An Energy-Saving Strategy for 5G Base Stations in Vehicular Edge Computing

Fei Zhao[1,3], Lei Shi[1,3(✉)], Yi Shi[2], Shuangliang Zhao[1,3], and Zengwei Lv[1,3]

[1] The School of Computer Science and Information Engineering, Hefei University of Technology, Hefei 230009, China

[2] Department of ECE, The Virginia Tech, Blacksburg, VA 24061, USA

[3] The Engineering Research Center of Safety Critical Industrial Measurement and Control Technology, Ministry of Education, Hefei 230009, China

shilei@hfut.edu.cn

Abstract. With the rapid development of the Internet of Vehicles (IoV), various types of compute-intensive vehicle applications are emerging and present significant challenges to resource-constrained vehicles. Emerging vehicular edge computing (VEC) can alleviate this situation by offloading computational tasks from vehicles to base stations (BSs) with edge servers at the roadside. And the excellent transmission performance of 5G provides more reliable support for VEC. However, due to the drawbacks of small coverage area and high energy cost of 5G BSs, long-term usage will result in huge costly resource investment. In this paper, we design a new 4G–5G hybrid task offloading framework for the VEC scenario. We consider switching some of the 5G BSs to sleep state during low traffic and low data consumption conditions, while letting the 4G BS process the tasks generated in these areas. We first build the mathematical model and find that it cannot be solved directly. Then we design the algorithm for the offline case and the online case, respectively. Simulation results show that our scheme significantly reduces the energy cost while ensuring high task success rate.

Keywords: 5G · Energy saving · Vehicular edge computing · Task offloading · Internet of vehicles

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, real-time video analytics and on-board infotainment services [1,2]. These applications all require intensive real-time computation. However, due to limited computing resources,

The work is supported by the major science and technology projects in Anhui Province, Grant No. 202003a05020009 and innovation foundation of the city of Bengbu, Grant No. JZ2022YDZJ0019.

H. Gao et al. (Eds.): CollaborateCom 2022, LNICST 460, pp. 309–325, 2022.
https://doi.org/10.1007/978-3-031-24383-7_17

vehicle local computing units are often unable to satisfy the computing demands of such applications. To overcome the limitation, vehicular edge computing (VEC) is an emerging and promising paradigm that provides fast computing services for vehicle users [3–5]. Specifically, through vehicle-to-infrastructure (V2I) communications, resource-constrained vehicle users are allowed to offload their latency-sensitive, compute-intensive tasks to 5G BSs configured with edge servers for processing [6–8]. In addition, compared to the conventional cloud computing, VEC can provide lower communication latency due to the proximity of edge servers to vehicles [9,10]. Consequently, vehicle users can receive better quality of service (QoS) [11].

However, in order to support the high density of vehicle users in cities, transportation systems need dense deployment of 5G BSs at the roadside. For now, 5G BSs have the disadvantages of high energy consumption and small coverage area [12], long-term usage will result in huge costly resource investment. In addition, the large amount of energy consumption will also accelerate global warming and deteriorate the environment. Therefore, how to reduce the energy consumption in VEC deserves investigation.

Previous studies have made some contributions to reducing the energy cost of VEC [13,14]. In these works, they focus on the energy consumed during task offloading and computation. In fact, more energy is wasted during low-traffic periods [15]. It is not necessary to keep 5G BSs active all the time. We can save energy by switching 5G BSs to sleep status during low-traffic periods. However, switching 5G BSs to sleep status raises a new problem. Unlike 5G BSs in cells, 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. 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 minimizing the energy consumption of 5G BSs in VEC. The contributions of this paper are summarized as follows.

– We propose a new hybrid 4G-5G offloading framework for VEC scenarios, where 5G BSs can be switched to sleep status during low-traffic periods, while tasks are offloaded to 4G BS while satisfying latency constraints.
– We developed a mathematical model in terms of both energy consumption and latency, and 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 reduce the energy cost of 5G BSs while ensuring a high task success rate.

