5.

[24] S. Kapoor, D. Grace, and T. Clarke, "A base station selection scheme for handover in a mobility-aware ultra-dense small cell urban vehicular environment," in *2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, Oct 2017. [Online]. Available: http://dx.doi.org/10.1109/pimrc.2017.8292760

[25] Q. Hua, K. Yu, Z. Wen, and T. Sato, "A novel base-station selection strategy for cellular vehicle-to-everything (c-v2x) communications," *Applied Sciences*, p. 556, Feb 2019. [Online]. Available: http://dx.doi.org/10.3390/app9030556

[26] C. Skouroumounis, C. Psomas, and I. Krikidis, "Low-complexity base station selection scheme in mmwave cellular networks," *IEEE Transactions on Communications*, vol. 65, no. 9, pp. 4049–4064, 2017.

[27] Y. Dai, D. Xu, S. Maharjan, and Y. Zhang, "Joint load balancing and offloading in vehicular edge computing and networks," *IEEE Internet of Things Journal*, p. 4377–4387, Jun 2019. [Online]. Available: http://dx.doi.org/10.1109/jiot.2018.2876298

[28] S. Wan, X. Li, Y. Xue, W. Lin, and X. Xu, "Efficient computation offloading for internet of vehicles in edge computing-assisted 5g networks," *The Journal of Supercomputing*, p. 2518–2547, Apr 2020. [Online]. Available: http://dx.doi.org/10.1007/s11227-019-03011-4

[29] J. Yan, J. Wu, Y. Wu, L. Chen, and S. Liu, "Task offloading algorithms for novel load balancing in homogeneous fog network," in *2021 IEEE 24th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, May 2021. [Online]. Available: http://dx.doi.org/10.1109/cscwd49262.2021.9437748

[30] L. Tang, B. Tang, L. Zhang, F. Guo, and H. He, "Joint optimization of network selection and task offloading for vehicular edge computing," *Journal of Cloud Computing*, Dec 2021. [Online]. Available: http://dx.doi.org/10.1186/s13677-021-00240-y

[31] T. Tran and D. Pompili, "Joint task offloading and resource allocation for multi-server mobile-edge computing networks," *IEEE Transactions on Vehicular Technology,IEEE Transactions on Vehicular Technology*, May 2017.

[32] T. B. Iliev, G. Y. Mihaylov, I. S.Stoyanov, and E. P. Ivanova, "Lte and 5g nr – coexistence and collaboration," in *2020 43rd International Convention on Information, Communication and Electronic Technology (MIPRO)*, 2020, pp. 389–392.

[33] P. Dastranj, V. Solouk, and H. Kalbkhani, "Energy-efficient deep-predictive airborne base station selection and power allocation for uav-assisted wireless networks," *Computer Communications*, vol. 191, pp. 274–284, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0140366422001475

[34] S. Wan, R. Gu, T. Umer, K. Salah, and X. Xu, "Toward offloading internet of vehicles applications in 5g networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 7, pp. 4151–4159, 2020.

[35] P. Zhao, H. Tian, C. Qin, and G. Nie, "Energy-saving offloading by jointly allocating radio and computational resources for mobile edge computing," *IEEE Access*, vol. 5, pp. 11 255–11 268, 2017.

[36] X. He, S. Wang, X. Wang, S. Xu, and J. Ren, "Age-based scheduling for monitoring and control applications in mobile edge computing systems," in *IEEE INFOCOM 2022 - IEEE Conference on Computer Communications*, 2022, pp. 1009–1018.

[37] A. P. Miettinen and J. K. Nurminen, "Energy efficiency of mobile clients in cloud computing," in *2nd USENIX Workshop on Hot Topics in Cloud Computing (HotCloud 10)*. Boston, MA: USENIX Association, Jun. 2010. [Online]. Available: https://www.usenix.org/conference/hotcloud-10/energy-efficiency-mobile-clients-cloud-computing

[38] Z. Li, W. Wei, T. Zhang, M. Wang, S. Hou, and X. Peng, "Online multi-expert learning for visual tracking," *IEEE Transactions on Image Processing*, vol. 29, pp. 934–946, 2020.

