AoI-Aware Service Provisioning in Edge Computing for Digital Twin Network Slicing Requests

Jing Li, Song Guo, *Fellow, IEEE*, Weifa Liang, *Senior Member, IEEE*, Jianping Wang, *Fellow, IEEE*, Quan Chen, *Member, IEEE*, Zicong Hong, Zichuan Xu, *Member, IEEE*, Wenzheng Xu, *Member, IEEE*, and Bin Xiao, *Fellow, IEEE*

***Abstract*—Digital twins are poised to enter our lives with Industry 4.0. The Digital Twin Network (DTN) paradigm is projected to deliver upon the promise of efficient collaboration among digital twins to enable complicated and systematic services across many domains, through depicting an overall picture of a group of physical objects. To achieve timely data processing of digital twins, Mobile Edge Computing (MEC) shifts the computational power towards the network edge, and network slicing is well-suited to bundle heterogeneous physical resources to build logical networks based on edge servers for accommodating DTNs. In light of this, in this paper we investigate DTN slicing-enabled service provisioning in MEC, where each DTN slice consists of one master digital twin and a set of worker digital twins, and each worker digital twin is synchronized through collecting data from a respective object periodically. The master digital twin aggregates the processed data from worker digital twins to model the DTN continuously for user query services, whilst meeting delay requirements of users. We capture the utility gain of a DTN slicing request based on the DTN model quality at its master digital twin that is impacted by the Age of Information (AoI), and we focus on two novel optimization problems: the utility maximization problem for a single DTN slicing request, and the dynamic utility maximization problem for multiple DTN slicing requests. We propose an approximation algorithm for the former, and an online algorithm with a provable competitive ratio for the latter. We also evaluate the performance of the proposed algorithms through simulations. Experimental results demonstrate that the proposed algorithms are promising, outperforming their counterparts by at least 10.2%.**

***Index Terms*—Digital twin, mobile edge computing, network slicing request, age of information, approximation algorithm, online algorithm, and resource allocation.**

Received 5 December 2023; revised 3 July 2024; accepted 14 August 2024. Date of publication 26 August 2024; date of current version 5 November 2024. The work of Jing Li, Weifa Liang, Song Guo, and Jianping Wang were supported in part by Hong Kong Research Grants Council (RGC) under the Collaborative Research Fund (CRF) under Grant C1042-23GF. The work of Jing Li was supported by the Post-Doc Fellowship Award from the Research Grants Council of the Hong Kong Special Administrative Region, China under Grant CityU PDFS2425-1S02. The work of Song Guo was supported in part by funds from the Key-Area Research and Development Program of Guangdong Province under Grant 2021B0101400003, in part by Hong Kong RGC Research Impact Fund under Grant R5060-19 and Grant R5034-18, in part by Areas of Excellence Scheme under Grant AoE/E-601/22-R, and in part by General Research Fund under Grant 152203/20E, Grant 152244/21E, Grant 152169/22E, and Grant 152228/23E. The work of Weifa Liang was supported by Hong Kong Research Grants Council (HK RGC) under CityU HK under Grant 7005845, Grant 8730094, Grant 9043510, and Grant 9380137, respectively. The work of Quan Chen was supported in part by the National Natural Science Foundation of China (NSFC) under Grant 62372118, and in part by the Guangdong Basic and Applied Basic Research Foundation under Grant 2024A1515030136. The work of Zichuan Xu was supported in part by NSFC under Grant 62172068 and in part the Shandong Provincial Natural Science Foundation under Grant ZR2023LZH013, and the work of Wenzheng Xu was supported in part by NSFC under Grant 62272328, and in part by Sichuan Science and Technology Program under Grant 2024NSFJQ0026. Recommended for acceptance by J. Choi. *(Corresponding author: Wenzheng Xu.)*

Jing Li, Weifa Liang, and Jianping Wang are with the Department of Computer Science, City University of Hong Kong, Hong Kong (e-mail: jing.li@cityu.edu.hk; weifa.liang@cityu.edu.hk; jianwang@cityu.edu.hk).