The rest of this paper is organized as follows. In Sect. 2, the related works are introduced. In Sect. 3, the system model is presented, including the 5G BS energy consumption model and the task delay model. In Sect. 4, our offline and online algorithms are described. Detailed simulation results and conclusions of the paper are given in Sects. 5 and 6, respectively.

2 Related Works

There has been a lot of studies on energy cost optimization for vehicle edge computing, mainly focused on two aspects, one is the optimization of energy consumption for vehicles, and the other is the optimization of energy consumption for infrastructure such as base stations.

First, we briefly introduce the research on vehicle energy consumption optimization. Authors in [16] propose a multi-device and multi-server task Joint Task Offloading Game (JTOG) algorithm in order to minimize the energy consumption for all vehicular terminal devices generating tasks. Authors in [17] jointly optimize the offloading proportion and uplink/computation/downlink bit allocation of multiple vehicles, for the purpose of minimizing the total energy consumption of the vehicles under the delay constraint. Authors in [13,14] jointly optimize the latency and cost by considering both offloading decisions, communication and computational resource allocation.

Next, we introduce the research on energy consumption optimization of 5G BSs. Authors in [18] 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. Authors in [19] propose a sleep model for base stations in cellular networks and investigates the benefits of turning off a portion of base stations during low traffic. In the article, the authors propose a simple analytical model that determines the optimal base station shutdown time based on daily traffic patterns. However, in that paper the authors consider only one switchover for the base station, and the effect of this switchover on reducing the energy consumption and operating costs of the base station is relatively small. Authors in [20] reduce the power consumption of BSs by having unloaded BSs alternate between on and off in a cyclical manner. Authors in [21] propose an efficient algorithm to minimize the energy consumption by jointing the cell association and on-off scheme. Authors in [22] optimize the task latency while allowing the candidate BSs to randomly switch states between sleep and work to save energy consumption. Authors in [23] minimize energy consumption 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. Authors in [24] consider the scenario where multiple mobile users share multiple heterogeneous edge servers and propose an approximation algorithm to minimize the energy consumption of the MEC system. Authors in [25] consider optimizing the quality of user experience under a long-term energy budget constraint.

However, the above studies mainly focus on the base station switching approach for cellular network environments, and they do not consider the switching costs incurred when the base station switches states. In light of the existing works, we propose a new hybrid 4G–5G offloading framework for VEC scenarios. In this offloading framework, the 5G BSs can dynamically adjust its state, and vehicles can dynamically select the offloading method according to the state of the 5G BS. Meanwhile, we design the heuristic algorithm for the offline case and the online case, respectively. Through experimental simulations, it is confirmed that our scheme significantly reduce the energy cost of 5G BSs while ensuring a high task success rate.

3 System Model

We first describe the system model (see Fig. 1). Suppose a straight road with a long distance is covered by one 4G BS and several 5G BSs. Suppose the 5G BSs hava a quicker data transmission speed, a bigger energy consuming and a smaller cover area comparing with the 4G BS. 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 base stations 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, which means these BSs can do communicating jobs and computing jobs simultaneously. Suppose during the whole scheduling time vehicles will pass through the road, and when they pass through it, they may have tasks needed to be transmitted for handling. However, since the 5G BS energy consuming is high, we want to design an algorithm for "turn on/off " these 5G BSs appropriately, so that we can save energy while not influence vehicles work.

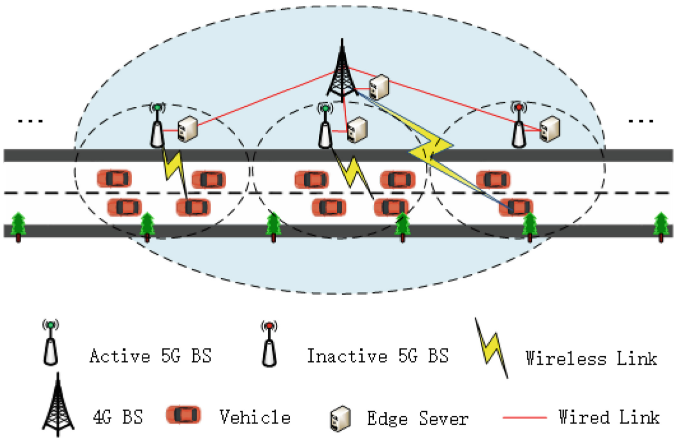

Fig. 1. 4G-5G hybrid task offloading framework in VEC.

