This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVT.2021.3133586, IEEE Transactions on Vehicular Technology

Efficient Multi-Vehicle Task Offloading for Mobile Edge Computing in 6G Networks

Ying Chen, *Member, IEEE,* Fengjun Zhao, Xin Chen, and Yuan Wu, *Senior Member, IEEE*

***Abstract*—With the development of 6G wireless communication technologies, various resource-intensive and delay-sensitive vehicle application tasks are generated. These application tasks can be offloaded to Mobile Edge Computing (MEC) which deploys computing resources at the edge of networks. Besides, the recent proposed Cybertwin, as the digital representation of the complicated physical end-systems, can help the terminals obtain the required services from networks. Vehicles enabled by Cybertwin can offload their tasks to MEC and achieve better performance. In this paper, we focus on the study of a hybrid energy-powered multi-server MEC system with Cybertwin. Vehicles enabled by Cybertwin and edge servers send the current network status and unprocessed vehicle application tasks to the macro base station (MBS) to achieve the better allocation of resources. Energy harvesting (EH) devices are deployed on edge servers to form a "green energy-grid" hybrid energy supply model. We formulate a stochastic offloading optimization problem, and the goal is to minimize the system cost. The stochastic optimization problem is decomposed into three sub-problems. Then, we design an efficient multi-vehicle task offloading (EMT) algorithm to achieve the trade-off between system cost and task queue length. Theoretical analysis shows that EMT algorithm can optimize the total cost of the MEC system and guarantee the system performance. According to experimental evaluation, we verify the performance of the EMT algorithm.**

***Index Terms*—Mobile edge computing, Cybertwin, Task offloading, Hybrid energy supply, Stochastic optimization**

I. Introduction

The next-generation networks (6G) based on the Cybertwin provide a variety of functions when vehicles acquire services, such as communication assistants, network behavior records, and digital assets [1], [2]. The 6G networks enhance edge-cloud computing capabilities by establishing vehicle digital twin services. As a digital service technology in the 6G networks, Cybertwin creates a digital world through the digital representation of real-world vehicles and edge servers, making the information interaction between vehicles and edge networks more direct and efficient [3]. In the Cybertwin-enabled vehicle communications, the vehicles acquire services directly from the network through the corresponding Cybertwin, which can better realize the efficient communication and satisfying service for the vehicle. In addition, the future networks based on Cybertwin help guarantee the security and Quality of Service (QoS) of vehicle application information [4].

With the development of technologies such as autonomous driving and vehicle communication, application services that require high computing ablities and communication resources are widely used in the Internet of Vehicles (IoV) [5]. Nevertheless, the limited computing ablities of the vehicles seriously hinder the realization of smart environments. Relying only on vehicles for processing tasks cannot achieve good application performance. Thus, vehicles need to offload the complicated computing tasks of applications to platforms with more computing resources [6]. Traditional mobile cloud computing (MCC) has more resources and larger storage space, and can migrate tasks to remote clouds with powerful computing capabilities. However, the distance between MCC and vehicles is large, and the transmission of massive data from the vehicles to the MCC will cause large energy cost and transmission delay. In order to meet the development needs of Internet of Vehicles (IoV), mobile edge computing (MEC) is proposed as an emerging technology [7]. MEC deploys the computing resources of MCC at the wireless access network near the users, greatly reducing the computing delay and improving the service quality of terminal applications. MEC effectively solves the shortcomings of the MCC. In addition, a heterogeneous MEC network architecture including multiple small base stations (SBS) and macro base stations (MBS) is proposed recently, which helps allocate the computing and communication resources in a better way [8].

In this paper, we consider providing services for vehicles

This work is supported by National Natural Science Foundation of China (No. 61902029, No. 61972414, 62072490, No. 61973161), the Excellent Talents Projects of Beijing (No. 9111923401), the Scientific Research Project of Beijing Municipal Education Commission (No. KM202011232015), Beijing Nova Program (No. Z201100006820082), Beijing Natural Science Foundation (No. 4202066), the Fundamental Research Funds for Central Universities (No. 2462018YJRC040),the FDCT-MOST Joint Fund Project under Grant 0066/2019/AMJ, the Macao Science and Technology Development Fund under Grants 0060/2019/A1 and 0162/2019/A3, and Research Grant of University of Macau under Grant MYRG2018-00237-FST. *(Corresponding author: Ying Chen, E-mail: chenying@bistu.edu.cn)*

Y. Chen, F. Zhao and X. Chen are with School of Computer Science, Beijing Information Science and Technology University, Beijing 100101, China(E-mail: chenying@bistu.edu.cn; zhaofengjun@bistu.edu.cn; chenxin@bistu.edu.cn).

Y. Wu is with the State Key Laboratory of Internet of Things for Smart City, University of Macau, Macau SAR and the Department of Computer and Information Science, University of Macau. He is also with Zhuhai-UM Science and Technology Research Institute, Zhuhai, China (E-mail: yuanwu@um.edu.mo).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVT.2021.3133586, IEEE Transactions on Vehicular Technology

through the collaboration of SBSs and MBS in the Cybertwin-enabled MEC architecture. Energy harvesting devices are also integrated for providing energy [9]. Through the combination of Cybertwin and MEC, the status of the 6G communication environment can be monitored in real time, and real-time updated data can be quickly provided for vehicle offloading decisions. Cybertwin corresponding to various things in MEC is the key technology of the 6G digital world. However, because the 6G environment is complex, dynamic and unpredictable, it has encountered great challenges in vehicle task offloading. First, for vehicle task offloading in 6G networks, it is difficult or even impossible to precisely predict or estimate the statistical information such as the vehicle task arrival/generation process. Second, the wireless channel is stochastic and affected by the external environment such as the weather, and the energy harvesting process is dynamic and uncontrollable. Third, there is a slight deviation between the actual value of the service required by the vehicle task offloading and its digital representation. Fourth, with the increase of number of vehicles and amount of tasks, the offloading solution space size grows exponentially, and centralized methods are cost-consuming and suffer from high complexity.

In this paper, we propose an architecture of Cybertwin-enabled vehicle offloading for MEC (CEVOM) in 6G, and we study efficient multi-vehicle task offloading for mobile edge computing with the Cybertwin network architecture. The main contributions are as follows:

- First, we consider a multi-vehicle MEC system with the Cybertwin network architecture, including multiple heterogeneous SBSs and the MBS, and the offloading goal is to minimize the system cost with the green energy collaboration. The energy harvesting (EH) devices are deployed on the SBSs side to reduce the energy cost. By integrating EH technology into MEC, a "green energy-grid" hybrid energy supply mode is formed.
  The vehicle task request under the Cybertwin architecture is represented in the digital world. Cybertwin captures the time-varying resource supply and demand, and achieves unified resource scheduling and allocation. Cybertwin enabled vehicles are first connected to their corresponding Cybertwin, and the services required by the vehicles are provided by the edge server through the distributed MEC operating system.
- Second, we provide three optimization models of local computing, edge offloading and MBS computing. On the premise of meeting the task queue constraints, vehicles select the optimal edge server to offload tasks across regions in real time.
  The vehicles and SBSs under the Cybertwin network architecture build digital models of tasks and current network status, and then send digital information to the MBS.
  Our goal is to provide an optimal offloading strategy through resource scheduling to minimize the system cost.
- Finally, we use stochastic optimization techniques to transform the stochastic optimization problem into three deterministic sub-problems. Then, we design an efficient multi-vehicle task offloading (EMT) algorithm, which can make efficient task scheduling decisions and minimize the cost of MEC without any statistical external environmental information. According to experiments, we verify the effectiveness of the EMT algorithm.

TABLE I: KEY NOTATIONS

Notation	Definition
$\mathcal{I}$	vehicles set
$\mathcal{J}$	edge servers set
τ	Time slot length
$a_i(t)$	Tasks processsed locally from the vehicle i
$b_{ij}(t)$	Tasks offloaded from the vehicle i to the SBS j
$c_{ij}(t)$	Tasks of the vehicle i processsed by the SBS j
$d_{ij}(t)$	Tasks offloaded from the SBS j to the MBS
$P_{ij}^{l}(t)$	Transmit power of the vehicle i to the SBS j
N_0	Channel noise power spectral density
$e_i(t)$	Energy consumption of vehicle i
$H_{ij}(t)$	Power consumption of SBS j
$Q_{ij}(t)$	Queue length of the i-th vehicle's tasks in the edge server j

The rest of this paper is as follows. Section II proposes the system model and formulates the optimization problem with the goal of minimizing system costs. Section III decomposes the optimization problem into three sub-problems and design the EMT algorithm. Section IV performs a mathematical analysis to demonstrate the performance of the EMT algorithm. In Section V, according to parameter analysis and comparative experiments, we verify the performance of the EMT algorithm. Section VI reviews related work and Section VII summarizes this paper and discusses the future directions.

