# An economic cost-saving strategy based on a hybrid 4G-5G offload architecture in vehicular edge computing

Lei Shi, Fei Zhao*, Shuangliang Zhao, Zengwei Lyu and Yang Zhang

School of Computer Science and Information Engineering,
Hefei University of Technology,
Hefei, 230009, China
and
Engineering Research Centre of Safety Critical Industrial Measurement and Control Technology,
Ministry of Education,
Hefei, 230009, China
and
Hefei Origin IoT Technology Co., Ltd.,
A-Region of the Intelligent Technology Park North to the Crossing of
Susong Road and Guanhai Road,
Hefei, China
Email: shilei@hfut.edu.cn
Email: hfut_zf@163.com
Email: 15345686808@163.com
Email: lzw@hfut.edu.cn
Email: rambo.zhang@ori-iot.com
*Corresponding author

**Abstract:** The excellent transmission performance of 5G provides reliable support for vehicular edge computing (VEC). However, due to the drawbacks of small coverage area and high energy cost of 5G base stations (BSs), long-term usage will bring huge economic costs to service operators. In this paper, we design a new 4G-5G hybrid offload architecture for VEC scenarios. On this basis, we first build the mathematical model and find that it cannot be solved directly. Then we design algorithms for offline and online cases respectively. Simulation results show that our online algorithm (ONA) can significantly reduces the operator's economic cost while achieving a high task success rate, and the effect is better than other comparison schemes. For example, when the number of tasks is 705,600, the economic cost is reduced by 8.47%–48.7% compared to AA, RS and GA scheme.

**Keywords:** cost saving; 5G; vehicular edge computing; VEC; task offloading; internet of vehicles; IoV.

**Reference** to this paper should be made as follows: Shi, L., Zhao, F., Zhao, S., Lyu, Z. and Zhang, Y. (2023) 'An economic cost-saving strategy based on a hybrid 4G-5G offload architecture in vehicular edge computing', *Int. J. Sensor Networks*, Vol. 42, No. 4, pp.227–237.

**Biographical notes:** Lei Shi received his BS in 2002, MS in 2005, and PhD in 2012, all in Computer Science from the Hefei University of Technology, Hefei, Anhui, China. He is currently an Associate Professor in the School of Computer and Information, Hefei University of Technology. His main research area lies in edge computing and wireless network optimisation.

Fei Zhao is currently pursuing his Master's degree in the School of Computer Science and Information Engineering, Hefei University of Technology, Hefei, Anhui, China. His research interests include internet of vehicles and edge computing.

Shuangliang Zhao is currently pursuing his Master's degree in the School of Computer Science and Information Engineering, Hefei University of Technology, Hefei, Anhui, China, His research interests include internet of vehicles and edge computing.

Zengwei Lyu received his BS, MS and PhD from the School of Computer Science and Information Engineering, Hefei University of Technology, Hefei, China, in 2012, 2015, and 2019, respectively. He is currently an Associate Professor and a MSc Supervisor with the School of Computer Science and Information Engineering, Hefei University of Technology. His research interests include wireless sensor network and deep learning.

Yang Zhang is the Leader of Hefei Origin IoT Technology Co., Ltd. His main research area lies in edge computing and wireless network optimisation.

This paper is a revised and expanded version of a paper entitled 'An energy-saving strategy for 5G base stations in vehicular edge computing' presented at CollaborateCom, Hangzhou, China, 15–16 October 2022.

## 1 Introduction

In recent years, the rapid development of vehicle technology and wireless communication has enabled the modern vehicles to be more intelligent. Many new vehicle applications are emerging, such as autonomous driving and on-board infotainment services (**??**). These applications all require intensive real-time computation. However, due to limited computing resources, the local computing unit of the vehicle is often unable to meet the computing demands of such applications. To overcome this limitation, vehicular edge computing (VEC) has been proposed to address this problem. As an emerging and recognised promising paradigm, VEC can provide fast computing services for vehicle users (**???**). Specifically, through vehicle-to-infrastructure (V2I) communications, resource-constrained vehicles can offload their delay-sensitive computation-intensive tasks to 5G base stations (BSs) configured with edge servers for processing (Cai et al., 2019; **?**; Cai and Shi, 2021). For example, vehicles can transmit traffic information within the current perception range to the edge server, and the server makes a global judgment on the information and returns the result to the vehicle as a reference. In addition, compared with cloud computing, the edge server in VEC is closer to the vehicles, so VEC can achieve lower data transmission delay (**??**), and vehicle users can get better quality of service (QoS) (**?**).

However, in order to support the high density of vehicle users in cities, the transportation system needs to deploy 5G BSs densely on the roadside. At present, 5G BSs have the disadvantages of high energy consumption and small coverage area (Cheng et al., 2017), long-term usage will bring huge economic costs to service operators. Therefore, how to reduce the economic cost of operators providing VEC services deserves investigation.

Previous research has made some contributions to reducing the economic cost of VEC service operators (**???**). In these works, they focus on the energy consumed during task offloading and computing, and save costs by making reasonable resource allocation and task offloading decisions. In fact, there is a huge waste of energy in the low-traffic time period, which needs more attention (Auer, 2011). It is not necessary to keep 5G BSs active all the time. We can save cost by switching 5G BSs to sleep status during low-traffic periods. However, switching 5G BSs to sleep status raises a new problem. The 5G BSs in VEC are arranged along roads and each segment of the road is served by only one 5G BS. When a 5G BS switches to sleep state in low-traffic periods, vehicles within its coverage will not be able to offload tasks. In other words, the success rate of the tasks will be greatly reduced. This makes it very demanding for 5G BSs to sleep without affecting vehicle users.

Motivated by the aforementioned discussion, we aim to give a new and more suitable solution. In this paper, we further investigate the problem of minimising the economic cost of VEC service operators. The contributions of this paper are summarised as follows.

- We propose a new 4G-5G hybrid offload architecture for VEC scenarios, in which 5G BS can switch to sleep state during low-traffic periods, and tasks generated in the corresponding area will be offloaded to 4G BS for processing. Of course, this will not break the delay constraint of the tasks during low traffic time periods.
- We established a mathematical model and discussed in detail from three aspects of communication, calculation and cost. Furthermore, our work is among the few efforts to consider BS switching cost in the formulation of the problem.
- We design the heuristic algorithm for the offline case and the online case, respectively. Through experimental simulations, it is confirmed that our scheme significantly reduces the economic cost of VEC service operators while achieving a high task success rate.

The rest of this paper is organised as follows. In Section 2, the related works are introduced. In Section 3, the system model is presented, including communication model, computation model and cost model. In Section 4, our offline and online algorithms are described. Detailed simulation results and conclusions of the paper are given in Sections 5 and 6, respectively.

## 2 Related works

There has been a lot of studies on the cost optimisation for VEC, mainly focused on two aspects, one is the optimisation of terminal vehicle cost, and the other is the optimisation of infrastructure cost such as roadside BSs.

First, we give a brief introduction to the research on terminal vehicle cost optimisation. **?** propose a multi-device and multi-server task joint task offloading game (JTOG) algorithm in order to minimise the energy consumption for all vehicular terminal devices generating tasks. **?** jointly optimise the offloading proportion and uplink/computation/downlink bit allocation of multiple vehicles, for the purpose of minimising the total energy consumption of the vehicles under the delay constraint. **?** jointly optimise the latency and cost by considering both offloading decisions, communication and computational resource allocation.