[39] X. Deng, Z. Sun, D. Li, J. Luo, and S. Wan, "User-centric computation offloading for edge computing," *IEEE Internet of Things Journal*, vol. 8, no. 16, pp. 12 559–12 568, 2021.

[40] C. W. Parsonson, A. Laterre, and T. D. Barrett, "Reinforcement learning for branch-and-bound optimisation using retrospective trajectories," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 4, 2023, pp. 4061–4069.

[41] H. He, H. Daume III, and J. M. Eisner, "Learning to search in branch and bound algorithms," in *Advances in Neural Information Processing Systems*, Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Weinberger, Eds., vol. 27. Curran Associates, Inc., 2014. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2014/file/757f843a169cc678064d9530d12a1881-Paper.pdf

Xinggang Fan (Member, IEEE) received his Ph.D. degree in 2004 from Zhejiang University, now he is the professor with the college of Zhijiang in Zhejiang University of Technology. His main research interests include wireless sensor network, Internet of thing. He has published more than 40 referred technical papers in proceedings and journals like IEEE Transactions on Mobile Computing, IEEE Systems Journal, etc.

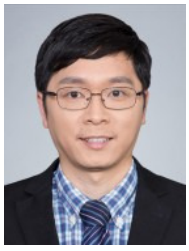
Wenting Gu (Student Member, IEEE) received the B.S. degree in Software Engineering from Shandong Management University, Jinan, China, in 2021. She is currently pursuing the M.S. degree in Computer Science and Technology, Software Engineering at Zhejiang University of Technology, Hangzhou, China.

Her current research interests include Internet of Things, Edge Computing, and Wireless Sensor Networks.

Changqing Long (Student Member, IEEE) received the B.S. degree in Applied Mathematics and M.S. degree in Operational Research and Cybernetics from the School of Mathematics and Statistics, South-Central Minzu University, Wuhan, China, in 2018 and 2021, respectively. He is currently pursuing the Ph.D. degree in control science and engineering with the College of Control Science and Engineering, Zhejiang University, Hangzhou, China.

His current research interests include Edge Computing, Stability Analysis, and Machine Learning.

Chaojie Gu (Member, IEEE) received the B.Eng. degree in information security from the Harbin Institute of Technology, Weihai, China, in 2016, and the Ph.D. degree in computer science and engineering from Nanyang Technological University, Singapore, in 2020. He was a Research Fellow with Singtel Cognitive and Artificial Intelligence Lab for Enterprise, in 2021. He is currently an Assistant Professor with the College of Control Science and Engineering, Zhejiang University, Hangzhou, China. His research interests include IoT, industrial IoT, edge computing, and low-power wide area network.



Shibo He (Senior Member, IEEE) received the Ph.D. degree in control science and engineering from Zhejiang University, Hangzhou, China, in 2012. From November 2010 to November 2011, he was a Visiting Scholar with the University of Waterloo, Waterloo, ON, Canada. From March 2014 to May 2014, he was an Associate Research Scientist and from May 2012 to February 2014, a Postdoctoral Scholar with Arizona State University, Tempe, AZ, USA. He is currently a Professor with Zhejiang University. His research interests include Internet of Things, crowdsensing, and Big Data analysis.

Prof. He is on the editorial board of the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, Peer-to-Peer Networking and Applications, and KSII Transactions on Internet and Information Systems. He is also the Guest Editor of Computer Communications and International Journal of Distributed Sensor Networks. He was a Symposium Co-Chair of the IEEE GlobeCom 2020 and IEEE ICC 2017, a TPC Co-Chair of the i-Span 2018, a Finance and Registration Chair of the ACM MobiHoc 2015, a TPC Co-Chair of the IEEE ScalCom 2014, a TPC Vice Co-Chair of the ANT 2013'lC2014, a Track Co-Chair of the Pervasive Algorithms, Protocols, and Networks of EUSPN 2013, a Web Co-Chair of the IEEE MASS 2013, and a Publicity Co-Chair of the IEEE WiSARN 2010 and FCN 2014.