II. System Model and Problem Formulation

A. Cybertwin-enabled Edge Computing Network Model in 6G

We consider a multi-vehicle CEVOM system with a MBS and multiple SBSs in the 6G networks. Each SBS has a server to provide services for n vehicles. The vehicles and SBSs under the Cybertwin network architecture build digital models of tasks and current network status, and then, send the digital information to the MBS. Define $\mathcal{I}$ as the set of the vehicles

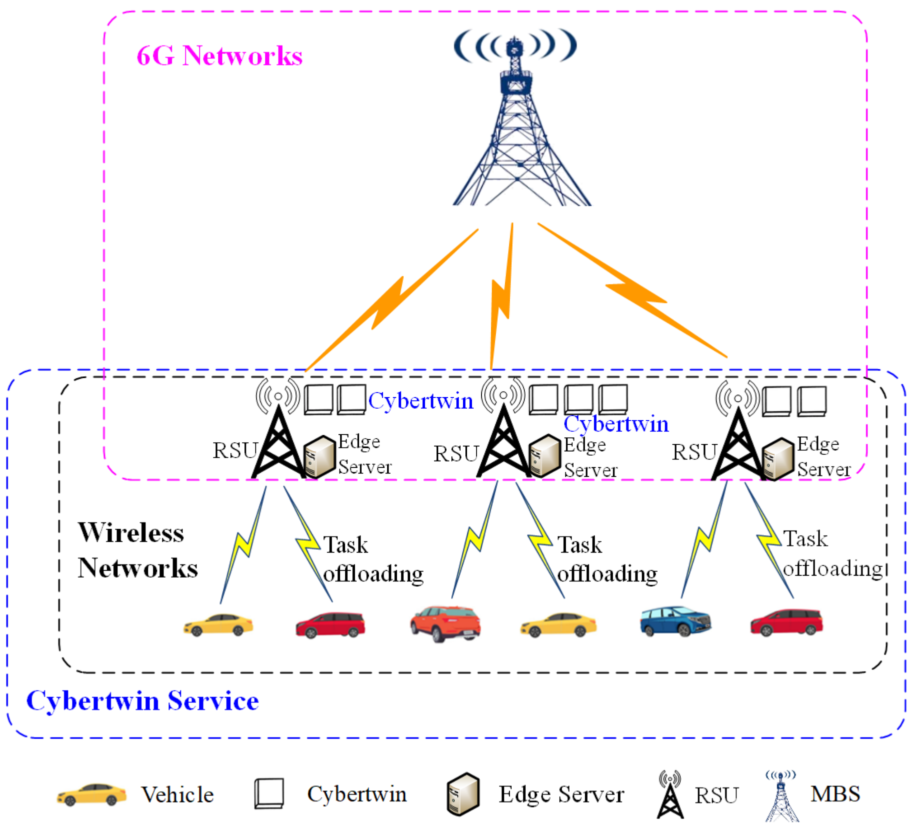

Fig. 1: Architecture of Cybertwin-enabled vehicle offloading for MEC in 6G

index i, where $\mathcal{I} = \{1, 2, ..., i, ..., n\}$. Define $\mathcal{J}$ as the set of the edge servers index j, where $\mathcal{J} = \{1, 2, ..., j, ..., m\}$. We assume that heterogeneous edge servers have different computing capabilities [10]. In this paper, task processing runs in units of time slot t, and $t \in \{0, 1, ..., T-1\}$. Fig. 1 shows the architecture of the Cybertwin-enabled vehicle offloading for MEC in 6G. In the 6G networks scenario based on the Cybertwin system architecture, Cybertwin abstracts the vehicle service request into digital information, and the vehicle is first connected to its corresponding Cybertwin. The 6G networks provide computing and communication resources for vehicles through a distributed network, and Cybertwin represents the services that vehicles obtain from the MEC. Vehicle communication in 6G networks is digitized by Cybertwin, which can update and record vehicle behavior data in real time.

In this paper, we consider a type of Cybertwin, that is, the Cybertwin of the request service between the vehicle and the edge server. The Cyberwin is a digital copy of the vehicle's offloading request, which continuously interacts with the vehicle and the edge server. In addition, the Cybertwin can be updated in real time according to the resource deployment of the edge server and the request of the vehicle. The Cybertwin can be stored on the edge server itself or on a resource-rich adjacent edge server. There is a slight deviation between the digital twin of the vehicle service request and the real value of the edge server. Thus, we use CPU frequency $\tilde{f}_i(t)$ to represent the deviation between the actual value of the edge server and its Cybertwin in the time slot t, where $\tilde{f}_i(t)$ can be positive or negative [11]. For the request service provided by the edge server j to the vehicle i, the Cybertwin in the time slot t can be expressed as

$$F_j(t) = \Omega(f_j(t), \tilde{f}_j(t)), \tag{1}$$

where $f_j(t)$ is the CPU cycle frequency estimated by the edge server deployed in SBS j.

B. Local Computing and offloading in CEVOM Model

$A_i(t)$ represents the tasks generated in time slot t. Different vehicles generate different $A_i(t)$. The tasks are divided into three parts of calculation, which are vehicles, edge server and MBS. Generally, $A_i(t)$ is difficult to obtain in reality, and we do not require any forecast information about $A_i(t)$. $S_{ij}(t)$ is the signal power and N_0 is the power density of the background noise. In addition, $S_{ij}(t)$ is related to transmission power $P_{ij}^l(t)$ and the channel power gain $h_{ij}^l(t)$. $h_{ij}^l(t)$ is negatively correlated with the distance of task offloading, and $h_{ij}^l(t) = \alpha_{ij}^r(t) g_{ij}^r(t)$. $a_{ij}^r(t)$ is the large-scale fading power

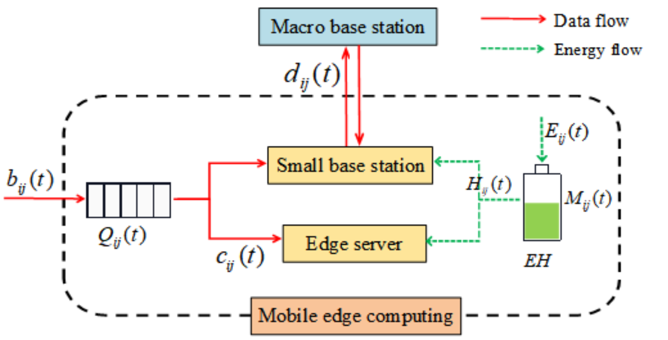

Fig. 2: Vehicle task offloading architecture of the MEC system in the 6G network

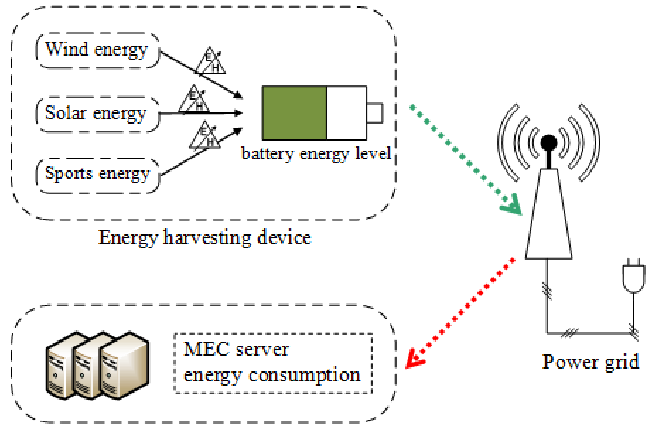

Fig. 3: Hybrid energy supply system composed of power grid and green energy

component of the vehicle communication channel, and $g_{ij}^r(t)$ is the small-scale channel fading component of the vehicle communication. According to $Shannon\ formula$, we obtain the rate $R_{ij}(t)$ of offloading tasks as follows:

$$R_{ij}(t) = B^l \log_2(1 + \frac{S_{ij}(t)}{B^l N_0}). \quad (2)$$

$a_i(t)$ is defined as the number of tasks calculated by the vehicle i. Computing 1 bit data requires δ_i CPU cycles, where $\delta_i > 0$. δ_i depends on different vehicle applications tasks. The CPU cycle frequency of the vehicle is denoted by $f_i^l(t)$, where $f_i^l(t) \leq f_i^{l,max}(t)$. Thus, it should satisfy

$$a_i(t) \leq \frac{f_i^l(t)\tau}{\delta_i}. \quad (3)$$

Define $b_{ij}(t)$ as the tasks offloaded from the vehicle i to the SBS j. The tasks $b_{ij}(t)$ of the vehicle i is limited, which is

$$b_{ij}(t) \leq R_{ij}(t)\tau. \quad (4)$$

C. Task Queuing Model and Energy Consumption Model

In time slot t, $Q_{ij}(t)$ represents the task queue length for offloading tasks from the vehicle i to the SBS j.