Song Guo is with the Department of Computer Science and Engineering, The Hong Kong University of Science and Technology, Hong Kong (e-mail: songguo@cse.ust.hk).

Quan Chen is with the School of Computers, Guangdong University of Technology, Guangzhou 510006, China (e-mail: quan.c@gdut.edu.cn).

Zicong Hong and Bin Xiao are with the Department of Computing, The Hong Kong Polytechnic University, Hong Kong (e-mail: zicong.hong@connect.polyu.hk; csbxiao@comp.polyu.edu.hk).

Zichuan Xu is with the School of Software, Dalian University of Technology, Dalian 116024, China (e-mail: z.xu@dlut.edu.cn).

Wenzheng Xu is with the College of Computer Science, Sichuan University, Chengdu 610065, China (e-mail: wenzheng.xu@scu.edu.cn).

Digital Object Identifier 10.1109/TMC.2024.3449818

I. Introduction

The rapid advancement of digital twin technology sheds light on seamless cyber-physical integration with the Industry 4.0 initiative [31], by virtue of developing high-fidelity digital clones of physical objects of interest to evolve alongside the latter in the course of their life cycles [34]. Through continuous monitoring, digital twins can synchronize with the states of their physical objects, so as to analyze, simulate, and predict the behaviours of the physical objects for informed and wise decision-making, in view of dynamic system changes [9].

With the proliferation of digital twins, the Digital Twin Network (DTN) is anticipated as an emerging network paradigm that consists of a group of digital twins to form a virtual network, which can provide services to users based on global information of digital twins [37], [44]. Specifically, It is desired to implement a digital twin to reflect all features of its object as closely as needed, which however leads to significant resource demands for data processing and storage. Therefore, a digital twin usually represents only a subset of features of its object [35], and an object can have its digital twin playing different roles in different DTNs with the specific needed features. For example, the DTN for traffic monitoring focuses on the location information and driving behaviors of vehicles, while the DTN for vehicle maintenance pays more attention to the states of software and hardware of vehicles.

As a consensus, digital twins rely on the timely processing of real-time data from objects to maintain living digital models [11], [30], and the traditional cloud-based architecture with

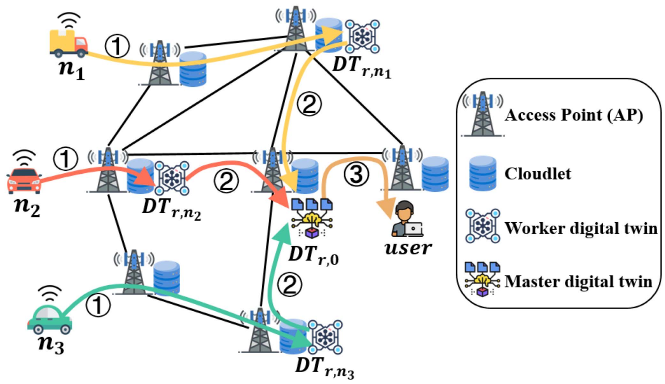


Fig. 1. An illustrative example of a DTN slice of a request r in an MEC network, where each Access Point (AP) has a co-located cloudlet. The DTN slice accommodates a DTN, which consists of one master digital twin $DT_{r,0}$ and three worker digital twins (DT_{r,n_1}, DT_{r,n_2} and DT_{r,n_3}), corresponding to three objects n_1, n_2 and n_3 , respectively. Each worker digital twin describes a subset of features of its object in the DTN, and the master digital twin models the entire DTN. The service provisioning via the DTN slicing is as follows. ① Each object periodically sends its raw data to its worker digital twin for processing and synchronization. ② The processed data from all its worker digital twins are sent to the master digital twin for further processing to model the DTN continuously. ③ The master digital twin provides query services to users.