Next, we introduce research on roadside infrastructure such as BSs. **?** save infrastructure costs by using coherent beamforming techniques to reduce the density of 5G BS placement at the roadside. They designed a heuristic algorithm for the iterative coherent beamforming node design (ICBND) algorithm to obtain the approximate optimal solution. And they significantly reduce the cost of communication network infrastructure. **?** propose a sleep model for BSs in cellular networks and investigates the benefits of turning off a portion of BSs during low traffic. In the article, the authors propose a simple analytical model that determines the optimal BS shutdown time based on daily traffic patterns. However, in that paper the authors consider only one switchover for the BS, and the effect of this switchover on reducing the energy consumption and operating costs of the BS is relatively small. **?** proposed a simple and effective algorithm to save costs. The main idea of the algorithm is to let BSs cycle alternately between ON and OFF states, and adjust the alternating cycle appropriately according to the vehicle request. Chavarria-Reyes et al. (2015) propose an efficient algorithm to minimise the energy cost by jointing the cell association and on-off scheme. **?** optimise the task latency while allowing the candidate BSs to randomly switch states between sleep and work to save cost. **?** minimise energy cost by forcing idle BSs to sleep or dynamically adjusting the signal range of BSs through a software-defined network, considering connectivity, communication, and power perspectives, respectively. **?** consider the scenario where multiple mobile users share multiple heterogeneous edge servers and propose an approximation algorithm to minimise the energy cost of the MEC system. **?** consider optimising the quality of user experience under a long-term energy budget constraint.

However, some of the above-mentioned studies only focus on the cost of task offloading and computing, while others do not take into account the impact of switching the BS to the sleep state on the task success rate. In this paper, we propose a new 4G-5G hybrid offload architecture for VEC scenarios based on existing work. It combines the advantages of 4G BSs and 5G BSs. Furthermore, we design the heuristic algorithm for the offline case and the online case, respectively. Through experimental simulations, it is confirmed that our scheme significantly reduces the economic cost of VEC service operators while achieving a high task success rate.

## 3 System model

In this section, we introduce the complete process of tasks being processed under the 4G-5G hybrid offload architecture and formulate the optimisation problem.

**Figure 1** 4G-5G hybrid task offloading framework in VEC (see online version for colours)

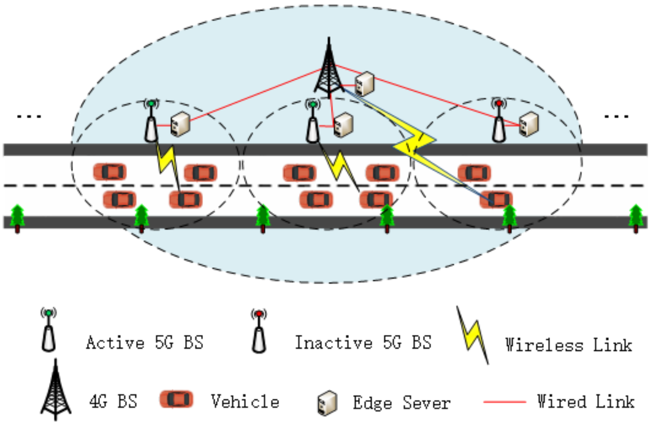


**Table 1** Major notations

| *Notations* | *Descriptions* |
|---|---|
| $D_j$ | Data size of task $l_j$ |
| $C_j$ | Processing density of task $l_j$ |
| $T_j^{\max}$ | Maximum tolerable delay of task $l_j$ |
| $R^{4G}, R^{5G}$ | Maximum transmission rate between vehicle and 4G BS/5G BS |
| $k^{4G}(t_j)$ | Number of tasks transmitted to the 4G BS at time slot $t_j$ |
| $k_i^{5G}(t_j)$ | Number of tasks transmitted to 5G BS $s_i$ at time slot $t_j$ |
| $\mu_j$ | Offloading decision of task $l_j$ |
| $r_j$ | Transmission rate of task $l_j$ |
| $f$ | Computing capability of edge server |
| $E^a, E^{ua}$ | Static energy consumed by an active/inactive 5G BS in a time slot |
| $E^{4G}$ | Static energy consumed by the 4G BS in a time slot |
| $x_i^{on}, x_i^{off}$ | Number of times the 5G BS $s_i$ is turned on/off in time period $T$ |
| $E^{on}, E^{off}$ | Energy required for a 5G BS to turn on/off once |
| $\alpha_i(t)$ | State of 5G BS $si$ at time slot $t$ |
| $\zeta$ | Dynamic energy consumed by a BS to process a unit of task data |
| $\rho$ | Money corresponding to each unit of energy consumed by the BSs |

Consider a scenario consisting of multiple vehicles and multiple BSs (including one 4G BS and several 5G BSs), as shown in Figure 1.

Suppose the whole road is divided into many segments, each segment is covered by one 5G BS and the whole road can be covered by the 4G BS. Suppose these BSs are connected by wired links, so the communication time among them can be ignored. Suppose each BS is equipped with an edge server with the same computing capability. Suppose that during the entire scheduling time, there will be vehicles passing the road and need to offload tasks to the BS for processing.

However, since the 5G BS energy consuming is high, long-term usage will bring huge economic costs to service operators. We aim to minimise the economic cost of the operator while ensuring the success of the tasks as much as possible. We want to achieve the optimisation goal by appropriately 'turning on/off' 5G BSs and reasonably determining the offloading method of each task.

The main notations used in the following discussion are listed in Table 1.

### 3.1 Communication model

We first discuss the communication model. Suppose the whole scheduling time $T$ can be divided into $h$ time slots $\tau$ equally and we normalise $\tau = 1$. Suppose that in the scheduling time period $T$, there are $m$ vehicles passing by and generating tasks, and each vehicle generates only one task. Suppose tasks generated at any time slot can be completed before the next time slot. Suppose tasks received by the same BS in the same time slot have the same transmission rate after the communication resources are allocated. Denote $t$ $(t \in T,\ 1 \le t \le h)$ as a time slot. Denote $N$ as the set of all 5G BSs, and $s_i$ $(s_i \in N,\ 1 \le i \le n)$ as a 5G BS. Denote $L$ as a set of all tasks, and $l_j$ $(l_j \in L,\ 1 \le j \le m)$ as a task. We describe the task with three attributes. One, the task data size, and we denote it as $D_j$. Two, the processing density of the task (in CPU cycles/bit), which can be multiplied by the data size of the task to obtain the computing resources required by the task, and we denote it as $C_j$. Three, the maximum tolerable delay for tasks, and we denote it as $T_j^{\max}$.