3.1 Energy Model

We first discuss the energy model. Suppose the whole scheduling time T can be divided into h time slots τ equally and we normalize $\tau = 1$. Denote $t(t \in T, 1 \leq t \leq h)$ as a time slot. Denote $s_i(s_i \in N, 1 \leq i \leq n)$ as a 5G BS. Denote $l_j(l_j \in L, 1 \leq j \leq m)$ as a task. We divide the total energy consumption of 5G

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 [26]. Denote E^{total} as the total energy consumed by all 5G BSs in time period T. Therefore E^{total} can be expressed as

$$E^{total} = E^s + E^d + E^{switch}, \tag{1}$$

where E^s and E^d are the static energy and dynamic energy consumed by all 5G BSs in the time period T, respectively. E^{switch} is the state-switching energy consumed by all 5G 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 s_i at time slot t, then we have

$$\alpha_i(t) = \begin{cases} 1 : s_i \text{ is active at time slot } t; \\ 0 : \text{otherwise.} \end{cases} \tag{2}$$

Thus E^s can be expressed as

$$E^s = \sum_{t=1}^{h}\sum_{i=1}^{n}(\alpha_i(t) \cdot E^a + (1 - \alpha_i(t)) \cdot E^{ua}), \tag{3}$$

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.

For the second item E^d, suppose that there are m tasks needed to be offloaded and processed in time period T, and all tasks are equivalent. We describe these tasks with three attributes. One, the task data size, and we denote it as D. Two, the computing resource required for accomplishing the task which is quantified by the number of CPU cycles, and we denote it as W. Three, the maximum tolerable delay for tasks, and we denote it as T^{max}.

There are two possibilities for offloading tasks. First, the task is transmitted to the 4G BS. Second, the task is transmitted to the 5G BS. For each task, it can only be transmitted to one base station, and we use μ_j to indicate the offloading result for task l_j, then we have

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

Then E^d can be expressed as

$$E^d = \sum_{j=1}^{m}(1 - \mu_j) \cdot \zeta \cdot D, \tag{5}$$

where ζ is the dynamic energy consumed by a 5G BS to process a unit of task data.

For the third item E^{switch}, 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

$$x_i^{on} = \sum_{t=1}^{h} max\{(\alpha_i(t) - \alpha_i(t-1)), 0\}, \tag{6}$$

$$x_i^{off} = \sum_{t=1}^{h} max\{(\alpha_i(t-1) - \alpha_i(t)), 0\}. \tag{7}$$

Then E^{switch} can be expressed as

$$E^{switch} = \sum_{i=1}^{n} (x_i^{on} \cdot E^{on} + x_i^{off} \cdot E^{off}), \tag{8}$$

where E^{on} and E^{off} are the energy cost of turning on and off a 5G BS once, respectively.

3.2 Delay Model

In this subsection we continue to discuss the delay model. Suppose tasks generated at any time slot can be completed before the next time slot. The total task delay includes the transmission delay, the waiting delay and the calculation delay. Then for task l_j, we have

$$T_j^{total} = T_j^{trans} + T_j^{w} + T_j^{comp}, \tag{9}$$

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 first item T_j^{trans}, suppose tasks received by the same BS in the same time slot have the same transmission rate after the communication resources are allocated. Then when l_j is transmitted to the 4G BS, we can get the data transmission rate as

$$r_j^{4G} = \frac{R^{4G}}{k^{4G}(t_j)}, \tag{10}$$

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

$$r_{i,j}^{5G} = \frac{R^{5G}}{k_i^{5G}(t_j)}, \tag{11}$$

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

$$\beta_{i,j}(t) = \begin{cases} 1 : \text{the vehicle generating } l_j \text{ is within } s_i\text{'s coverage at time slot } t; \\ 0 : \text{otherwise.} \end{cases} \tag{12}$$