long communication delay is insufficient [23]. In this context, Mobile Edge Computing (MEC) presents a new perspective for building digital twins close to objects at the network edge in a communication-efficient way [21], [28]. Network slicing is another crucial capacity towards future mobile networks, via establishing a series of logical virtual networks (network slices) to share the network infrastructure [45]. These network slices are tailored to fulfil diversified service requirements, serving as a basis for Slice-as-a-Service [25]. Assisted by network slicing in MEC, DTNs can be implemented in network slices based on edge servers (cloudlets) to support personalized services with fine-grained resource orchestration, low-delay customization, and performance isolation [10], [31].

In this paper, we deal with DTN slicing in an MEC network for admitting DTN slicing requests. As illustrated in Fig. 1, the *DTN slice* of each request consists of a set of *worker digital twins* and one *master digital twin*, where each worker digital twin describes a subset of features of an object in the DTN. As a centralized space, the master digital twin models the entire DTN, through aggregating the processed data from worker digital twins for further processing and providing query services based on the DTN model. For example, a master digital twin may adopt a Deep Neural Network (DNN) to model the DTN [32], [37], while the data collected from its worker digital twins will serve as the training data of the DNN. Because the volume of the processed data at a worker digital twin usually is far smaller than that of the raw data from its object, deploying worker digital twins for data prepossessing helps save communication resource for transmitting the data to the master digital twin, compared with sending the raw data from objects directly. Also, the prepossessing of the raw data at worker digital twins distributively alleviates the workload of the cloudlet hosting the master digital twin, especially when the DTN has a large number of digital twin components.

We consider the Quality of Service (QoS) for DTN slicing requests as follows. The quality of an established DTN model at the master digital twin is inevitably impacted by its data freshness [29], [34], which is usually quantified by the Age of Information (AoI), i.e., the duration of a piece of data since its generation [33]. Moreover, a user may issue multiple queries to the master digital twin during the DTN slicing, whilst imposing a service delay requirement [20], [27].

Service provisioning through DTN slicing in MEC networks poses several challenges. For example, how to measure the quality of the DTN model at the master digital twin to capture the utility gain of its request for DTN slicing? How to deploy the master and worker digital twins of a DTN slice to the MEC network to maximize the utility gain, while meeting its delay requirement? How to deal with dynamic admissions of DTN slicing requests without the future knowledge of request arrivals, with the aim to maximize the accumulative utility gain, considering limited resource capacities on cloudlets?

The novelty of this study lies in investigating AoI-aware service provisioning in an MEC for DTN slicing requests. A novel metric to capture the utility gain of a DTN slicing request is introduced to measure the quality of each DTN model by its AoI. Two optimization problems for admitting DTN slicing requests are formulated, and approximation and online algorithms for them are developed.

The main contributions of this paper are given as follows.

- We formulate two novel problems for DTN slicing in an MEC network: the utility maximization problem for a single DTN slicing request, and the dynamic utility maximization problem for multiple DTN slicing requests. We also show the NP-hardness of the two defined problems.
- We propose a performance-guaranteed approximation algorithm for the utility maximization problem.
- We propose the Integer Linear Program (ILP) for the offline version of the dynamic utility maximization problem, and develop an online algorithm with a provable competitive ratio for the problem based on the ILP, by adopting the primal-dual dynamic updating technique.
- We evaluate the performance of the proposed algorithms via simulations. The simulation results demonstrate the proposed algorithms are promising, outperforming their counterparts by at least 10.2%.

The rest of this article is arranged as follows. Section II surveys the related work on AoI-aware DTN slicing in MEC. Section III provides problem definitions and NP-hardness proofs of the defined problems. Section IV devises an approximation algorithm for the utility maximization problem, and Section V develops an online algorithm for the dynamic utility maximization problem. Section VI evaluates the algorithm performance, and Section VII concludes the article.