$c_{ij}(t)$ represents the number of tasks calculated by the edge server j, and $d_{ij}(t)$ represents the number of tasks transmitted by the SBS j to the MBS. Here we consider that MBS is supported by the back-end cloud and has huge computing resources. Thus, the edge server j queue backlog for the next time slot τ is:

$$Q_{ij}(t+1) = \max[Q_{ij}(t) - c_{ij}(t) - d_{ij}(t), 0] + b_{ij}(t). \quad (5)$$

All task queues need to satisfy (5) to guarantee the stability of MEC and reduce the queue delay of edge server.

$$\lim_{T\to\infty} \frac{E[Q_{ij}(t)]}{T} = 0. \quad (6)$$

We consider the energy cost includes the energy consumption of vehicles, SBSs and MBS. Vehicle terminal is divided into energy consumption of vehicle computing and offloading tasks, which is:

$$e_i(t) = \sigma f_i^{l2}(t)\delta_i a_i(t) + \sum_{j\in J} P_{ij}^l(t)\frac{b_{ij}(t)}{R_{ij}(t)}. \quad (7)$$

The first part on the right side of (6) is the energy consumption of vehicle i locally computing $a_i(t)$, and the latter part is the energy consumption of offloading $b_{ij}(t)$. σ represents the effective switched capacitor.

We obtain the total energy consumption of vehicle processing tasks, which is

$$e(t) = \sum_{i\in I}\{e_i(t)\}. \quad (8)$$

D. EH Model and Electricity cost Model

In this CEVOM system, the energy consumption of the SBSs are supported by two parts. One part comes from the grid, and the other part comes from the green energy captured by the energy harvesting (EH) device [12], [13]. Fig. 3 shows a hybrid energy supply system composed of green energy and the grid. When the task arrives, the edge servers will first use green energy to work. When the green energy is exhausted, the edge servers use the power grid.

$c_{ij}(t)$ is the number of tasks calculated by the edge server j, and f_j^{max} represents the maximum cycle frequency. For the task offloading request $c_{ij}(t)$ of the vehicle, there is a corresponding Cybertwin service digital system on the SBS j in the coverage area.

Thus, the computing capability of the edge server of the SBS j should satisfy

$$\sum_{i\in I} c_{ij}(t) \leq \frac{f_j^{max}\tau}{\delta_j}. \quad (9)$$

We consider that the power consumption of the SBS j is proportional to the number of computing tasks $c_{ij}(t)$. Define ξ as the power consumption for processing 1 bit data. Hence, the power consumption of SBS j can be defined as

$$H_{ij}(t) = \xi c_{ij}(t). \tag{10}$$

Each SBS has an EH device to provide energy for edge server. Green energy is stored in batteries, including solar and wind energy. $E_{ij}(t)$ represents the total amount of energy harvesting. Define $Y_{ij}(t)$ as the battery level, and $Y_{ij}(t)$ should satisfy

$$E_{ij}^{min} \leq Y_{ij}(t) \leq E_{ij}^{max}, \tag{11}$$

where E_{ij}^{min} and E_{ij}^{max} are the maximum and minimum amounts of energy stored by the battery.

We obtain the battery level as follows:

$$Y_{ij}(t+1) = \max[Y_{ij}(t) - H_{ij}(t), 0] + E_{ij}(t). \tag{12}$$

We consider the electricity cost is directly proportional to power consumption, where the unit price is defined as k [14]. Thus, the total electricity cost of SBSs is

$$s(t) = \sum_{i \in I}\sum_{j \in J} k \max[H_{ij}(t) - Y_{ij}(t), 0]. \tag{13}$$

E. MBS cost Model

In order to save energy consumption, SBS offloads some delay-insensitive tasks to MBS. The task offload rate of SBS j is defined as $R_{ij}^e(t)$ as follows:

$$R_{ij}^e(t) = B^e \log_2(1 + \frac{S_{ij}^e(t)}{B^e N_0}). \tag{14}$$

In time slot τ, since the wireless channel has a certain bandwidth, the task $d_{ij}(t)$ offload of SBS j has the following restrictions:

$$d_{ij}(t) \leq R_{ij}^e(t)\tau. \tag{15}$$

For task $d_{ij}(t)$ offloaded by SBS, MBS will calculate these tasks. $E(t)$ represents the energy consumption of MBS computing task $d_{ij}(t)$, which is

$$E(t) = l_1 \sum_{i \in I}\sum_{j \in J} d_{ij}(t), \tag{16}$$

where l_1 is the energy consumption coefficient. We assume the unit price of MBS energy consumption is $x(t)$. Thus, the grid energy required for MBS calculation task $d_{ij}(t)$ is

$$y(t) = x(t) l_1 \sum_{i \in I}\sum_{j \in J} d_{ij}(t). \tag{17}$$

F. Unified Optimization Problem

In this paper, the system cost mainly includes energy consumption and electricity charge. According to the tasks scheduling strategy, the tasks are divided into three parts to calculate: vehicle calculation, SBSs and MBS. System costs include the vehicles' energy consumption, edge server electricity costs, and MBS electricity costs [15]. Then, we weigh the minimum total cost in time slot t as follows

$$g(t) = \omega e(t) + \gamma s(t) + \eta y(t), \tag{18}$$

where $\omega \in [0,1]$, $\gamma \in [0,1]$ and $\eta \in [0,1]$ are the weight factors for calculating the cost of vehicles, SBSs and the MBS, respectively, $\omega + \gamma + \eta = 1$. ω, γ and η can be adjusted according to the priorities, preferences or specifications of the CEVOM system in 6G networks to realize the balance of cost among vehicles, edge servers in SBSs and the MBS.

In different time slots t, the states of wireless communication channel and the arrival of the task are dynamic and uncertain. Thus, we consider the MEC system's long-term ($T \to \infty$) average cost, which is:

$$\begin{aligned}
\min_{b(t),c(t),d(t)} g = \lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} E\{g(t)\} \\
s.t.\quad & C1: a_i(t) \leq \frac{f_i^l(t)\tau}{\delta_i}, \\
& C2: b_{ij}(t) \leq R_{ij}(t)\tau, \\
& C3: \lim_{T\to\infty} \frac{E[Q_{ij}(t)]}{T} = 0, \\
& C4: \sum_{i \in I} c_{ij}(t) \leq \frac{f_j^{max}\tau}{\delta_j}, \\
& C5: E_{ij}^{min} \leq Y_{ij}(t) \leq E_{ij}^{max}, \\
& C6: d_{ij}(t) \leq R_{ij}^e(t)\tau.
\end{aligned} \tag{19}$$

Solving this problem is of great challenges. First, for vehicle task offloading in 6G networks, it is difficult or even impossible to precisely predict or estimate the statistical information such as the vehicle task arrival/generation process. Second, the wireless channel is stochastic and affected by the external environment such as the weather, and the energy harvesting process is dynamic and uncontrollable. Third, with the increase of number of vehicles and amount of tasks, the offloading solution space size grows exponentially, and centralized methods are cost-consuming and suffer from high complexity.

III. ALGORITHM DESIGN

Next, we quote the *Lyapunov* optimization technique and design an EMT algorithm [16]. EMT algorithm can make a task offloading decision without some prior information, such as the arrival of the task, the state of the wireless channel and the amount of energy harvesting device collected.

A. Problem Transformation

$Lyapunov$ optimization technology can transform stochastic optimization problems (19) into deterministic problems. $\Gamma(t)$ represents the matrix of the server queue backlog, the $Lyapunov\ function$ is defined

$$\Upsilon(\Gamma(t)) = \frac{1}{2}\sum_{i\in I}[Q_{ij}^2(t)]. \tag{20}$$

$\Upsilon(\Gamma(t))$ is the status of the queue backlog. We can reduce the queue backlog by reducing $\Upsilon(\Gamma(t))$. Thus, we define $\Delta(\Gamma(t))$ as $Lyapunov\ drift$, which is

$$\Delta(\Gamma(t)) = E\{\Upsilon(\Gamma(t+1)) - \Upsilon(\Gamma(t))|\Gamma(t)\}. \tag{21}$$

We combine queue backlog and system cost to define $drift\ plus\ system\ cost$, which is:

$$\Delta(\Gamma(t)) + ME\{g(t)|\Gamma(t)\}, \tag{22}$$

where $M \geq 0$ is a trade-off parameter of queue backlog and system cost. Generally, when M is too large, it means that the system cost occupies a larger weight. Hence, the optimization problem can be expressed as

$$\begin{aligned} \min_{b(t),c(t),d(t)} \ & \Delta(\Gamma(t)) + ME\{g(t)|\Gamma(t)\} \\ s.t.\ \ & C1 : a_i(t) \leq \frac{f_i^l(t)\tau}{\delta_i}, \\ & C2 : b_{ij}(t) \leq R_{ij}(t)\tau, \\ & C3 : \lim_{T\to\infty} \frac{E[Q_{ij}(t)]}{T} = 0, \\ & C4 : \sum_{i\in I} c_{ij}(t) \leq \frac{f_j^{max}\tau}{\delta_j}, \\ & C5 : E_{ij}^{min} \leq Y_{ij}(t) \leq E_{ij}^{max}, \\ & C6 : d_{ij}(t) \leq R_{ij}^e(t)\tau. \end{aligned} \tag{23}$$

THEOREM 1: According to the scheduling strategy formulated by EMT algorithm, we calculate the upper bound of $drift\ plus\ cost$, which is:

$$\begin{aligned} \Delta(\Gamma(t)) &+ ME\{g(t)|\Gamma(t)\} \\ &\leq W + ME\{\omega e(t) + \gamma s(t) + \eta y(t)|\Gamma(t)\} \\ &+ \sum_{i\in I}\sum_{j\in J} Q_{ij}(t)E\{b_{ij}(t) - [c_{ij}(t) + d_{ij}(t)]|\Gamma\}, \end{aligned} \tag{24}$$

where $W = \frac{1}{2}\Sigma_{i\in I}\Sigma_{j\in j}\{[R_i(t)\tau]^2 + [\frac{f_e^{max}\tau}{\delta_i} + R_i(t)\tau]^2\}$ is a constant.