Suppose that all tasks have the same attributes, that is, the data size, processing density and maximum tolerable delay of all tasks are the same. For each task, it can only be transmitted to one BS, and we use a binary variable $\mu_j$ to indicate the offloading result for task $l_j$, then we have

$$\mu_j = \begin{cases} 1\, l_j \text{ is transmitted to the 4G BS;} \\ 0\, l_j \text{ is transmitted to one 5G BS.} \end{cases} \tag{1}$$

Then when $l_j$ is transmitted to the 4G BS, we can get the data transmission rate as

$$r_j^{4G} = \frac{R^{4G}}{k^{4G}(t_j)}, \tag{2}$$

where $R^{4G}$ is the maximum transmission rate between a vehicle and the 4G BS, $t_j$ is the time slot when $l_j$ is generated, and $k^{4G}(t_j)$ is the number of tasks transmitted to the 4G BS at time slot $t_j$. We can similarly get the data transmission rate between the corresponding vehicle and the 5G BS $s_i$ as

$$r_{i,j}^{5G} = \frac{R^{5G}}{k_i^{5G}(t_j)}, \tag{3}$$

where $R^{5G}$ is the maximum transmission rate between a vehicle and the 5G BS, $k_i^{5G}(t_j)$ is the number of tasks transmitted to $s_i$ at time slot $t_j$. We use $\beta_{i,j}(t)$ to indicate whether the vehicle generating $l_j$ is within the coverage of $s_i$ at time slot $t$, then we have

$$\beta_{i,j}(t) = \begin{cases} 1 \text{ the vehicle generating } l_j \text{ is within} \\ \quad s_i\text{'s coverage at time slot } t; \\ 0 \text{ otherwise.} \end{cases} \tag{4}$$

Therefore, the transmission rate of $l_j$ can be expressed as

$$r_j = \begin{cases} r_j^{4G} \text{ if } \mu_j = 1; \\ r_{i,j}^{5G} \text{ if } \mu_j = 0,\ \beta_{i,j}(t_j) = 1. \end{cases} \tag{5}$$

### 3.2 Computation model

In this subsection we continue to discuss the computation model. The total task delay includes the transmission delay, the waiting delay and the calculation delay. Then for task $l_j$, we have

$$T_j^{total} = T_j^{trans} + T_j^{w} + T_j^{comp}, \tag{6}$$

where $T_j^{trans}$, $T_j^{w}$ and $T_j^{comp}$ correspond to the transmission delay, the waiting delay and the calculation delay for $l_j$ respectively. In the following, we will give the specific formula for each component.

For the transmission delay of task $l_j$, it mainly depends on the task data size that needs to be transmitted and the offloading decision of vehicle $- j$. Based on the task transmission rate we obtained in the previous section, the transmission delay of $l_j$ can be expressed as

$$T_j^{trans} = \frac{D_j}{r_j}, \tag{7}$$

For the waiting delay of task $l_j$, it mainly depends on the load condition of the edge server. We assume that edge servers use non-preemptive CPU allocation and allocate computing resources to one task at a time until the task is completed. When $l_j$ arrives at the task queue of the edge server, if there are no other tasks in front of it, it will be calculated immediately, that is, its waiting delay is 0. On the contrary, if there are other tasks waiting to be calculated or being calculated in front of $l_j$, then it needs to wait for its previous task to be calculated. We use $\gamma_{j'}(l_j)$ to indicate whether $l_{j'}$ is the previous task in the task queue of $l_j$, then we have

$$\gamma_{j'}(l_j) = \begin{cases} 1\, l_{j'} \text{ is the previous task in the task} \\ \quad \text{queue of } l_j; \\ 0 \text{ otherwise.} \end{cases} \tag{8}$$

Thus the waiting delay required for the part of task $l_j$ to be processed by the edge server can be expressed as

$$T_j^{w} = \max\left\{ \sum_{j'=1}^{m} \gamma_{j'}(l_j) \cdot \left( (t_{j'} + T_{j'}^{trans} + T_{j'}^{w} + \frac{D_{j'} \cdot C_{j'}}{f} \right) - (t_j + T_j^{trans})), 0 \right\}, \tag{9}$$

where $f$ is the computing capability of an edge server.

For the calculation delay of task $l_j$, it is mainly related to the computing resource requirements of $l_j$ and the computing capability of the edge server. Therefore, the calculation delay required by the data computed by the edge server can be expressed as

$$T_j^{comp} = \frac{D_j \cdot C_j}{f}. \tag{10}$$

Based on the above discussion, we can get the total delay of $l_j$.

### 3.3 Cost model

In this subsection we continue to discuss the cost model. We assume that the economic cost of the service operator is positively related to the energy consumed by the BSs. We divide the total energy consumption of BSs into three parts, including static energy consumption(energy consumption of power transmission and cooling, etc.), load-related dynamic energy consumption and state-switching energy consumption (**?**). Denote $E^{total}$ as the total energy consumed by all BSs in time period $T$. Therefore $E^{total}$ can be expressed as

$$E^{total} = E^s + E^d + E^{switch}, \tag{11}$$

where $E^s$ and $E^d$ are the static energy and dynamic energy consumed by all BSs in the time period $T$, respectively. $E^{switch}$ is the state-switching energy consumed by all BSs in the time period $T$. In the following, we will give the specific formula for each component.

For the first item $E^s$, we use a binary variable $\alpha_i(t)$ to indicate the state of 5G BS $s_i$ at time slot $t$, then we have

$$\alpha_i(t) = \begin{cases} 1\ s_i \text{ is active at time slot } t; \\ 0 \text{ otherwise.} \end{cases} \tag{12}$$

Thus $E^s$ can be expressed as

$$\begin{aligned} E^s = &\sum_{t=1}^{h}\sum_{i=1}^{n}(\alpha_i(t) \cdot E^a + (1-\alpha_i(t)) \cdot E^{ua}) \\ &+ h \cdot E^{4G}, \end{aligned} \tag{13}$$

where $E^a$ and $E^{ua}$ are the static energy that a 5G BS needs to consume when it is active and inactive in a time slot, respectively. And $E^{4G}$ represents the static energy consumed by the 4G BS in a time slot.

For the second item $E^d$, it is mainly related to the computing resources required to process the tasks. Then $E^d$ can be expressed as

$$E^d = \sum_{j=1}^{m} \zeta \cdot D_j, \tag{14}$$

where $\zeta$ is the dynamic energy consumed by a BS to process a unit of task data.

For the third item $E^{switch}$, it depends on how many times the 5G BSs are 'turned on' and 'off'. We use $x_i^{on}$ and $x_i^{off}$ to represent the number of times that $s_i$ is turn on and off in time $T$ respectively. We have

$$x_i^{on} = \sum_{t=1}^{h} \max\{(\alpha_i(t) - \alpha_i(t-1)), 0\}, \tag{15}$$

$$x_i^{off} = \sum_{t=1}^{h} \max\{(\alpha_i(t-1) - \alpha_i(t)), 0\}. \tag{16}$$

Then $E^{switch}$ can be expressed as

$$E^{switch} = \sum_{i=1}^{n}(x_i^{on} \cdot E^{on} + x_i^{off} \cdot E^{off}), \tag{17}$$

where $E^{on}$ and $E^{off}$ are the energy cost of turning on and off a 5G BS once, respectively. Based on the above discussion, the economic cost of the service operator in the time period $T$ can be expressed as

$$M^{total} = \rho \cdot E^{total}, \tag{18}$$

where $\rho$ represents the money that the operator needs to pay for each unit of energy consumed by the BSs.