Therefore, the transmission rate of l_j can be expressed as

$$r_j = \begin{cases} r_j^{4G} : \text{if } \mu_j = 1; \\ r_{i,j}^{5G} : \text{if } \mu_j = 0,\ \beta_{i,j}(t_j) = 1,\ \alpha_i(t_j) = 1. \end{cases} \tag{13}$$

Then, we have:

$$T_j^{trans} = \frac{D}{r_j}. \tag{14}$$

For the second item T_j^w, 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. 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

$$\gamma_{j'}(l_j) = \begin{cases} 1 : l_{j'} \text{ is the previous task in the task queue of } l_j; \\ 0 : \text{otherwise.} \end{cases} \tag{15}$$

So T_j^w can be expressed as

$$T_j^w = \max\{\sum_{j'=1}^{m} \gamma_{j'}(l_j) \cdot ((t_{j'} + T_{j'}^{trans} + T_{j'}^w + \frac{W}{f}) - (t_j + T_j^{trans})), 0\}, \tag{16}$$

where f is the computing capability of an edge server.

For the third item T_j^{comp}, it can be expressed as

$$T_j^{comp} = \frac{W}{f}. \tag{17}$$

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

$$T_j^{total} \le T^{max}. \tag{18}$$

Then our problem can be formulated as

$$\begin{aligned} &\min \ \tfrac{E^{total}}{T} \\ &\text{s.t.} \ (1), (3), (5) - (11), (13), (14), (16) - (18) \\ &\qquad \alpha_i(t) \in \{0, 1\}, \forall i \in [1, n], \forall t \in [1, h] \\ &\qquad \mu_j \in \{0, 1\}, \forall j \in [1, m]. \end{aligned} \tag{19}$$

In (19), $\alpha_i(t)$ and μ_j are binary variables. T, h, m, and other symbols are all constants or determinable values. However, these binary variables almost appear in all items with different forms. In real scenarios, we may only know the tasks that have been generated or being generated. So the values of $\alpha_i(t)$ and μ_j are difficult to be solved directly. We will try to get an approximate optimal solution in the next section.

4 Algorithms

In Sect. 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 sub Sect. 4.1. Then we discuss the online algorithm in sub Sect. 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 summarized into four steps as the following.

Algorithm 1. Offline Algorithm

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: $\frac{E^{total}}{T}$

for 5G BS $s_i \in N$ **do**
  for Task $l_j \in L$ $(\beta_{i,j}(t_j) = 1)$**do**
    Calculate T_j^{total} according to formula (9) under the condition that l_j is transmitted to the 4G BS;
    if$(T_j^{total} \leq T^{max})$ **then**
      $\alpha_i(t_j) = 0$, $\mu_j = 1$;
    else
      $\alpha_i(t_j) = 1$, $\mu_j = 0$;
    end if
  end for
  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 $\mathbb{T}$;
  for $T_q^{ua} \in \mathbb{T}$ **do**
    if$(T_q^{ua} < T_{on} + T_{off} || (E^a - E^{ua}) \cdot T_q^{ua} \leq E^{on} + E^{off})$
      $\alpha_i(t) = 1 (t \in T_q^{ua})$;
      Remove T_q^{ua} from $\mathbb{T}$;
    end if
  end for
end for
Calculate the final optimization total energy efficiency $\frac{E^{total}}{T}$;

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, after obtaining the state of s_i at each time slot by step one, we can find time slots which are adjacent and have the same state. Denote $T_q^{ua}(T_q^{ua} \in \mathbb{T}, q = 1, ...)$ as a time period consisting of a series of adjacent time slots in which the state of s_i is inactive.

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 as shown in Algorithm 1.

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. 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 $\frac{k^{max} \cdot D}{R^{4G}} + \frac{k^{max} \cdot W}{f} \leq T^{max}$. When both sides of the formula are equal, we can get $k^{max} = \lfloor \frac{T^{max} \cdot R^{4G} \cdot f}{D \cdot f + R^{4G} \cdot W} \rfloor$. 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. Suppose that vehicles are evenly distributed on the road, i.e., tasks are generated with equal probability in the coverage area of each 5G BS. Based on this, 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 summarized into three steps as the following.

Step one, we first initialize 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 as shown in Algorithm 2.