II. Related Work

Recently, there has been a significant body of studies of digital twins in MEC systems [9], [11], [12], [13], [14], [18], [19], [23], [30]. For instance, Jia et al. [9] proposed a concurrent synchronization and data re-sampling mechanism for constructing digital twins within an edge-centric platform. Li et al. [12], [13] studied the reliability-aware placement of service function chains assisted by digital twins in MEC, by proposing algorithms to minimize the request admission cost. Li et al. [18] investigated the delay-sensitive query service provisioning in DT-empowered serverless edge computing with user mobility. They deployed digital twin replicas in cloudlets to mitigate user query service delays while enhancing user service satisfactions by a utility function, for which they devised an approximation algorithm and an online algorithm for the problem under static and dynamic settings of user request arrivals. Li et al. [19] formulated an accumulative fidelity maximization problem that jointly considers the placement of DTs and service models, with the aim to refresh service models to satisfy their users, for which they devised efficient algorithms that strive to maximize the accumulative fidelity gain of service models while minimizing the cost of various resources consumed for digital twin synchronizations and continual training of service models. Lin et al. [23] designed a congestion control scheme to ensure the stability of long-term digital twin services with the aim of maximizing the profit, by adopting the Lyapunov optimization technique. However, the above studies did not consider optimizing the AoI metric in providing digital twin services.

There are also efforts devoted to providing AoI-aware digital twin services effectively [20], [29], [34]. Li et al. [15], [20] dynamically deployed digital twins in an MEC network to deal with the mobility of objects, whilst devising approximation and online algorithms to ensure query services for digital twin data with low AoI. Li et al. [16], [17] considered a joint optimization problem of the freshness of query results and query service delays in a DT-assisted MEC network, and devised an approximation algorithm for it. Liang et al. [22] developed algorithms to refresh the states of both DTs and service models, while the state freshness of DTs or service models is achieved through frequent synchronizations between DTs and their objects, using these updated digital twin data for their service model training. Vaezi et al. [34] considered the digital twin placement problem, and proposed algorithms to minimize the maximum response delay of requests while meeting AoI requirements of requests. Zhang et al. [40] considered multi-Federated Learning (FL) services by leveraging digital twin models to optimize mobile device scheduling and MEC resource allocation. They proposed efficient algorithms for the problems with the aim to maximize the accumulative utility of multiple FL services. Han [6] considered the system AoI to design an Internet of Things (IoT) system, where Unmanned Aerial Vehicles (UAVs) are adopted to collect data and serve as MEC servers. Zhang et al. [41], [42] considered mobility-aware inference service provisioning in a DT-assisted MEC network under the mobility assumption of both objects and users requesting for services, and developed efficient algorithms for the problem, by leveraging digital twin replica deployments [42] and migrations [41], respectively. Zhang et al. [43] studied the relationship between the cost of digital twin synchronization and placements and the cost of users for digital twin services through cost modeling and algorithm development.

The DTN paradigm is one main focus of the current research [3], [8], [26], [37], [44]. Hui et al. [8] created a digital twin for each vehicle to form a virtual network of an MEC network, and the information exchanges between digital twins are beneficial to collaborative driving with global information. Ren et al. [26] built a DTN consistent with a physical network to perform service configurations to validate potential solutions and designed a Deep Reinforcement Learning (DRL) method to improve service quality based on the DTN. Zhao et al. [44] presented a hierarchical routing strategy in a digital twin-based vehicle network, where DTNs are constructed to provide various functions. However, the above studies did not employ network slicing to facilitate the implementation of services enabled by DTNs.