### 3.4 Problem formulation

In this paper, we aim to reduce the economic cost of the service operator by minimising the total energy consumption of all BSs, while we guarantee the success of the tasks as much as possible. Based on the above discussion, our optimisation problem can be formulated as follows

$$\min_{\alpha_i(t),\mu_j} \quad M^{total} \tag{19a}$$

$$\text{s.t. } \alpha_i(t) \in \{0,1\}, \tag{19b}$$

$$\mu_j \in \{0,1\}, \tag{19c}$$

$$\forall i \in [1,n], \tag{19d}$$

$$\forall t \in [1,h], \tag{19e}$$

$$\forall j \in [1,m], \tag{19f}$$

$$T_j^{total} \le T_j^{\max}. \tag{19g}$$

The variables in the optimisation problem are $\mu_j$ and $\alpha_i(t)$, which almost appear in all items with different forms. $T$, $h$, $m$, and other symbols are all constants or determinable values. In real scenarios, we may only know the tasks that have been generated or being generated, and the results of the decisions that have been made are also irreversible. So the values of $\alpha_i(t)$ and $\mu_j$ are difficult to be solved directly. We will try to get an approximate optimal solution in the next section.

## 4 Algorithms

In Section 3, we give the original problem model and show it is difficult to be solved directly. In this section, we will try to find a feasible solution for the problem. First, we design an offline algorithm. In the offline algorithm, we suppose that we know the total number of tasks and the time slot in which any task is generated. Then we design an online algorithm. In the online algorithm, we only know tasks that have been or are being generated, while tasks that will be generated are not known. This means that we need to dynamically change the state of the 5G BSs and the way for transmitted tasks based on the situation of past and current time slots. In the following, we first discuss the offline algorithm in Subsection 4.1. Then we discuss the online algorithm in Subsection 4.2.

### *4.1 Offline strategy*

For the offline strategy, we assume that we know the corresponding location and time slot when any task is generated. Based on this information, we will first determine cases on where tasks transmitted to the 4G BS cannot satisfy the delay constraint. Then, we further determine cases on where 5G BSs should be in sleep. Then main idea for the offline strategy can be summarised into four steps as the following.

**Algorithm 1** Offline algorithm (OFFA)

```
Input: $L$: the task set; $t_j$: the time slot when any task $l_j$
    is generated; $\beta_{i,j}(t_j)$: the location when any task $l_j$ is
    generated;
Output: $M^{total}$
 1: for 5G BS $s_i \in N$ do
 2:     for Task $l_j \in L$ $(\beta_{i,j}(t_j) = 1)$ do
 3:         Calculate $T_j^{total}$ according to formula (6) under the
            condition that $l_j$ is transmitted to the 4G BS;
 4:         if $(T_j^{total} \leq T_j^{\max})$ then
 5:             $\alpha_i(t_j) = 0$, $\mu_j = 1$;
 6:         else
 7:             $\alpha_i(t_j) = 1$, $\mu_j = 0$;
 8:         end if
 9:     end for
10:     Get all values of $\alpha_i(t)$ $(\forall t \in T)$, select all periods when
        $\alpha_i(t)$ is equal to 0 continuously and get the set $\mathbb{T}$;
11:     for $T_q^{ua} \in \mathbb{T}$ do
12:         if $(T_q^{ua} < T_{on} + T_{off} || (E^a - E^{ua}) \cdot T_q^{ua} \leq E^{on} + E^{off})$
13:             $\alpha_i(t) = 1 (t \in T_q^{ua})$;
14:             Remove $T_q^{ua}$ from $\mathbb{T}$;
15:         end if
16:     end for
17: end for
18: Calculate the economic cost $M^{total}$;
```

Step one, we first randomly select a 5G BS $s_i$ and pick out all tasks generated within its coverage. When no task is generated within the coverage of $s_i$ in a time slot, the state of $s_i$ in this time slot is tentatively set as inactive. For a task generated within range of $s_i$, we try to transmit it to the 4G BS and calculate its total delay. If the total delay can satisfy the delay constraint, the state of $s_i$ in this time slot is tentatively set as inactive too. Otherwise, let $s_i$ be active at this time slot and let generated tasks at this time slot transmit to $s_i$.

Step two, according to the state information corresponding to each time slot $s_i$ obtained in the first step, a series of time periods consisting of adjacent time slots in which $s_i$ is in an inactive state are screened and obtained. Denote the set of these time periods as $\mathbb{T}$, and denote $T_q^{ua}$ ($T_q^{ua} \in \mathbb{T}$, $q = 1, ...$) as one of the time periods.

Step three, judge if $s_i$ can be switched into the sleep state in time period $T_q^{ua}$. We notice that if two conditions are satisfied then $s_i$ can be switched into the sleep state. First, $T_q^{ua} \geqslant T^{on} + T^{off}$, where $T^{on}$ and $T^{off}$ are the time required to turn on and off a 5G BS once, respectively. Second, $(E^a - E^{ua}) \cdot T_q^{ua} > E^{on} + E^{off}$. When both conditions $T_q^{ua}$ are satisfied, it is reasonable and can reduce energy consumption for $s_i$ to switch to sleep state at time period $T_q^{ua}$. Judge the rest of the time period in $\mathbb{T}$ like this.

Step four, repeat the above operation for all other 5G BSs.

Based on these discussions, we can get the offline algorithm (OFFA) as shown in Algorithm 1. Step one corresponds to lines 1–9 of the pseudocode in Algorithm 1, step two corresponds to line 10 of the pseudocode, and step three corresponds to lines 11–16 of the pseudocode.

### *4.2 Online strategy*

For the online strategy, we only know tasks that have been generated and are being generated. We should make strategies based on this information in real-time. In this subsection, we will first discuss the case where we need to increase the active 5G BSs, and then we will discuss the case where we need to decrease the active 5G BSs. After that, we will give the steps of the online algorithm.

First, we determine the situation where we need to increase an active 5G BS. Suppose that vehicles are evenly distributed on the road. Since the coverage area of each 5G BS is the same, the number of vehicles in each area can be regarded as the same. Since each vehicle will generate a task within the scheduling time period $T$, the task will be generated with equal probability within the coverage of each 5G BS. Denote $k(t)$ as the total number of tasks generated at time slot $t$. Denote $k^{\max}$ as the maximum number of tasks that can be transmitted to the 4G BS at the same time slot while satisfying the time delay constraint. Then we have $\frac{k^{\max} \cdot D_j}{R^{4G}} + \frac{k^{\max} \cdot D_j \cdot C_j}{f} \leq T_j^{\max}$. When both sides of the formula are equal, we can get $k^{\max} = \lfloor \frac{T_j^{\max} \cdot R^{4G} \cdot f}{D_j \cdot f + R^{4G} \cdot D_j \cdot C_j} \rfloor$. Denote $s(t)$ as the number of 5G BSs in sleep state at time slot $t$. Denote $a(t)$ as the number of 5G BSs in active state at time slot $t$. So we can use $\frac{s(t)}{n} \cdot k(t)$ to approximate the number of tasks transmitted to the 4G BS at time slot $t$. We notice that if $\frac{s(t)}{n} \cdot k(t) > k^{\max}$, the number of currently active 5G BSs is not enough to match the number of tasks. Therefore, when $s(t) > 0$ and $k(t) > \frac{n}{s(t)} \cdot k^{\max}$, we turn on a 5G BS in a sleeping state.