Algorithm 2. Online Algorithm

```
Initialization;
Initialize all 5G BSs to sleep state;
End Initialization;
for Time slot t ∈ [1, h] do
    for s_i ∈ N do
        if(α_i(t) = 0) then
            μ_j = 1(β_{i,j}(t_j) = 1, t_j = t);
        else
            μ_j = 0(β_{i,j}(t_j) = 1, t_j = t);
        end if
    end for
    if(s(t) > 0 && k(t) > n/s(t) · k^{max})then
        Turn on a 5G BS;
    else if(a(t) > 0 && all values from k(t − (E^{on}+E^{off})/(E^{a}−E^{ua})) to k(t) are less than n/(n−a(t)+1) · k^{max}) then
        Turn off a 5G BS;
    else
        all 5G BSs remain in their current state;
    end if
end for
Calculate the final optimization total energy efficiency E^{total}/T;
```

5 Simulation

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.

5.1 Simulation Setup

In the simulation, we consider a one-way road with a length of 800 m. A 4G BS is deployed in the middle of the roadside, and its coverage radius is 400 m. Since the coverage of 5G BS is generally 200–300 m, 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 base stations is shown in Table 1.

Table 1. 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, 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)	$\{0.2 \sim 1\text{Mbit}\}$
Computation size of task (W)	10M CPU cycles
Maximum delay of task (T^{max})	1 s
Efficiency of dynamic energy consumption of 5G BS (ζ)	0.1 kJ/Mbit
Number of tasks(m)	$441000 \sim 882000$

5.2 Simulation Results

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

- **Always-Active:** where all 5G BSs are always in active state.
- **Always-Sleep:** where all 5G BSs are always in sleeping state.
- **Random-Switch:** where all 5G BSs are turned on and off randomly.
- **Greedy-Algorithm:** 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 online algorithm in terms of task success rate. In our experiments, we set T=86400 s , which means that the scheduling time in our simulation is a whole day consisting of 86400 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. Figure 2 shows the relationship between the task success rate and the number of tasks when we set $D = 0.8$M, $W = 10$M CPU cycles. It can be seen that task success rate of all schemes decrease with the increasing of the number of tasks except the Always-Active scheme and the offline algorithm. Because an increase in the total number of tasks equates to faster task generation. This will increase the burden on the base station, 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 Always-Active scheme and the offline algorithm, the effectiveness of our online algorithm is better than other algorithms.

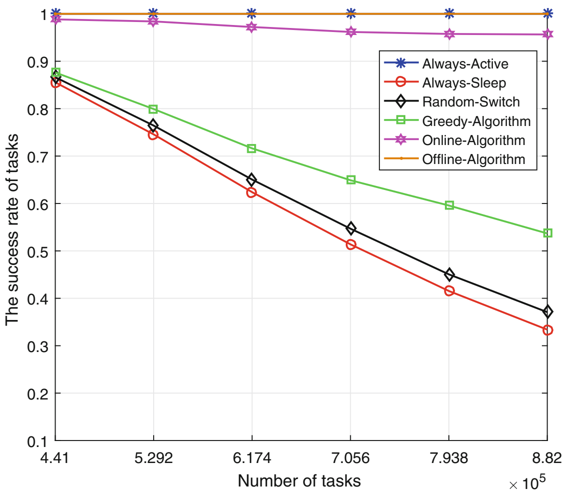

Fig. 2. The success rate of tasks under different number of tasks.