According to $max[Q_{ij}(t) - c_{ij}(t) - d_{ij}(t)]^2 \leq Q_{ij}^2(t) + [c_{ij}(t) + d_{ij}(t)]^2 - 2Q_{ij}(t)[c_{ij}(t) + d_{ij}(t)]$, we can obtain

$$\begin{aligned} Q_{ij}^2(t+1) \leq\ & Q_{ij}^2(t) + b_{ij}^2(t) + [c_{ij}(t) + d_{ij}(t)]^2 \\ & - 2Q_{ij}(t)[c_{ij}(t) + d_{ij}(t)] \\ & + 2b_{ij}(t)\max[Q_{ij}(t) - c_{ij}(t) - d_{ij}(t), 0]. \end{aligned} \tag{25}$$

Assuming $D_{ij}^{cd}(t) = c_{ij}(t) + d_{ij}(t)$ is the tasks calculated and offloaded by the edge server j, we can obtain

$$D_{ij}^{cd}(t) = \begin{cases} c_{ij}(t) + d_{ij}(t), & c_{ij}(t) + d_{ij}(t) \leq Q_{ij}(t) \\ Q_{ij}(t), & otherwise, \end{cases}$$

Since $\max[Q_{ij}(t) - c_{ij}(t) - d_{ij}(t)] = Q_{ij}(t) - D_{ij}^{cd}(t)$ and $D_{ij}^{cd}(t)$ is non-negative, we can rewrite (26) as follows

$$\begin{aligned} \frac{1}{2}[Q_{ij}^2(t+1) - Q_{ij}^2(t)] \leq\ & \frac{1}{2}\{b_{ij}^2(t) + [c_{ij}(t) + d_{ij}(t)]^2\} \\ & + Q_{ij}(t)\{b_{ij}(t) - [c_{ij}(t) + d_{ij}(t)]\}. \end{aligned} \tag{26}$$

Summing all the queues on both sides of (26), and $i \in I, j \in J$, we give the upper bound of $Lyapunov\ queue\ drift$, which is

$$\begin{aligned} \Delta(\Gamma(t)) \leq\ & \frac{1}{2}\sum_{i\in I}\sum_{j\in J}\{b_{ij}^2(t) + [c_{ij}(t) + d_{ij}(t)]^2\} \\ & + \sum_{i\in I}\sum_{j\in J} Q_{ij}(t)E\{b_{ij}(t) - [c_{ij}(t) + d_{ij}(t)]|\Gamma(t)\}. \end{aligned} \tag{27}$$

Adding $ME\{g(t)|\Theta(t)$ on both sides of (27), and assume $b_{ij}(t) \leq R_{ij}(t)\tau$ $c_{ij}(t) \leq \frac{f_e^{max}\tau}{\delta_{ij}}$ $d_{ij}(t) \leq R_{ij}^e(t)\tau$, we can have

$$\begin{aligned} \Delta(\Theta(t)) &+ ME\{g(t)|\Gamma(t)\} \\ &\leq W + ME\{g(t)|\Gamma(t)\} \\ &+ \sum_{i\in I}\sum_{j\in J} Q_{ij}(t)E\{b_{ij}(t) - [c_{ij}(t) + d_{ij}(t)]|\Gamma(t)\}. \end{aligned} \tag{28}$$

Substituting (18) into the right section of (28), we can obtain the formula (24). Since W is a constant, we do not consider W in the following part. Our goal problem is equivalent to minimize the upper bound, which is

$$\begin{aligned} \min_{b(t),c(t),d(t)} \ & ME\{\omega e(t) + \gamma s(t) + \eta y(t)|\Gamma(t)\} \\ & + \sum_{i\in I}\sum_{j\in J} Q_{ij}(t)E\{b_{ij}(t) - [c_{ij}(t) + d_{ij}(t)]|\Gamma(t)\} \\ s.t.\ \ & C1 : a_i(t) \leq \frac{f_i^l(t)\tau}{\delta_i}, \\ & C2 : b_{ij}(t) \leq R_{ij}(t)\tau, \\ & C3 : \lim_{T\to\infty} \frac{E[Q_{ij}(t)]}{T} = 0, \\ & C4 : \sum_{i\in I} c_{ij}(t) \leq \frac{f_j^{max}\tau}{\delta_j}, \\ & C5 : E_{ij}^{min} \leq Y_{ij}(t) \leq E_{ij}^{max}, \\ & C6 : d_{ij}(t) \leq R_{ij}^e(t)\tau. \end{aligned} \tag{29}$$

B. Efficient Tasks Offloading and Resource Management Algorithm

In this paper, we specifically analyze the characteristics of optimization problem and propose a dynamic EMT algorithm based on the Cybertwin architecture of the 6G networks. We study the optimization problems including a series of offloading decisions in the process of vehicle task offloading, in order to minimize the long-term task offloading cost under the constraint of average offloading delay. Problem (29) is the core problem, where the decision variables of the optimization problem include $b(t)$, $c(t)$, and $d(t)$. According to the stochastic optimization scheme, we transform the stochastic optimization problem into a deterministic problem, and transform the long-term offloading cost constraint into a multi-objective dynamic optimization problem. Because the decision variables $b(t)$, $c(t)$, and $d(t)$ are not coupled, we divide the optimization problem into three sub-problems to obtain the optimal solution.

We decompose problem (29) into three subproblems, namely (30), (31) and (34). In order to solve the problem (30), we propose a solution method based on the variable relaxation theory. We can obtain the optimal solution $b(t)$ for vehicle offloading by solving problem (30). Because problem (31) is a group discussion problem, we decompose problem (31) into (32) and (33) for solving, and the optimization of $c(t)$ can be obtained by solving (31). Problem (34) is a linear programming (LP) problem. Similar to problem (30), solving (34) can obtain the optimal solution $d(t)$ for MBS computing task.

1) Task Offloading Allocation: By considering (29) two parts related to $b_{ij}(t)$, and constraint (4), we can obtain the optimal solution for task offloding allocation $b_{ij}(t)$.

$$\begin{aligned} \min_{b(t)} \sum_{i\in I}\sum_{j\in J} & [\frac{M\omega P_{ij}^{l}(t)}{R_{ij}(t)} + Q_{ij}(t)]b_{ij}(t) \\ s.t.\quad & C1: b_{ij}(t) \le R_{ij}(t)\tau, \\ & C2: \lim_{T\to\infty}\frac{E[Q_{ij}(t)]}{T} = 0. \end{aligned} \tag{30}$$

$MwP_{ij}^{l}(t)/R_{ij}(t) + Q_{ij}(t)$ is the weight of $b_{ij}(t)$, and the $b_{ij}(t)$ can be expressed as

$$b_{ij}(t) = \begin{cases} R_{ij}(t)\tau, & i = i^{'} \\ 0, & i \ne i^{'}. \end{cases}$$

When $i = i^{'}$, the value of $\{V\omega P_{ij}^{l}(t)/R_{ij}(t) + Q_{ij}(t)\}$ is the minimum.