Second, we determine the situation where we need to decrease an active 5G BS. We notice that two conditions need to be met. First, similar to the last paragraph, $a(t) > 0$ and $k(t) < \frac{n}{n - a(t) + 1} \cdot k^{\max}$. Because we need to ensure that after an active 5G BS is turned off, all tasks generated at the current time slot still meet the delay constraint. Second, the first condition has been maintained for a period of time, which is at least the shortest time that a 5G BS is worth sleeping. In this case it is reasonable to assume that the decrease in the number of tasks is not episodic. From the offline algorithm we can obtain the minimum time that a 5G BS is worth sleeping is $\frac{E^{on}+E^{off}}{E^{a}-E^{ua}}$. When both of these conditions hold, we turn off an active 5G BS.

So the main idea for the online strategy can be summarised into three steps as the following.

Step one, we first initialise all 5G BSs to be in sleep state.

Step two, for time slot $t = 1$, tasks generated within the range of an active 5G BS are transmitted to the corresponding 5G BS, and tasks generated within the range of an inactive 5G BS are transmitted to the 4G BS. Determines whether the number of 5G BSs currently active matches the number of current generated tasks. If the status of the current time slot meets the condition of increase an active 5G BS, then turn on a 5G BS in a sleeping state. If the status of the current time slot meets the condition of decrease an active 5G BS, then turn off an active 5G BS. Otherwise, all 5G BSs remain in their current state.

Step three, repeat the above judgment operation until $t = h$.

Based on these discussions, we can get the online algorithm (ONA) as shown in Algorithm 2.

**Algorithm 2** Online algorithm (ONA)

```
Initialisation;
Initialise all 5G BSs to sleep state;
End Initialisation;
for Time slot $t \in [1, h]$ do
    for $s_i \in N$ do
        if $(\alpha_i(t) = 0)$ then
            $\mu_j = 1(\beta_{i,j}(t_j) = 1, t_j = t)$;
        else
            $\mu_j = 0(\beta_{i,j}(t_j) = 1, t_j = t)$;
        end if
    end for
    if $(s(t) > 0$ && $k(t) > \frac{n}{s(t)} \cdot k^{\max})$ then
        Turn on a 5G BS;
    else if $(a(t) > 0$ && all values from $k(t - \frac{E^{on}+E^{off}}{E^{a}-E^{ua}})$ to
  $k(t)$ are less than $\frac{n}{n-a(t)+1} \cdot k^{\max})$ then
        Turn off a 5G BS;
    else
        all 5G BSs remain in their current state;
    end if
end for
Calculate the economic cost $M^{total}$;
```

## 5 Simulation results

In this section, we conduct simulations and present representative numerical results to evaluate the performance of the proposed online algorithm. We first describe the simulation setup and then discuss the simulation results.

In the simulation, we consider a one-way road with a length of 800 metres. A 4G BS is deployed in the middle of the roadside, and its coverage radius is 400 metres. Since the coverage of 5G BS is generally 200–300 metres, we set the number of 5G BSs to 3, that is, $n$ = 3. We set the length of a time slot $\tau$ = 1 s. The detailed parameters setting about tasks and BSs is shown in Table 2.

### *5.1 Simulation setup*

**Table 2** Parameter settings

| *Description* | *Value* |
|---|---|
| Computing capability of edge server ($f$) | 300 M CPU/s |
| Maximum transmission rate between vehicle and 4G BS/5G BS ($R^{4G}, R^{5G}$) | 10 Mbit/s,<br>80 Mbit/s |
| Static energy consumed by an active/inactive 5G BS in one time slot ($E^{a}, E^{ua}$) | 3 kJ, 0.5 kJ |
| Switching time of 5G BS ($T^{on}$,$T^{off}$) | 5 s, 5 s |
| Switching energy of 5G BS ($E^{on}, E^{off}$) | 40 kJ, 40 kJ |
| Data size of task ($D_j$) | {0.2∼1 Mbit} |
| Processing density of task $l_j$ ($C_j$) | {10∼20 M CPU cycles/Mbit} |
| Maximum delay of task ($T_j^{\max}$) | 1 s |
| Static energy consumed by the 4G BS in a time slot ($E^{4G}$) | 1 kJ |
| Efficiency of dynamic energy consumption of BS ($\zeta$) | 0.1 kJ/Mbit |
| Money corresponding to each unit of energy consumed by the BSs ($\rho$) | $1.39 * 10^{-4}$ |
| Number of tasks ($m$) | 441,000∼882,000 |

### *5.2 Simulation results*

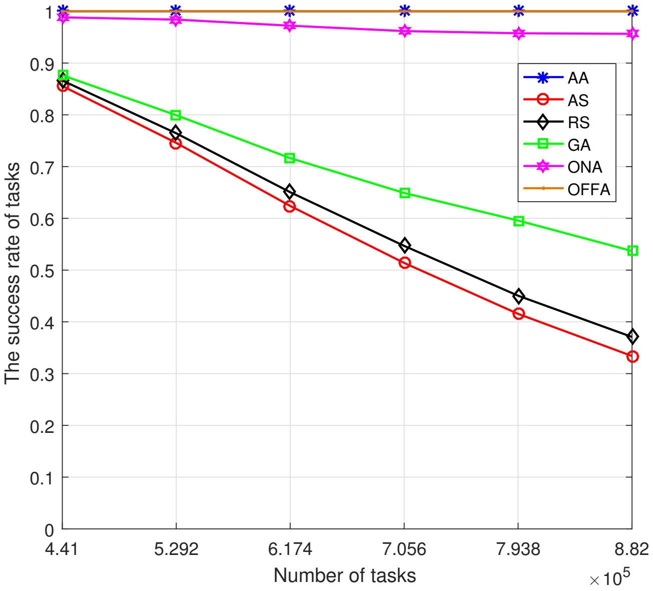
We consider the following schemes as benchmarks to evaluate our proposed ONA scheme.

- *Always-active (AA):* Where all 5G BSs are always in active state.
- *Always-sleep (AS):* Where all 5G BSs are always in sleeping state.
- *Random-switch (RS):* Where all 5G BSs are turned on and off randomly.
- *Greedy-algorithm (GA):* When reducing an active 5G BS, only the first condition corresponding to the online algorithm needs to be satisfied, and other parts are the same as the online algorithm.

We first evaluate the performance of the proposed ONA scheme in terms of task success rate. In our experiments, we set $T$ = 86,400 s, which means that the scheduling time in our simulation is a whole day consisting of 86,400 time slots. We use a traffic flow dataset from a freeway near Heathrow Airport in the UK as reference, and generate tasks in proportion to the number of vehicles in the corresponding time period. The task succeeds when the total delay of the task is less than or equal to the maximum tolerable delay, otherwise the task fails. When a task fails, that is, when the task is not completed within the maximum tolerable delay, the corresponding edge server will stop computing the task and directly calculate the next task in the task queue.

Figure 2 shows the relationship between the task success rate and the number of tasks when we set $D_j$ = 0.8 M, $C_j$ = 12.5 M CPU cycles/Mbit. It can be seen that task success rate of all schemes decrease with the increasing of the number of tasks except the AA scheme and the OFFA scheme. Because an increase in the total number of tasks equates to faster task generation. This will increase the burden on the BS, strain communication and computing resources, and ultimately increase the possibility that the total task delay exceeds the maximum tolerable delay. In contrast, except for the AA scheme and the OFFA scheme, the effectiveness of our ONA scheme is better than other schemes.