Another investigation line incorporates network slicing to orchestrate the rising complexity of networks [7], [25], [27], [45]. Prasad et al. [25] proposed a network slicing mechanism for resource reservation, considering the urgency of slice requests. They designed offline and online algorithms to maximize the revenue of 5G networks. Zheng et al. [45] proposed a constrained game for network slicing and a flexible framework to optimize resource allocation. There are only a few studies exploring network slicing and DTN techniques to enhance network efficiency [21], [31], [36]. Li et al. [21] constructed a DTN to map an MEC network to realize a series of functions, such as prediction and decision-making. Assisted by the constructed DTN, they proposed a learning-based algorithm for dynamic cooperative network slicing, with the aim to optimize the resource allocation and the long-term utility of operators. Tang et al. [31] applied a DRL method to design a DTN-assisted network slicing framework, where a digital twin layer is set for simulation prediction and resource management. Wang et al. [36] adopted a graph neural network to build a digital twin of slicing-enabled networks in MEC to capture entangled interactions among slices and predict the end-to-end metrics of slices. However, the above studies did not consider dynamic admissions of multiple AoI-aware services enabled by DTN slicing.

Unlike the aforementioned studies, in this paper we investigate AoI-aware service provisioning in an MEC network for DTN slicing requests. We propose performance-guaranteed approximation and online algorithms for the problems of DTN slicing.

III. Preliminaries

In this section, we introduce the system model, notions, notations, and problem definitions.

A. System Model

Consider an MEC network $G = (V, E)$ with a set V of Access Points (APs) and a set E of links connecting APs. Each AP is co-located with a cloudlet, whilst connected via an optical fiber cable with negligible communication delay [24]. Notion $v \in V$ is abused to represent an AP or its co-located cloudlet. Let d_e be the communication delay for transmitting a unit of data along link $e \in E$ [39]. Denote by $\mathcal{N}$ the set of physical objects. The system model is illustrated in Fig. 1.

B. DTN Slicing Requests

There is a set R of DTN slicing requests issued by users. *The DTN slice* of request $r \in R$ consists of a set of *worker digital twins* and a *master digital twin*, where each worker digital twin describes a subset of features of one in a set N_r of objects with $N_r \subseteq \mathcal{N}$. Denote by $\mathbb{DT}_r = \{DT_{r,n_1}, \ldots, DT_{r,n_{|N_r|}}\}$ the worker digital twins of request r, and denote by $DT_{r,0}$ the master digital twin. Let $\mathbb{DT}_r^+ = \mathbb{DT}_r \cup \{DT_{r,0}\}$ be the set of digital twins for deployment for the admission of DTN slicing request r.

The service provisioning based on DTN slicing for a request is as follows. We first build a DTN slice, through deploying its master and worker digital twins in cloudlets of an MEC network. Within the built DTN slice, each object $n \in N_r$ periodically sends its raw data to the worker digital twin $DT_{r,n}$ for processing and synchronization, while the updated data from all worker digital twins are sent to the master digital twin for further processing, thereby modelling the DTN continuously. With such a DTN slice, a user can issue a series of queries to the master digital twin for services, and the master digital twin then responds to each query by sending the query result back to its user.

C. Expected AoI at the Master Digital Twin

The freshness of the collected data impacts the quality (accuracy) of the established DTN model at the master digital twin [29], [34], where the data freshness is commonly qualified by the AoI [33]. To achieve timely synchronization and provide fresh data, each object $n \in N$ transmits its updated data to its worker digital twin periodically, and let t_n be the synchronization interval of object n between its two consecutive synchronizations [15], [20], [34].

Objects usually are moving in the MEC network, and the mobility profile $L_{r,n} = \{p_{r,n,v} \mid v \in V\}$ of each object $n \in N_r$ can be obtained during the DTN slicing of request r, based on its historical movement traces [1], [24], where $p_{r,n,v}$ is the probability that object n moves to location v during the DTN slicing.

Because of the dynamic mobility of objects in an MEC network, it is difficult to calculate the exact AoI. In the following, we show *the expected AoI* of the master digital twin, through introducing the expected communication delay from an object to its worker digital twin, the processing delay at the worker digital twin, and the communication delay from a worker digital twin to its master digital twin.