2) Edge Computing Allocation: According to the queue backlog of the MEC system, EMT algorithm can make task offloading decisions. Assuming $b_{ij}(t)$ and $d_{ij}(t)$ are known, we can deduce the second sub-problem (31).

$$\begin{aligned} \min_{c(t)} \sum_{i\in I}\sum_{j\in J} & \{M\gamma k \max[H_{ij}(t) - Y_{ij}(t), 0] - Q_{ij}(t)c_{ij}(t)\} \\ s.t.\quad & C1: \lim_{T\to\infty}\frac{E[Q_{ij}(t)]}{T} = 0, \\ & C2: \sum_{i\in I} c_{ij}(t) \le \frac{f_j^{max}\tau}{\delta_j}, \\ & C3: E_{ij}^{min} \le Y_{ij}(t) \le E_{ij}^{max}. \end{aligned} \tag{31}$$

Since the value of $H_{ij}(t) = \sum_{i\in I}\xi c_{ij}(t)$ and $Y_{ij}(t)$ cannot be determined, we will discuss in two cases.

a. If $H_{ij}(t) \ge Y_{ij}(t)$, we can rewrite second sub-question (31) to obtain (32).

$$\begin{aligned} \min_{c(t)} \sum_{i\in I}\sum_{j\in J} & [M\gamma k\xi - Q_{ij}(t)]c_{ij}(t) \\ s.t.\quad & C1: \lim_{T\to\infty}\frac{E[Q_{ij}(t)]}{T} = 0, \\ & C2: \sum_{i\in I} c_{ij}(t) \le \frac{f_j^{max}\tau}{\delta_j}. \end{aligned} \tag{32}$$

Problem (32) is a minimum weight problem. The task scheduling of the SBS j is weighted by the value of $M\gamma k\xi - Q_{ij}(t)$. According to the constraint (9), we obtain the optimal solution for edge scheduling $c_{ij}(t)$ as follows:

$$c_{ij}(t) = \begin{cases} \frac{f_j^{max}\tau}{\delta_j}, & i = i^{'} \\ 0, & i \ne i^{'}. \end{cases}$$

When $i = i^{'}$, $\{M\gamma k\xi - Q_{ij}(t)\}$ is the minimum.

b.If $H_{ij}(t) < Y_{ij}(t)$, we can rewrite the second sub-problem (31) to obtain (33).

$$\begin{aligned} \min_{c(t)} - \sum_{i\in I}\sum_{j\in J} & Q_{ij}(t)c_{ij}(t) \\ s.t.\quad & C1: \lim_{T\to\infty}\frac{E[Q_{ij}(t)]}{T} = 0, \\ & C2: \sum_{i\in I} c_{ij}(t) \le \frac{f_j^{max}\tau}{\delta_j}. \end{aligned} \tag{33}$$

By solving the sub optimization problem, we obtain

$$c_{ij}(t) = \begin{cases} \frac{f_j^{max}\tau}{\delta_j}, & i = i^{'} \\ 0, & i \ne i^{'}, \end{cases}$$

When $i = i^{'}$, $\{-Q_{ij}(t)\}$ is the minimum.

3) MBS Computing Allocation: For the MBS task scheduling problem $d_{ij}(t)$, we assume $b_{ij}(t)$ and $c_{ij}(t)$ are known.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVT.2021.3133586, IEEE Transactions on Vehicular Technology

Then, the third sub-question is as follows:

$$\begin{aligned} \min_{d(t)} \sum_{i\in I}\sum_{j\in J} &\{M\eta x(t)l_1 - Q_{ij}(t)\} d_{ij}(t) \\ s.t.\ \ C1 &: \lim_{T\to\infty} \frac{E[Q_{ij}(t)]}{T} = 0, \\ C2 &: d_{ij}(t) \le R^e_{ij}(t)\tau. \end{aligned} \tag{34}$$

Combining constraint conditions (15), we obtain the optimal solution of $d_{ij}(t)$.

$$d_{ij}(t) = \begin{cases} R^e_{ij}(t)\tau, & i = i^{'} \\ 0, & i \ne i^{'}. \end{cases}$$

When $i = i^{'}$, $\{M\eta x(t)l_1 - Q_{ij}(t)\}$ is the minimum.

IV. Analysis of EMT Algorithm

According to mathematical analysis, we verify the performance of the EMT algorithm. EMT can optimize the system cost for a long time, while guaranteeing the stable state of the queue length. Define $\overline{Q}$ as the long-term average queue backlog, which is

$$\overline{Q} = \lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} \sum_{i\in I}\sum_{j\in J} E\{Q_{ij}(t)\}. \tag{35}$$

We have formulated an optimal scheduling decision independent of the queue backlog, which can obtain the minimum of g^*. The specific details are explained in *lemma 1*.

Lemma 1: In each time slot, the arriving task $A_i(t)$ is random. We assume the task scheduling rate $\beta \in \Gamma$, where Γ represents the capacity range. There is an optimal scheduling strategy ς^* that can achieve the following goals:

$$E\{g^{\varsigma^*}(t)\} = g^*(\beta); \tag{36}$$

$$E\{b_{ij}(t)\} \le E\{c^{\varsigma^*}_{ij}(t) + d^{\varsigma^*}_{ij}(t)\}, \tag{37}$$

where $g^*(\beta)$ is the optimal scheduling cost under ς^* strategy.

proof: Prove *lemma 1* through Caratheodorys theorem [16], we will not elaborate here.

THEOREM 2: Since the arrives task $0 \le A_{ij}(t) \le A^{max}$, the scheduling cost g has an upper bound $\hat{g}$ and a lower bound $\breve{g}$. Assume a positive number ϖ, where $\beta+\varpi \in \Gamma$. Regardless of the value of V, the average queue length for a long time is satisfied:

$$\overline{Q} \le \frac{W + M(\hat{g} - \breve{g})}{\varpi}. \tag{38}$$

The average system cost g for a long time is (39).

$$g^{EMT} \le g^* + \frac{W}{M}. \tag{39}$$

W is a constant, which has been explained in *theorem 1*.

Algorithm 1 Efficient Multiuser Task Offloading (EMT) Algorithm

---

Input: $A_i(t)$, $f_i(t)$, B^l, $P^l_{ij}(t)$, B^e, $P^e_{ij}(t)$
Output: $Q_{ij}(t)$, $R_{ij}(t)$, $R^e_{ij}(t)$

```
for all i ∈ I do
    Traverse the i_1 of the minimum {M w P^l_ij(t)/R_ij(t) + Q_ij(t)}.
end for
Obtain the b_ij(t) from (27).
for all i ∈ I do
    if H_ij(t) ≥ Y_ij(t) then
        Traverse the index i_2 of the minimum value of {M r k ξ − Q_ij(t)}.
    else
        Traverse the index i_2 of the minimum value of {−Q_ij(t)}.
    end if
end for
Obtain the c_ij(t) from (28).
for all i ∈ I do
    Traverse the i_3 of the minimum value of {M η x(t) l_1 − Q_ij(t)}.
end for
Obtain the d_ij(t) from (31).
Calculate the total cost g(t) according (19).
```

proof: According to *lemma 1*, we suppose there is a random scheduling strategy $\varsigma^{'}$ that satisfies the following:

$$E\{g^{\varsigma^{'}}(t)\} = g^*(\beta + \varpi), \tag{40}$$

$$E\{b^{\varsigma^{'}}_{ij}(t)\} + \varpi \le E\{c^{\varsigma^{'}}_{ij}(t) + d^{\varsigma^{'}}_{ij}(t)\}. \tag{41}$$

Since the EMT optimization goal is the R.H.S of (19), referring to scheduling strategy $\varsigma^{'}$, we obtain

$$\begin{aligned} &\Delta(\Gamma(t)) + ME\{g(t)|\Gamma(t)\} \\ &\qquad \le W + MG\{g^{\varsigma^{'}}(t)|\Gamma(t)\} \\ &\qquad + \sum_{i\in I}\sum_{j\in J} Q_{ij}(t) E\{b_{ij}(t) - [c^{\varsigma^{'}}_{ij}(t) + d^{\varsigma^{'}}_{ij}(t)]\}. \end{aligned} \tag{42}$$

Substituting (40) and (41) into R.H.S of (42), and using iteration expectations, we have

$$\begin{aligned} &E\{\Upsilon(\Gamma(t+1)) - \Upsilon(\Gamma(t))\} + ME\{g(t)\} \\ &\le W + Mg^*(\beta+\varpi) - \varpi \sum_{i\in I}\sum_{j\in J} E\{Q_{ij}(t)\}. \end{aligned} \tag{43}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVT.2021.3133586, IEEE Transactions on Vehicular Technology

Moving part $ME\{g(t)\}$ to R.H.S of (43), we can have (44).

$$\begin{aligned} &E\{\Upsilon(\Gamma(t+1)) - \Upsilon(\Gamma(t))\} \\ &\leq W + M(g^*(\beta+\varpi) - E\{g(t)\}) - \varpi\sum_{i\in I}\sum_{j\in J}E\{Q_{ij}(t)\} \\ &\leq W + M(\hat{g} - \check{g}) - \varpi\sum_{i\in I}\sum_{j\in J}E\{Q_{ij}(t)\}. \end{aligned} \tag{44}$$

Next, (44) uses the scaling method to obtain (45).

$$\varpi\sum_{i\in I}\sum_{j\in J}E\{Q_{ij}(t)\} \leq W + M(\hat{g} - \check{g})T. \tag{45}$$

Adding (43) of all $t \in \{0, 1, ..., T-1\}$, and by the scaling method we obtain

$$M\sum_{t=0}^{T-1}E\{g(t)\} \leq (Mg^*(\beta+\varpi) + W)T. \tag{46}$$

Next, we divide both sides of (46) by T to get the following:

$$\frac{1}{T}\sum_{t=0}^{T-1}E\{g(t)\} \leq g^*(\beta+\varpi) + \frac{W}{M}. \tag{47}$$

Remark: Our EMT algorithm can achieve balance between queue length and system cost over long periods of time. By (39) we can conclude that if M is large, the scheduling cost g tends to be small. However, the queue length is limited by (38) upper bounds. When M is very large, the queue length will increase. Thus, EMT algorithm can achieve a trade-off between cost and queue length, and guarantee the stable state of the MEC system.