**Figure 2** The success rate of tasks under different number of tasks (see online version for colours)



Then, we evaluate the performance of the proposed ONA scheme in reducing economic cost of the service operator. Figure 3 shows the relationship between the economic cost of the service operator and the number of tasks when we set $D_j$ = 0.8 M, $C_j$ = 12.5 M CPU cycles/Mbit. It can be seen that the economic cost of all schemes increases with the number of tasks. In the RS scheme, since all 5G BSs are switched randomly in any case, the energy consumption does not change much. However, due to the frequent state switching of 5G BSs under this scheme, a large amount of switching energy consumption will be generated (see Figure 4), which leads to the highest economic cost compared with other schemes. In the AS scheme, since all 5G BSs are in a sleep state in any case, the economic cost required is the lowest. However, it can be seen from the Figure 2 that the success rate of tasks under the AS scheme is very low. Therefore, this scheme is not effective. In the GA scheme, the switching of 5G BSs is greatly affected by the occasional fluctuation of traffic flow, which results in many very short sleep periods that are not worthy of sleep for 5G BSs. This increases energy consumption and causes many tasks to fail due to wrong offload decisions. Therefore, the performance of the GA scheme is not good enough in terms of improving task success rate and reducing the economic cost. In contrast, our ONA scheme performs well both in terms of improving task success rate and reducing economic cost. Under the condition of the same task success rate, the ONA scheme is slightly weaker than the OFFA scheme in terms of reducing economic costs, but it is significantly better than other schemes. For example, when the number of tasks is 705,600, the economic cost is reduced by 8.47%–48.7% compared to AA, RS and GA scheme.

**Figure 3** The economic cost of the service operator under different number of tasks (see online version for colours)

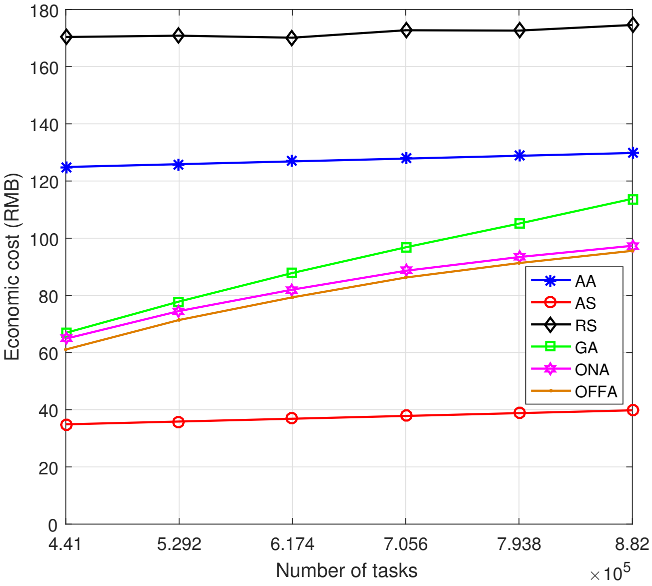


Figure 4 shows the proportion of each energy consumption of the schemes mentioned in this paper when the number of tasks is 705,600. We can intuitively see the composition of energy consumption in each scheme. It can be seen that, except for the RS scheme and the GA scheme, the static energy consumption of other schemes accounts for the largest part. In the RS scheme and the GA scheme, due to the frequent switching of 5G BSs, switching energy consumption accounts for the largest part.

To provide a more straightforward understanding, we present in Figure 5 the proportion of tasks corresponding to the two offloading decisions of the ONA scheme in Figures 2 and 3. It can be seen that as the total number of tasks increases, the proportion of tasks that are offloaded to the 5G BSs increases. Because when the total number of tasks increases, more tasks need to be offloaded to 5G BSs to meet their delay constraints, so as to improve the success rate of tasks.

**Figure 4** The proportion of each energy consumption under different schemes (see online version for colours)

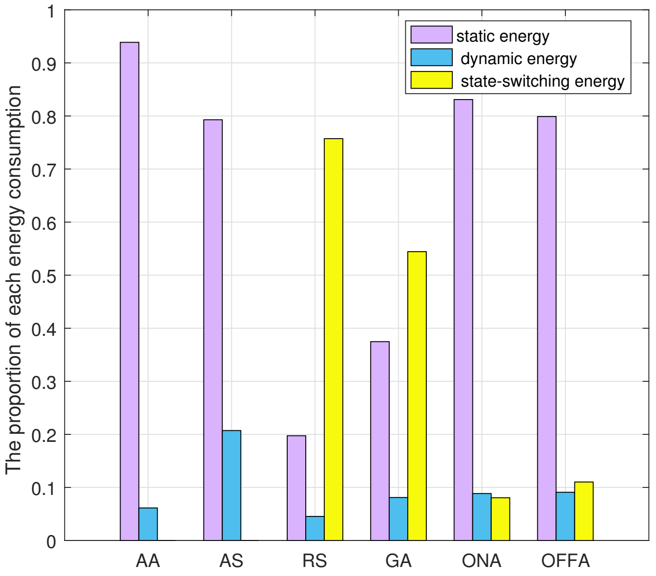


**Figure 5** The proportion of tasks corresponding to two offloading decisions under different total number of tasks (see online version for colours)

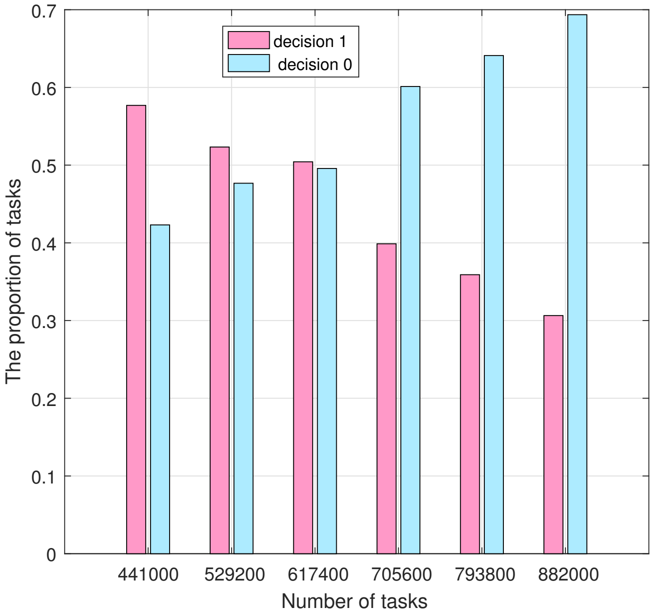


Figure 6 shows the relationship between task success rate and task data size when we set $m$ = 705,600, $C_j$ = 12.5 M CPU cycles/Mbit. It can be seen that, except for the AA scheme and the OFFA scheme, the task success rate of all other schemes decreases as the task data size increases. Because a larger data size requires more communication resources, this increases the probability of task failure. It can be seen that as the task data size increases, the performance of our ONA scheme in improving the task success rate does not drop too much. Compared with other schemes, its advantages are still obvious.