The expected communication delay from an object to its worker digital twin: Denote by $s_{r,n}^{raw}$ the volume of the updated raw data of object n sent to its worker digital twin $DT_{r,n}$. Let $d_{l_n,v}$ be the communication delay for transmitting a unit of data from location l_n of object n (under the coverage of AP l_n) to cloudlet v hosting $DT_{r,n}$. The expected communication delay of transmitting the raw data of object n to its worker digital twin $DT_{r,n}$ in cloudlet $v \in V$ then is

$$t_{com}^{worker}(r,n,v) = \sum\nolimits_{l_n \in V} p_{r,n,l_n} \cdot s_{r,n}^{raw} \cdot d_{l_n,v} \tag{1}$$

The processing delay at the worker digital twin: Denote by $a_{r,n,v}$ the delay of a worker digital twin $DT_{r,n}$ in cloudlet v to process a unit of data. The processing delay of worker digital twin $DT_{r,n}$ in cloudlet $v \in V$ is

$$t_{proc}^{worker}(r,n,v) = s_{r,n}^{raw} \cdot a_{r,n,v}. \tag{2}$$

The communication delay from a worker digital twin to its master digital twin: Denote by $s_{r,n}^{worker}$ the volume of the processed data by the worker digital twin $DT_{r,n}$, which will be sent to the master digital twin $DT_{r,0}$. Given d_{v,v_0} the communication delay for transmitting a unit of data from cloudlet v to another cloudlet v_0, the communication delay of transmitting the processed data of worker digital twin $DT_{r,n}$ in cloudlet v to its master digital twin $DT_{r,0}$ in cloudlet v_0 is

$$t_{com}^{master}(r,n,v,v_0) = s_{r,n}^{worker} \cdot d_{v,v_0}. \tag{3}$$

The expected AoI of the data at the master digital twin from a worker digital twin: Given the update interval t_n of object n, we can see the expected AoI of the data at the worker digital twin $DT_{r,n}$ in cloudlet v is $t_n/2 + t_{com}^{worker}(r,n,v)$ by (1). Therefore, by (2) and (3), the expected AoI of the updated data at the master digital twin $DT_{r,0}$ in cloudlet v_0 from the worker digital twin $DT_{r,n}$ in cloudlet v is

$$t_{AoI}(r,n,v,v_0) = t_n/2 + t_{com}^{worker}(r,n,v) + t_{proc}^{worker}(r,n,v) + t_{com}^{master}(r,n,v,v_0). \tag{4}$$

D. Utility Gain

We now define the utility gain by admitting a DTN slicing request, based on the expected AoI at its master digital twin for the DTN modelling. In the following, we first capture the utility gain u_{r,n,v,v_0} for the master digital twin $DT_{r,0}$ in cloudlet v_0 to collect data from a worker digital twin $DT_{r,n}$ in cloudlet v, based on the expected AoI $t_{AoI}(r,n,v,v_0)$ of the collected data by (4). Intuitively, the utility gain u_{r,n,v,v_0} is inversely proportional to the expected AoI $t_{AoI}(r,n,v,v_0)$, i.e., a larger utility implies a smaller AoI. We then derive the utility gain $\mathcal{U}_r$ from each accepted request r.

Let $\xi_{r,n}$ be the AoI threshold [34] for the data from object n to model the DTN of request r. If the expected AoI is no greater than $\xi_{r,n}$, i.e., $t_{AoI}(r,n,v,v_0) \le \xi_{r,n}$, the utility gain $f_{r,n}(t_{AoI}(r,n,v,v_0))$ is obtained, where $f_{r,n}(\cdot)$ is a monotonically none-increasing function for the data from object n for request r, and $f_{r,n}(0)$ is a constant indicating the largest utility gain to be obtained when the expected AoI is 0, however, achieving AoI with 0 is very challenging. Otherwise ($t_{AoI}(r,n,v,v_0) > \xi_{r,n}$), a small utility gain $\lambda_{r,n}$ will be obtained, where $\lambda_{r,n}$ is a constant with $\lambda_{r,n} \le f_{r,n}(\xi_{r,n})$. Then, we have

$$u_{r,n,v,v_0} = \begin{cases} \lambda_{r,n}, & \text{if } t_{AoI}(r,n,v,v_0) > \xi_{r,n} \\ f_{r,n}(t_{AoI}(r,n,v,v_0)), & \text{otherwise.} \end{cases} \tag{5}$$