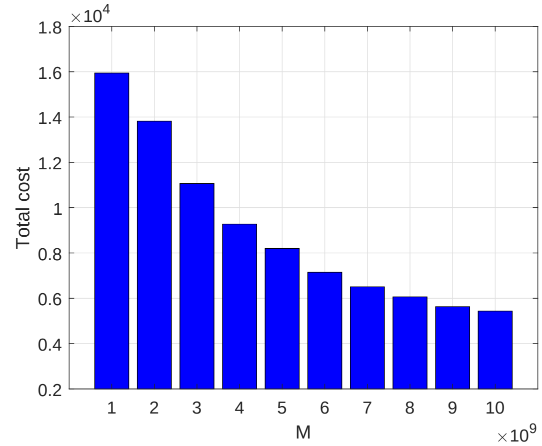

Fig. 4: Effect of different parameters M on total cost

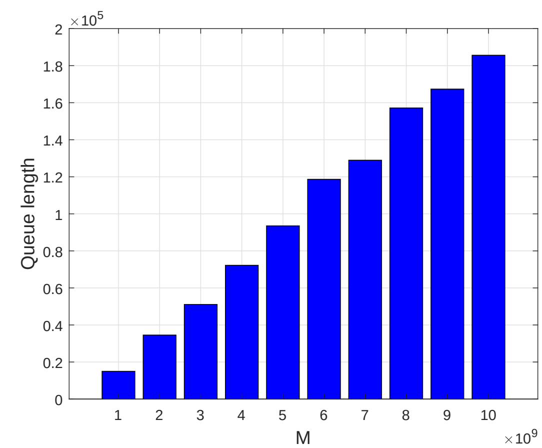

Fig. 5: Effect of different parameters M on queue length

V. Simulation Results

In this section, we apply matlab to run experiments. Then, we evaluate the performance of the EMT algorithm through parameter analysis and comparison experiments. SBSs deployed with edge servers are evenly distributed in an area of $3km \times 3km$ at a density of $1/km^2$, and each SBS provides wireless access to vehicles within a radius of 500 meters. The Cybertwin system can reflect the operating status of the 6G network in real time, and provide the edge server with training data for vehicle offloading requests to achieve efficient training. We list the parameters used in the evaluation. We assume there are 100 vehicles, and the distribution of task arrival process is $A_i(t) \sim U[0, 2000]$ bits in time slot t. We set the effective switching capacitance to 10^{-27} F, $N_0 = 10^{-9}$ W/Hz, and the number of CPU cycles frequency required to process 1 bit data is $[1000-2000]$ cycles/bit. The experiments are carried out for 300 slots.

In the time slot t, although the position of the vehicle changes at any time, the moving range of the vehicle in a short time period is relatively small, which has little affect on the task offloading. In different time slots, the set of connected SBSs for the vehicle is used as an input parameter for computing services based on the Cybertwin technology, not the decision variable. Similar models and methods have been widely adopted in related literature such as Ref. [11].

In the first section, we conduct multiple parameter analysis experiments. In the second section, we conduct a comparison experiment. By comparing random allocation, equal allocation and EMT algorithm, we validate the EMT algorithm's effectiveness.

A. Parameter Analysis

1) Effect of Parameter M: In this section, Fig. 4 and Fig. 5 analyze the effect of different M on MEC system costs and task queues. Fig. 4 shows that the system costs decreases as M increases, which is confirmed (39). This is because the increase in M represents an increase in the weight of the system cost. Thus, the EMT algorithm would schedule an optimization strategy to reduce system costs. In Fig. 5, we can see that the task queue increases with the increases of M, which is in line with (38). Regardless of the value of M,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVT.2021.3133586, IEEE Transactions on Vehicular Technology

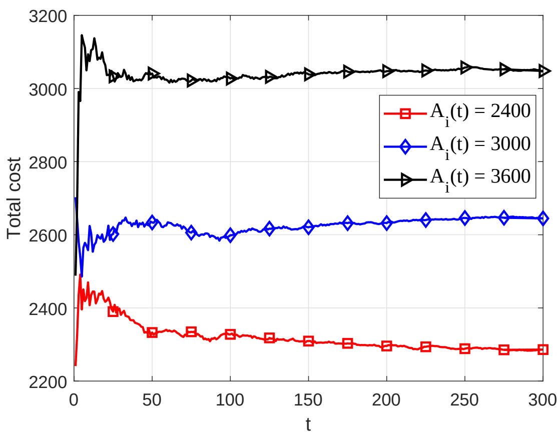

Fig. 6: Effect of different arrival tasks on total cost

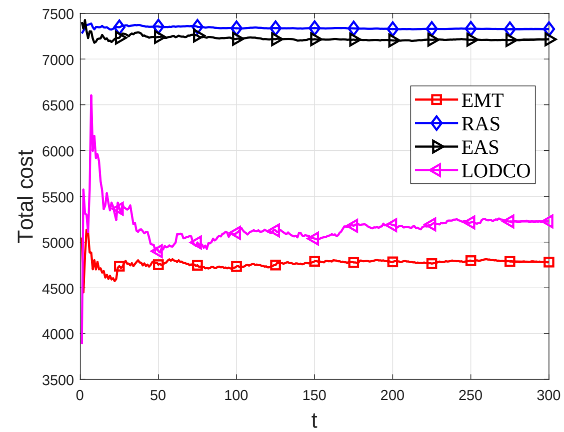

Fig. 8: Effect of different algorithms on total cost

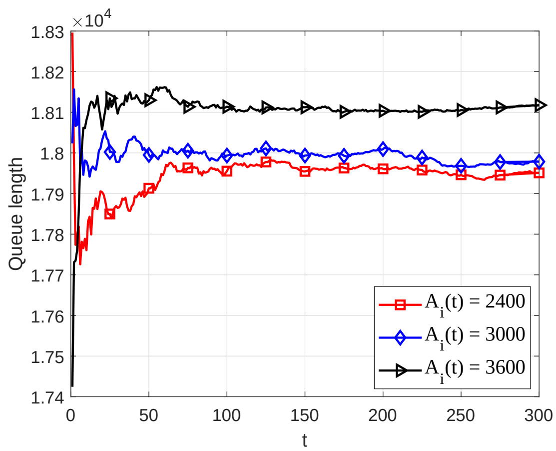

Fig. 7: Effect of different arrival tasks on queue length

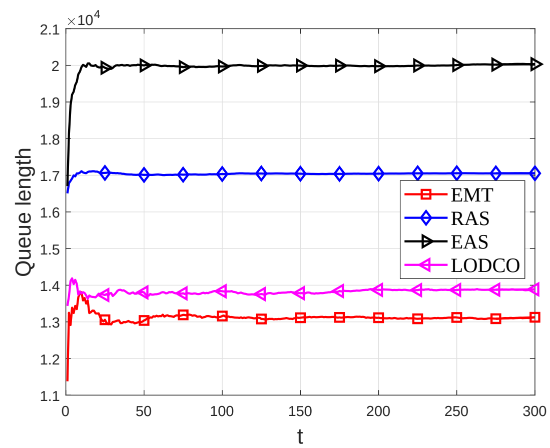

Fig. 9: Effect of different algorithms on queue length

EMT algorithm can make decisions to stabilize the task queue. From Fig. 4 and Fig. 5, we obtain that EMT algorithm can maintain the trade-off between system costs and tasks queue by adjusting M and gurantee the stability of the MEC system.

2) Effect of Task Arrived $A_i(t)$: We analyze the effects of task arrived $A_i(t)$ on MEC system cost and task queue. We set the task arrival rate $A_i(t)$ to be 2400 bits, 3000 bits, and 3600 bits respectively, and consider the different effects on the system cost and queue length. According to Fig. 8 and Fig. 9, the system cost and task queue increase with the number of tasks arrived. When the number of task arrived $A_i(t)$ increases, the calculation amount of the system increases. This causes increased local computing costs and offloading energy consumption. In addition, we can see that costs and queues will stabilize over a period of time. This shows that our EMT algorithm can adjust task scheduling strategies to adapt to different task amounts.