Figure 7 shows the relationship between the economic cost of the service operator and task data size when we set $m$ = 705,600, $C_j$ = 12.5 M CPU cycles/Mbit. It can be seen that the economic cost of all schemes increases with the task data size. Although our ONA scheme performs slightly worse than the GA scheme in reducing the economic cost when the task data size is small, it is more stable and effective in improving the task success rate.

**Figure 6** The success rate of tasks under different task data size (see online version for colours)

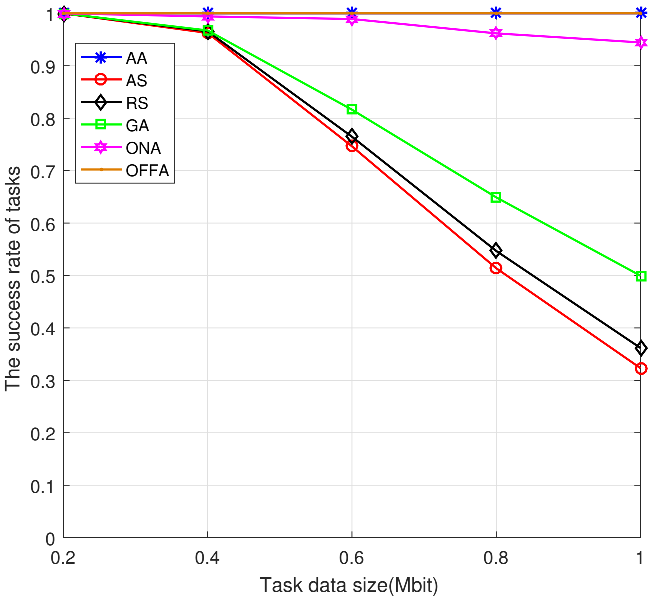


**Figure 7** The economic cost of the service operator under different task data size (see online version for colours)

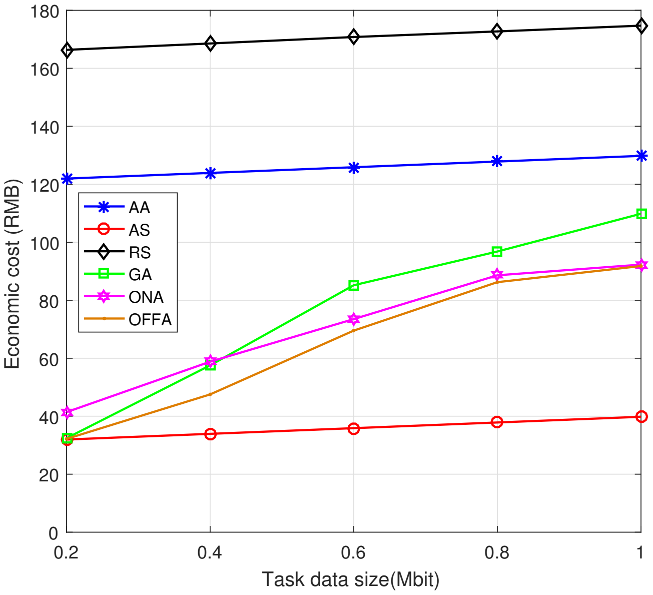


Figure 8 shows the relationship between task success rate and task processing density when we set $m$ = 705,600, $D_j$ = 0.8 M. It can be seen that, except for the AA scheme and the OFFA scheme, the task success rate of all other schemes decreases as the task processing density increases. Because greater task processing density will increase the processing time of the edge server, which increases the probability of task failure. It can be seen that as the task processing density increases, the performance of our ONA scheme in improving the task success rate does not drop too much. Compared with other schemes, its advantages are still obvious.

**Figure 8** The success rate of tasks under different task processing density (see online version for colours)

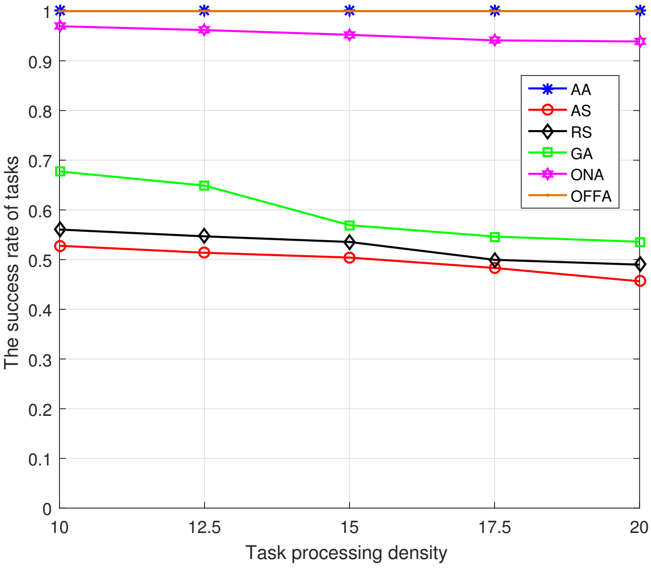


**Figure 9** The economic cost of the service operator under different task processing density (see online version for colours)

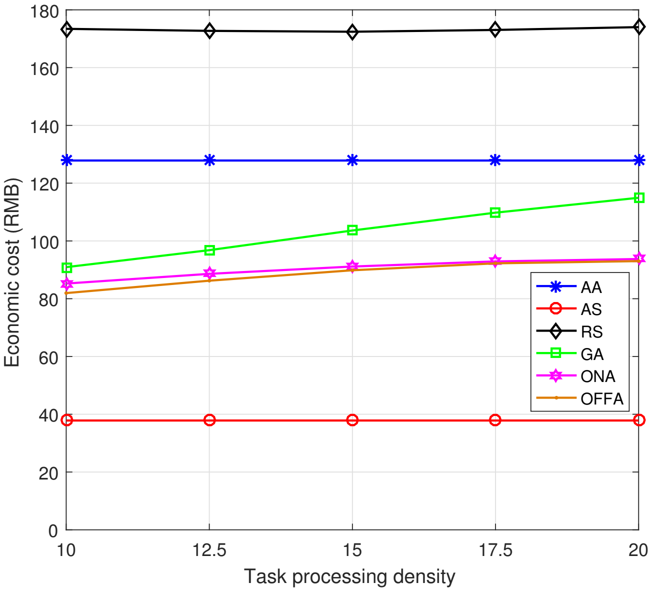


Figure 9 shows the relationship between the economic cost of the service operator and the task processing density when we set $m$ = 705,600, $D_j$ = 0.8 M. It can be seen that our ONA scheme performs very close to the OFFA scheme in reducing the economic cost. Of course, at the same economic cost, the OFFA scheme has a higher task success rate than the ONA scheme, but compared with other schemes, the ONA scheme has obvious advantages.

Based on the above analysis, it can be concluded that our ONA scheme can significantly reduce the economic cost of the service operator while achieving a high task success rate.

## 6 Conclusions

In this paper, we have investigate the problem of minimising the economic cost of VEC service operators, and propose a new 4G-5G hybrid offload architecture that combines the respective advantages of 4G BS and 5G BS. Specifically, we first establish a mathematical model which cannot be solved directly. Then we propose offline algorithms that can be iteratively tuned to achieve 100% success of the task. Considering the real-time requirements of realistic scenarios, we also proposed corresponding online algorithm. Finally, we use a real-world traffic flow dataset to implement the simulation. Simulation results show that our scheme significantly reduces the economic cost while achieving a high task success rate. For example, when the number of tasks is 705,600, the economic cost is reduced by 8.47%–48.7% compared to AA, RS and GA scheme.