Then, we evaluate the performance of the proposed online algorithm in reducing energy cost. Figure 3 shows the relationship between the average energy cost of 5G BSs and the number of tasks when we set $D = 0.8$M, $W = 10$M CPU cycles. It can be seen that the average energy cost of 5G BSs of all schemes increase with the increasing of the number of tasks except the Random-Switch scheme and the Always-Sleep scheme. In the Random-Switch scheme, since all 5G BSs are switched randomly under any cases, the energy consumption does not change much. However, due to the frequent switching of states of 5G BSs under this scheme, a large amount of switching energy consumption will be generated, resulting in the highest energy cost compared with other schemes. In the Always-Sleep scheme, since all 5G BSs are in a sleep state under any cases and do not process any tasks, the energy consumption remains unchanged, and only the static energy in the inactive state is consumed. Although this scheme consumes the lowest energy cost, it can be seen from Fig. 2 that the success rate of the tasks under the Always-Sleep scheme is very low. Therefore, this scheme is not effective. In the Greedy-Algorithm, 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 the energy cost and also affects the offloading of tasks, resulting in a lower task success rate. In contrast, it can be seen that the energy saving effect of our online algorithm is very close to the offline algorithm. In the case of the same task success rate, the energy saving effect is slightly weaker than that of the offline algorithm, but significantly better than other schemes except the Always-Sleep scheme.

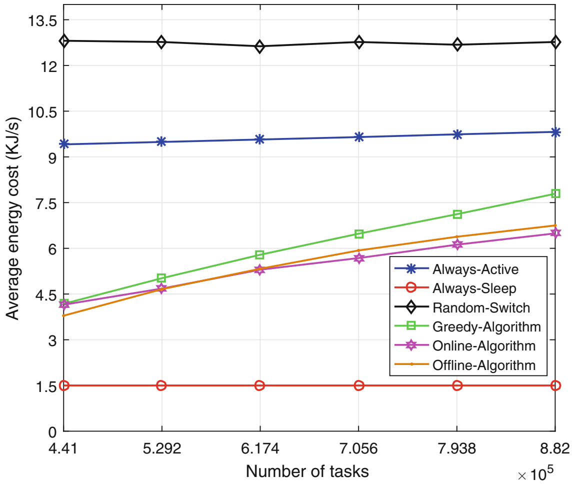

Fig. 3. The average energy cost of 5G BSs under different number of tasks.

To provide a more straightforward understanding, we present in Fig. 4 the proportion of tasks corresponding to the two offloading decisions of the online algorithm in Figs. 2 and 3. It can be seen that the proportion of tasks transmitted to the 5G BSs increases with the total number of tasks. Because this can ensure a higher task success rate.

Figure 5 shows the relationship between task success rate and task data size when we set $m = 705600$, $W = 10$M CPU cycles. It can be seen that task success rate of all schemes decrease with the increasing of the task data size except the Always-Active scheme and the offline algorithm. Because a larger data size requires more communication resources, this increases the probability of task failure. It can be seen that the advantages of our online algorithm are still obvious.

Figure 6 shows the relationship between the average energy cost of 5G BSs and task data size when we set $m = 705600$, $W = 10$M CPU cycles. It can

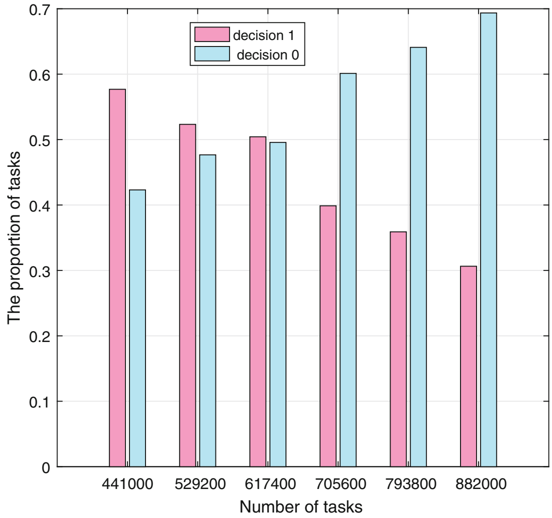

Fig. 4. The proportion of tasks corresponding to two offloading decisions under different total number of tasks.