B. Comparison Experiment

Next, we compare the EMT algorithm with 3 other algorithms to verify the effective performance of the EMT algorithm. The 3 comparison algorithms are the Lyapunov optimization-based dynamic computation offloading (LODCO, tasks offloading with energy harvesting in the MEC system [17]), Random allocation strategy (RAS, tasks arrived are randomly distributed to vehicles, edge servers and MBS) and Equal allocation strategy (EAS, tasks arrived are equally distributed to vehicles, edge servers and MBS).

In Fig. 8, we verify the effect of different task allocation strategies on system cost through experiments. Fig. 8 shows that the total cost of the EMT algorithm is the lowest. Hence, our EMT algorithm can obtain effective scheduling strategies to reduce system costs. In Fig. 8, the costs of RAS and EAS are higher than the costs of 2 other algorithms. The cost of the LODCO algorithm is also higher than the cost of our EMT algorithm. This is because the EMT algorithm can propose an optimal task offloading strategy to reduce MEC system costs. However, RAS algorithm and EAS algorithm can not provide dynamic task allocation strategy according to the current system state to reduce the system cost. Thus, the EMT algorithm has good performance in optimizing the cost of MEC.

In Fig. 9, we verify the effect of different task allocation strategies on queue length through simulation experiments. The queue length of the EMT is lower than the other three algorithms, and the queue length of the LODCO algorithm is larger than our EMT algorithm but smaller than the other 2 algorithms. This is because both EMT and LODCO propose task offloading strategies based on queue optimization theory, and RAS and EAS algorithms do not consider the backlog of task queue. Combining Fig. 8 and Fig. 9, EMT algorithm can make dynamic task offloading decisions according to the current system state to effectively reduce system costs, and guarantee the stable state of the queue length. Thus, the comparison experiments show the advantages of EMT in reducing system cost while maintaining small queue length.

VI. Related Work

Most of the related studies assume that the resource allocation follows a certain pattern over a period of time, which ignore the huge burden of centralized resource allocation schemes on the Internet of Vehicles (IoV) system. In order to solve this problem, a dynamic Cybertwin system architecture can be introduced to represent the dynamic characteristics of resources. Cybertwin is the digital representation of the complex physical end-systems, and it helps the vehicles to obtain the required services from networks. [11] established a dynamic digital twin of the aerial auxiliary vehicle to capture the time-varying resource supply and demand, and optimized the resource allocation strategy of each vehicle. The authors designed a distributed incentive mechanism based on the Alternating Direction Multiplier Method (ADMM) to maximize vehicle satisfaction and overall energy efficiency. In order to solve the problem of edge server overload and reduce the Quality of Service (QoS), [18] designed a multi-vehicle user offloading system, which represented QoS through service response time.

As more and more computation-intensive application tasks are generated, these tasks can be offloaded to MEC system with Cybertwin for processing. By quoting Cybertwin communication technology, [1] proposed a new core cloud network architecture. The authors digitized people and things in Cyberspace to provide real-time distributed services to terminal users. [2] proposed a new network system based on the next-generation 6G communication technology of the Cybertwin. In addition, the authors proposed to center on the core cloud for scheduling and allocating communication resources. By integrating the digital twin in the MEC, [19] proposed a digital twin distributed network model to achieve low-latency communication. Vehicle applications with different resource requirements and unpredictable vehicle topologies posed a huge challenge to the realization of efficient edge computing services. In order to meet these challenges, [20] combined digital twin technology and artificial intelligence into the design of the MEC network. The authors proposed distributed vehicle task offloading and edge resource allocation based on a digital twin network to minimize offloading costs.

Efficient offloading strategy can improve the performance of MEC system, existing research work has achieved certain results. This section explains the related research on MEC resource allocation and hybrid energy supply. [21] proposed a delay-fuzzy incentive service-floating strategy to serve future mobile by managing a complex network, and the delay and total cost are regarded as the standard to measure the system performance. Based on the cellular network architecture, [22] aimed to maximize vehicle allocation resources, and studied the utilization of vehicles on the back-to-city link resources in MEC. [23] used aggregate games with load-based billing mechanism to arrange task calculation sequence to improve the server's task calculation speed. Based on the deep Q-learning theory, [24] studied the offloading modes of multiple types of vehicles and the different operating states of multiple edge servers. [25] considered the heterogeneity of edge and central cloud servers. In the hybrid offloading model including MCC and MEC, a deep learning-driven distributed task offloading algorithm is proposed to minimize the total system cost. In [26], the authors considered the assistant's CPU status information in the collaborative computing system, and the user can transmit data to the assistant to share computing tasks, so as to reduce the computing resources on the terminal device.

In [27], the authors used Lyapunov optimization technology to propose an efficient "edge-cloud" task offloading framework. The framework can make greedy decisions without any external information, and allocate MEC and cloud computing resources to IoT devices with the best strategy. In [28], the authors proposed a new cellular technology in MEC system, which combines computing offloading and resource management, with the optimization problem is to calculate the average weighted sum of transmission latency and offload energy consumption. [29] considered resource constraints and task response time constraints, with the goal of increasing the profitability of MEC and distributed cloud computing systems, and proposed a resource allocation method between cloud data centers (CDCs) and the edge. However, these works study single user or single server. In this article, we focus on the study of multi-user and multi-server MEC systems. We introduce the green energy harvesting technology into the MEC system, and each edge server is deployed with energy harvesting devices to minimize system costs. Vehicles and edge servers return the current network status and unprocessed tasks to MBS at each time to achieve the optimal computing performance of the MEC system.

VII. Conclusion

In this paper, we minimize the total cost of MEC system with hybrid energy supply, and schedule vehicle application tasks among vehicles, edge servers and the MBS computing. Based on the next generation network architecture with Cybertwin, we propose the EMT algorithm to balance the system cost and queue length, and guarantee the stability of the MEC system. The EMT algorithm does not require any prior statistical information such as the arrival of the task, the state of the wireless channel and EH collection process. Both parameter analysis and comparison experiments are carried out to verify the long-term effectiveness of the EMT algorithm. For our future work, we will consider the privacy and security issues of the vehicle users as all the privacy data of the users can be obtained by Cybertwin.

References

[1] Q. Yu, J. Ren, Y. Fu, Y. Li and W. Zhang, "Cybertwin: An Origin of Next Generation Network Architecture," *IEEE Wireless Communications*, vol. 26, no. 6, pp. 111-117, December 2019, doi: 10.1109/MWC.001.1900184.

[2] Q. Yu, J. Ren, H. Zhou and W. Zhang, "A Cybertwin based Network Architecture for 6G," *2020 2nd 6G Wireless Summit (6G SUMMIT)*, 2020, pp. 1-5, doi: 10.1109/6GSUMMIT49458.2020.9083808.

[3] Y. Wu, K. Zhang and Y. Zhang, "Digital Twin Networks: a Survey", *IEEE Internet of Things Journal*, DOI: 10.1109/JIOT.2021.3079510, 2021

[4] H. Zhou, W. Xu, J. Chen and W. Wang, "Evolutionary V2X Technologies Toward the Internet of Vehicles: Challenges and Opportunities," *Proceedings of the IEEE*, vol. 108, no. 2, pp. 308-323, Feb. 2020, doi: 10.1109/JPROC.2019.2961937.

[5] T. D. T. Nguyen, V. Nguyen, V. -N. Pham, L. N. T. Huynh, M. D. Hossain and E. -N. Huh, "Modeling Data Redundancy and Cost-Aware Task Allocation in MEC-Enabled Internet-of-Vehicles Applications," *IEEE Internet of Things Journal*, vol. 8, no. 3, pp. 1687-1701, 1 Feb.1, 2021, doi: 10.1109/JIOT.2020.3015534.

[6] M. R. Anwar, S. Wang, M. F. Akram, S. Raza and S. Mahmood, "5G Enabled MEC: A Distributed Traffic Steering for Seamless Service Migration of Internet of Vehicles," *IEEE Internet of Things Journal*, doi: 10.1109/JIOT.2021.3084912.

[7] T. D. T. Nguyen, V. Nguyen, V. -N. Pham, L. N. T. Huynh, M. D. Hossain and E. -N. Huh, "Modeling Data Redundancy and Cost-Aware Task Allocation in MEC-Enabled Internet-of-Vehicles Applications," *IEEE Internet of Things Journal*, vol. 8, no. 3, pp. 1687-1701, 1 Feb.1, 2021, doi: 10.1109/JIOT.2020.3015534.

[8] J. Huang, B. Lv, Y. Wu, Y. Chen and X. S. Shen. "Dynamic Admission Control and Resource Allocation for Mobile Edge Computing Enabled Small Cell Network," *IEEE Transactions on Vehicular Technology*. doi: VT-2021-02207.

[9] K. Suto, H. Nishiyama and N. Kato, "Postdisaster User Location Maneuvering Method for Improving the QoE Guaranteed Service Time in Energy Harvesting Small Cell Networks," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 10, pp. 9410-9420, Oct. 2017, doi: 10.1109/TVT.2017.2702750.