In the future, we will further consider the case when there are different types of tasks for offloading. Of course, we also consider the case where tasks are allowed to be partially offloaded.

## Acknowledgements

The work is supported by the Major Science and Technology Projects in Anhui Province, No. 202003a05020009 and the Innovation Foundation of the City of Bengbu, Grant No. JZ2022YDZJ0019, and the Key Technology Research and Development Project of Hefei, No. 2021GJ029.

## References

Auer, G. (2011) 'How much energy is needed to run a wireless network?', *IEEE Wireless Commun.*, Vol. 18, No. 5, pp.40–49.

Cai, Z. and Shi, T. (2021) 'Distributed query processing in the edge-assisted IoT data monitoring system', *IEEE Internet of Things Journal*, Vol. 8, No. 16, pp.12679–12693.

Cai, Z., Zheng, X. and Yu, J. (2019) 'A differential-private framework for urban traffic flows estimation via taxi companies', *IEEE Transactions on Industrial Informatics*, Vol. 15, No. 12, pp.6492–6499.

Chavarria-Reyes, E., Akyildiz, I.F. and Fadel, E. (2015) 'Energy consumption analysis and minimization in multi-layer heterogeneous wireless systems', *IEEE Transactions on Mobile Computing*, Vol. 14, No. 12, pp.2474–2487.

Cheng, X., Chen, C., Zhang, W. and Yang, Y. (2017) '5G-enabled cooperative intelligent vehicular (5GenCIV) framework: when Benz Meets marconi', *IEEE Intelligent Systems*, Vol. 32, No. 3, pp.53–59.

Ciullo, D., Marsan, M.A., Chiaraviglio, L. and Meo, M. (2012) 'Jointly optimizing throughput and cost of IoV based on coherent beamforming and successive interference cancellation technology', *2012 Fourth International Conference on Communications and Electronics (ICCE)*, pp.245–250.

Elsherif, F., Chong, E.K.P. and Kim, J.H. (2019) 'Energy-efficient base station control framework for 5G cellular networks based on Markov decision process', *IEEE Transactions on Vehicular Technology*, Vol. 68, No. 9, pp.9267–9279.

Gao, J., Li, M., Zhao, L. and Shen, X. (2018) 'Contention intensity based distributed coordination for V2V safety message broadcast', *IEEE Transactions on Vehicular Technology*, Vol. 67, No. 12, pp.12288–12301.

Gu, X., Zhang, G. and Cao, Y. (2021) 'Cooperative mobile edge computing-cloud computing in internet of vehicle: architecture and energy-efficient workload allocation', *Transactions on Emerging Telecommunications Technologies*, Vol. 32, No. 8, p.e4095.

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

Jiang, H., Dai, X., Xiao, Z. and Iyengar, A.K. (2022) 'Joint task offloading and resource allocation for energy-constrained mobile edge computing', *IEEE Transactions on Mobile Computing*.

Ke, H., Wang, J., Deng, L., Ge, Y. and Wang, H. (2020) 'Deep reinforcement learning-based adaptive computation offloading for MEC in heterogeneous vehicular networks', *IEEE Transactions on Vehicular Technology*, Vol. 69, No. 7, pp.7916–7929.

Liu, J., Wan, J., Zeng, B., Wang, Q., Song, H. and Qiu, M. (2017) 'A scalable and quick-response software defined vehicular network assisted by mobile edge computing', *IEEE Communications Magazine*, Vol. 55, No. 7, pp.94–100.

Luo, Q., Li, C., Luan, T. and Shi, W. (2021) 'Minimizing the delay and cost of computation offloading for vehicular edge computing', *IEEE Transactions on Services Computing*, Vol. 15, No. 5, pp.2897–2909.

Luo, Q., Li, C., Luan, T.H., Shi, W. and Wu, W. (2021) 'Self-learning based computation offloading for internet of vehicles: model and algorithm', *IEEE Transactions on Wireless Communications*, Vol. 20, No. 9, pp.5913–5925.

Peng, H. and Shen, X. (2020) 'Deep reinforcement learning based resource management for multi-access edge computing in vehicular networks', *IEEE Transactions on Network Science and Engineering*, Vol. 7, No. 4, pp.2416–2428.

Shi, L., Wu, L., Shi, Y., Fan, Y., Xu, J. and Li, Z. (2022a) 'Joint CB and SIC technology to optimize throughput and cost under IoV', *IEEE Transactions on Vehicular Technology*, Vol. 71, No. 58, pp.8689–8701.

Shi, T., Li, Y. and Cai, Z. (2022b) 'To process a large number of concurrent top-k queries towards IoT data on an edge server', *The 42nd IEEE International Conference on Distributed Computing Systems (ICDCS 2022)*, pp.559–569.

Sun, Y., Zhou, S. and Xu, J. (2017) 'EMM: energy-aware mobility management for mobile edge computing in ultra dense networks', *IEEE Journal on Selected Areas in Communications*, Vol. 35, No. 11, pp.2637–2646.

Wen, C. and Zheng, J. (2015) 'An RSU on/off scheduling mechanism for energy efficiency in sparse vehicular networks', *2015 International Conference on Wireless Communications Signal Processing (WCSP)*, pp.1–5.

Wheeb, A.H. (2017) 'Performance analysis of VoIP in wireless networks', *International Journal of Computer Networks and Wireless Communications*, Vol. 7, No. 4, pp.1–5.

Wu, L., Xu, J., Shi, L., Bi, X. and Shi, Y. (2019) 'Jointly optimizing throughput and cost of IoV based on coherent beamforming and successive interference cancellation technology', *The 16th International Conference on Wireless Algorithms, Systems, and Applications (WASA)*, Nanjing, China, 25–27 June, pp.235–243.

Xu, X., Zhang, X., Liu, X., Jiang, J., Qi, L. and Bhuiyan, M.Z.A. (2021) 'Adaptive computation offloading with edge for 5G-envisioned internet of connected vehicles', *IEEE Transactions on Intelligent Transportation Systems*, Vol. 22, No. 8, pp.5213–5222.

Zhang, X. and Debroy, S. (2020) 'Energy efficient task offloading for compute-intensive mobile edge applications', *ICC 2020 – 2020 IEEE International Conference on Communications (ICC)*, pp.1–6.

Zhao, Q. and Gerla, M. (2020) 'Energy efficiency enhancement in 5G mobile wireless networks', *2019 IEEE 20th International Symposium on 'A World of Wireless, Mobile and Multimedia Networks' (WoWMoM)*, pp.1–3.

Zhang, N., Zhang, S., Yang, P., Alhussein, O., Zhuang, W. and Shen, X.S. (2017) 'Software defined space-air-ground integrated vehicular networks: challenges and solutions', *IEEE Communications Magazine*, Vol. 55, No. 7, pp.101–109.

Zhu, T., Shi, T., Li, J., Cai, Z. and Zhou, X. (2019) 'Task scheduling in deadline-aware mobile edge computing systems', *IEEE Internet of Things Journal*, Vol. 6, No. 3, pp.4854–4866.