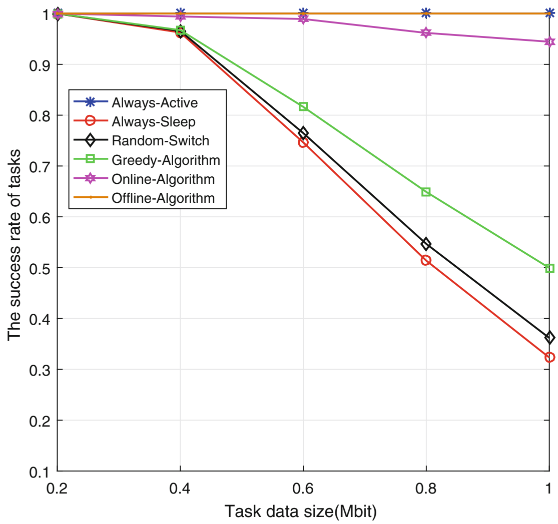

Fig. 5. The success rate of tasks under different task data size.

be seen that the average energy cost of 5G BSs of all schemes increase with the increasing of the task data size except the Always-Active scheme and the Random-Switch scheme. Although the energy-saving performance of our online algorithm is slightly inferior to that of the Greedy-Algorithm when the task data size is small, it is more stable and effective in guaranteeing a high task success rate.

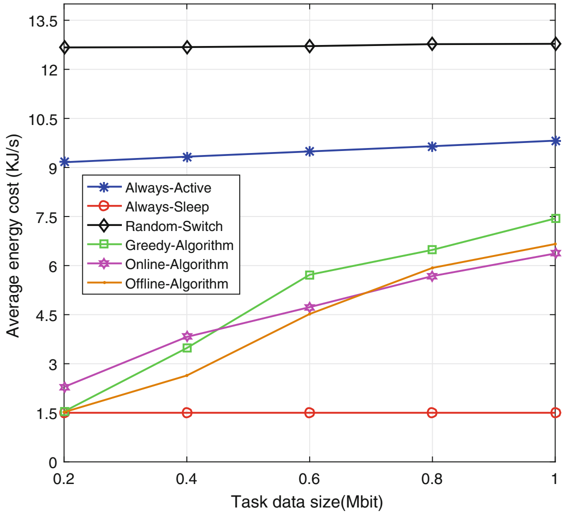

Fig. 6. The average energy cost of 5G BSs under different task data size.

Based on the above analysis, it can be concluded that our online algorithm can significantly reduce the energy cost while ensuring a high task success rate.

6 Conclusion

In this paper, we have investigated the problem of minimizing the energy cost of 5G BSs in VEC, and we propose a new hybrid 4G-5G task offloading framework which 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 experiments demonstrate that our scheme significantly reduces the energy cost while ensuring high task success rate.

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

References

1. Peng, H., Shen, X.: Deep reinforcement learning based resource management for multi-access edge computing in vehicular networks. IEEE Trans. Netw. Sci. Eng. **7**(4), 2416–2428 (2020)
2. Gao, J., Li, M., Zhao, L., Shen, X.: Contention intensity based distributed coordination for v2v safety message broadcast. IEEE Trans. Veh. Technol. **67**(12), 12288–12301 (2018)