[10] Q. Zhang, L. Gui, F. Hou, J. Chen, S. Zhu and F. Tian, "Dynamic Task Offloading and Resource Allocation for Mobile-Edge Computing in Dense Cloud RAN," *IEEE Internet of Things Journal*, vol. 7, no. 4, pp. 3282-3299, April 2020, doi: 10.1109/JIOT.2020.2967502.

[11] W. Sun, P. Wang, N. Xu, G. Wang and Y. Zhang, "Dynamic Digital Twin and Distributed Incentives for Resource Allocation in Aerial-assisted Internet of Vehicles," *IEEE Internet of Things Journal*, doi: 10.1109/JIOT.2021.3058213.

[12] Y. Chen, N. Zhang, Y. Zhang, X. Chen, W. Wu and X. S. Shen, "TOFFEE: Task Offloading and Frequency Scaling for Energy Efficiency of Mobile Devices in Mobile Edge Computing," *IEEE Transactions on Cloud Computing*, doi: 10.1109/TCC.2019.2923692.

[13] K. Suto, H. Nishiyama, N. Kato and T. Kuri, "Model Predictive Joint Transmit Power Control for Improving System Availability in Energy-Harvesting Wireless Mesh Networks," *IEEE Communications Letters*, vol. 22, no. 10, pp. 2112-2115, Oct. 2018, doi: 10.1109/LCOMM.2018.2860005.

[14] X. Wang, K. Wang, S. Wu, S. Di, H. Jin, K. Yang and S. Ou, "Dynamic Resource Scheduling in Mobile Edge Cloud with Cloud Radio Access Network," *IEEE Transactions on Parallel and Distributed Systems*, vol. 29, no. 11, pp. 2429-2445, 1 Nov. 2018, doi: 10.1109/TPDS.2018.2832124.

[15] K. Suto, K. Miyanabe, H. Nishiyama, N. Kato, H. Ujikawa and K. Suzuki, "QoE-Guaranteed and Power-Efficient Network Operation for Cloud Radio Access Network With Power Over Fiber," *IEEE Transactions on Computational Social Systems*, vol. 2, no. 4, pp. 127-136, Dec. 2015, doi: 10.1109/TCSS.2016.2518208.

[16] Y. Chen, F. Zhao, Y. Lu and X. Chen, "Dynamic Task Offloading for Mobile Edge Computing with Hybrid Energy Supply," *Tsinghua Science and Technology*, doi: 10.26599/TST.2021.9010050.

[17] Y. Mao, J. Zhang and K. B. Letaief, "Dynamic Computation Offloading for Mobile-Edge Computing With Energy Harvesting Devices," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 12, pp. 3590-3605, Dec. 2016, doi: 10.1109/JSAC.2016.2611964.

[18] X. Xu, B. Shen, S. Ding and G. Srivastava,"Service Offloading with Deep Q-Network for Digital Twinning Empowered Internet of Vehicles in Edge Computing," *IEEE Transactions on Industrial Informatics*, doi: 10.1109/TII.2020.3040180.

[19] Y. Lu, S. Maharjan and Y. Zhang, "Adaptive Edge Association for Wireless Digital Twin Networks in 6G," *IEEE Internet of Things Journal*, doi: 10.1109/JIOT.2021.3098508.

[20] K. Zhang, J. Cao and Y. Zhang, "Adaptive Digital Twin and Multi-agent Deep Reinforcement Learning for Vehicular Edge Computing and Networks," *IEEE Transactions on Industrial Informatics*, doi: 10.1109/TII.2021.3088407.

[21] A. Samanta and Y. Li, "Latency-Oblivious Incentive Service Offloading in Mobile Edge Computing," *IEEE/ACM Symposium on Edge Computing (SEC), Seattle, WA*, 2018, pp. 351-353, doi: 10.1109/SEC.2018.00042.

[22] H. Zhou, N. Cheng, J. Wang, J. Chen, Q. Yu and X. Shen, "Toward Dynamic Link Utilization for Efficient Vehicular Edge Content Distribution," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 9, pp. 8301-8313, Sept. 2019, doi: 10.1109/TVT.2019.2921444.

[23] M. Gao, R. Shen, J. Li, S. Yan, Y. Li, J. Shi, Z. Han and L. Zhuo, "Computation Offloading with Instantaneous Load Billing for Mobile Edge Computing," *IEEE Transactions on Services Computing,* doi: 10.1109/TSC.2020.2996764.

[24] K. Zhang, Y. Zhu, S. Leng, Y. He, S. Maharjan and Y. Zhang, "Deep Learning Empowered Task Offloading for Mobile Edge Computing in Urban Informatics," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 7635-7647, Oct. 2019, doi: 10.1109/JIOT.2019.2903191.

[25] H. Wu, Z. Zhang, C. Guan, K. Wolter and M. Xu, "Collaborate Edge and Cloud Computing With Distributed Deep Learning for Smart City Internet of Things," *IEEE Internet of Things Journal*, vol. 7, no. 9, pp. 8099-8110, Sept. 2020, doi: 10.1109/JIOT.2020.2996784.

[26] C. You and K. Huang, "Exploiting Non-Causal CPU-State Information for Energy-Efficient Mobile Cooperative Computing," *IEEE Transac-*

tions on Wireless Communications, vol. 17, no. 6, pp. 4104-4117, June 2018, doi: 10.1109/TWC.2018.2820077.

[27] M. Du, Y. Wang, K. Ye and C. Xu, "Algorithmics of Cost-Driven Computation Offloading in the Edge-Cloud Environment," *IEEE Transactions on Computers*, vol. 69, no. 10, pp. 1519-1532, 1 Oct. 2020, doi: 10.1109/TC.2020.2976996.

[28] L. Lei, H. Xu, X. Xiong, K. Zheng and W. Xiang, "Joint Computation Offloading and Multiuser Scheduling Using Approximate Dynamic Programming in NB-IoT Edge Computing System,"*IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 5345-5362, June 2019, doi: 10.1109/JIOT.2019.2900550.

[29] H. Yuan and M. Zhou, "Profit-Maximized Collaborative Computation Offloading and Resource Allocation in Distributed Cloud and Edge Computing Systems," *IEEE Transactions on Automation Science and Engineering*, doi: 10.1109/TASE.2020.3000946.

Ying Chen (S'13-M'19) is an Associate Professor in Computer School, Beijing Information Science and Technology University, China. She received the Ph. D degree from Tsinghua University in 2017, and was a joint Ph. D student in University of Waterloo, Canada from 2016 to 2017. She is the recipient of the Best Paper Award at IEEE SmartIoT 2019, 2016 Google Ph. D Fellowship Award and 2014 Google Anita Borg Award, respectively. She serves/served the TPC member of IEEE HPCC, and PC member of IEEE Cloud, CollaborateCom, IEEE CPSCom, CSS, etc. She is also the reviewer of several journals such as IEEE Wireless Communications Magazine, IEEE Transactions on Dependable and Secure Computing, IEEE Internet of Things Journal, IEEE Transactions on Computers, IEEE Transactions on Cloud Computing, IEEE Transactions on Services Computing, etc. Her current research interests include Internet of Things, mobile edge computing, wireless networks and communications, machine learning, etc.

Fengjun Zhao M. S. candidate in the School of Computer Science from Beijing Information Science and Technology University, Beijing, China. Her research interests include Internet of Things, mobile edge computing and wireless networks.

Xin Chen received the Ph.D. degree in computer science from the Beijing Institute of Technology, Beijing, China. He is currently a Professor with the Computer School, Beijing Information Science and Technology University. His current research interests include performance evaluation of wireless networks. Dr. Chen received the Postdoctoral Fellowship in Computer Architecture from Tsinghua University in 2006. He is a Senior Member of the China Computer Federation (CCF), a member of the CCF Technical Committee of Theoretical Computer Science, and the CCF Technical Committee of Petri Nets.

Yuan Wu (Senior Member, IEEE) received the Ph.D. degree in electronic and computer engineering from The Hong Kong University of Science and Technology in 2010. From 2016 to 2017, he was a Visiting Scholar with the Department of Electrical and Computer Engineering, University of Waterloo, Canada. He is currently an Associate Professor with the State Key Laboratory of Internet of Things for Smart City, University of Macau, and also with the Department of Computer and Information Science, University of Macau. His research interests include resource management for wireless networks, green communications and computing, mobile edge computing, and smart grids. He was a recipient of the Best Paper Award from the IEEE International Conference on Communications in 2016, the Best Paper Award from IEEE Technical Committee on Green Communications and Computing in 2017, and the Best Paper Award from the International Wireless Communications and Mobile Computing Conference in 2021. He is currently on the editorial boards of IEEE Transactions on Vehicular Technology, IEEE Transactions on Network Science and Engineering, IEEE Internet of Things Journal, and IEEE Open Journal of the Communication Society. He served as the guest editor for IEEE Communications Magazine, IEEE Network, IEEE Wireless Communications Magazine, IEEE Transactions on Industrial Informatics, and IET Communications.