3. Zhang, N., Zhang, S., Yang, P., Alhussein, O., Zhuang, W., Shen, X.S.: Software defined space-air-ground integrated vehicular networks: challenges and solutions. IEEE Commun. Mag. **55**(7), 101–109 (2017)
4. Liu, J., Wan, J., Zeng, B., Wang, Q., Song, H., Qiu, M.: A scalable and quick-response software defined vehicular network assisted by mobile edge computing. IEEE Commun. Mag. **55**(7), 94–100 (2017)
5. Shi, T., Li, Y., Cai, Z.: 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) (2022)
6. Cai, Z., Zheng, X., Yu, J.: A differential-private framework for urban traffic flows estimation via taxi companies. IEEE Trans. Industr. Inf. **15**(12), 6492–6499 (2019)
7. Ke, H., Wang, J., Deng, L., Ge, Y., Wang, H.: Deep reinforcement learning-based adaptive computation offloading for MEC in heterogeneous vehicular networks. IEEE Trans. Veh. Technol. **69**(7), 7916–7929 (2020)
8. Cai, Z., Shi, T.: Distributed query processing in the edge-assisted IoT data monitoring system. IEEE Internet Things J. **8**(16), 12679–12693 (2021)
9. Xu, X., Zhang, X., Liu, X., Jiang, J., Qi, L., Bhuiyan, M.Z.A.: Adaptive computation offloading with edge for 5g-envisioned internet of connected vehicles. IEEE Trans. Intell. Transp. Syst. **22**(8), 5213–5222 (2021)
10. Gu, X., Zhang, G., Cao, Y.: Cooperative mobile edge computing-cloud computing in internet of vehicle: architecture and energy-efficient workload allocation. Trans. Emerg. Telecommun. Technol. **32**(8), e4095 (2021)
11. Wheeb, A.H.: Performance analysis of VOIP in wireless networks. Int. J. Comput. Netw. Wirel. Commun. (IJCNWC) **7**(4), 1–5 (2017)
12. Cheng, X., Chen, C., Zhang, W., Yang, Y.: 5g-enabled cooperative intelligent vehicular (5genciv) framework: when Benz meets Marconi. IEEE Intell. Syst. **32**(3), 53–59 (2017)
13. Luo, Q., Li, C., Luan, T., Shi, W.: Minimizing the delay and cost of computation offloading for vehicular edge computing. IEEE Trans. Serv. Comput. **15**, 2897–2909 (2021)
14. Luo, Q., Li, C., Luan, T.H., Shi, W., Wu, W.: Self-learning based computation offloading for internet of vehicles: model and algorithm. IEEE Trans. Wirel.Commun. **20**(9), 5913–5925 (2021)
15. Auer, G.: How much energy is needed to run a wireless network? IEEE Wirel. Commun. **18**(5), 40–49 (2011)
16. Zhang, X., Debroy, S.: Energy efficient task offloading for compute-intensive mobile edge applications. In: ICC 2020–2020 IEEE International Conference on Communications (ICC), pp. 1–6 (2020)
17. Jang, Y., Na, J., Jeong, S., Kang, J.: Energy-efficient task offloading for vehicular edge computing: Joint optimization of offloading and bit allocation. In: 2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring), pp. 1–5 (2020)
18. Wu, L., Xu, J., Shi, L., Bi, X., Shi, Y.: Jointly optimizing throughput and cost of IoV based on coherent beamforming and successive interference cancellation technology. In: The 16th International Conference on Wireless Algorithms, Systems, and Applications(WASA), Nanjing, China, 25–27 June, pp. 235–243 (2021)
19. Ciullo, D., Marsan, M.A., Chiaraviglio, L., Meo, M.: Jointly optimizing throughput and cost of iov based on coherent beamforming and successive interference cancellation technology. In: 2012 Fourth International Conference on Communications and Electronics (ICCE), pp. 245–250 (2012)

20. Wen, C., Zheng, J.: An RSU on/off scheduling mechanism for energy efficiency in sparse vehicular networks. In: 2015 International Conference on Wireless Communications Signal Processing (WCSP), pp. 1–5 (2015)
21. Chavarria-Reyes, E., Akyildiz, I.F., Fadel, E.: Energy consumption analysis and minimization in multi-layer heterogeneous wireless systems. IEEE Trans. Mob. Comput. **14**(12), 2474–2487 (2015)
22. Sun, Y., Zhou, S., Xu, J.: EMM: energy-aware mobility management for mobile edge computing in ultra dense networks. IEEE J. Sel. Areas Commun. **35**(11), 2637–2646 (2017)
23. Zhao, Q., Gerla, M.: Energy efficiency enhancement in 5g mobile wireless networks. In: 2019 IEEE 20th International Symposium on "A World of Wireless, Mobile and Multimedia Networks" (WoWMoM), pp. 1–3 (2019)
24. Zhu, T., Shi, T., Li, J., Cai, Z., Zhou, X.: Task scheduling in deadline-aware mobile edge computing systems. IEEE Internet Things J. **6**(3), 4854–4866 (2019)
25. Jiang, H., Dai, X., Xiao, Z., Iyengar, A.K.: Joint task offloading and resource allocation for energy-constrained mobile edge computing. IEEE Trans. Mob. Comput. 1 (2022)
26. Elsherif, F., Chong, E.K.P., Kim, J.H.: Energy-efficient base station control framework for 5g cellular networks based on Markov decision process. IEEE Trans. Veh. Technol. **68**(9), 9267–9279 (2019)