We further introduce a constant coefficient $w_{r,n}$ associated with each object $n \in N_r$ for request r, which indicates the importance of guaranteeing the data freshness from object n. Also, the value of $w_{r,n}$ is normalized for request r with $\sum_{n \in N_r} w_{r,n} = 1$. Then, the utility gain U_{r,n,v,v_0} of master digital twin $DT_{r,0}$ in cloudlet v_0 to collect data from worker digital twin $DT_{r,n}$ in cloudlet v

is

$$U_{r,n,v,v_0} = w_{r,n} \cdot u_{r,n,v,v_0}. \tag{6}$$

Given a request r, its master digital twin $DT_{r,0}$ in cloudlet v_0 aggregates all data from its worker digital twins, and establishes a service model based on the DTN. Suppose each worker digital twin $DT_{r,n} \in \mathbb{DT}_r$ is deployed in a cloudlet v_n, and the utility gain of admitting DTN slicing request r is

$$\mathcal{U}_r = \sum\nolimits_{DT_{r,n} \in \mathbb{DT}_r} U_{r,n,v_n,v_0}. \tag{7}$$

E. Delay Requirement

Given the DTN slice of request r, its user can issue a set Q_r of queries to the master digital twin $DT_{r,0}$ with a delay requirement [15], [20], [27]. The query service delay consists of (1) the query processing delay by $DT_{r,0}$; and (2) the communication delay of the query result from $DT_{r,0}$ to the user.

Denote by $t^{master}_{proc}(r, q, v_0)$ the processing delay of query $q \in Q_r$ by $DT_{r,0}$ in cloudlet v_0 and $s_{r,q}$ the volume of the query result. Supposing $DT_{r,0}$ is deployed in cloudlet v_0, the maximum query service delay $\mathcal{D}^{\max}_{r,v_0}$ for request r is $\mathcal{D}^{\max}_{r,v_0} = \max\{t^{master}_{proc}(r, q, v_0) + s_{r,q} \cdot d(v_0, loc_{r,q}) \mid q \in Q_r\}$, where $d(v_0, loc_{r,q})$ is the communication delay for transmitting a unit of data from cloud v_0 to the location $loc_{r,q}$ of the user for query q.

Users are usually mobile, and their mobility profiles can be obtained by analyzing historical movement traces [1], [24]. Therefore, $\mathcal{D}^{\max}_{r,v_0}$ for each incoming request r can be identified, and there is a set of cloudlets $\mathbb{V}_r \subseteq V$ for potential placements of $DT_{r,0}$ to meet the delay requirement D^{req}_r, with $\mathbb{V}_r = \{v_0 \mid v_0 \in \mathbb{V}_r \ \& \ \mathcal{D}^{\max}_{r,v_0} \le D^{req}_r\}$, i.e., the master digital twin $DT_{r,0}$ of a DTN request r can be placed in a cloudlet $v_0 \in \mathbb{V}_r$ only if the incurred maximum service delay $\mathcal{D}^{\max}_{r,v_0}$ is no greater than the delay requirement D^{req}_r.

F. Problem Definitions

Definition 1: Given an MEC network $G = (V, E)$, a set $\mathcal{N}$ of physical objects, and a request r for a DTN slice consisting of a set $\mathbb{DT}_r$ of worker digital twins and a master digital twin $DT_{r,0}$, *the utility maximization problem for a single DTN slicing request* is to maximize the utility gain of the request, through deploying its master and worker digital twins in G, subject to computing capacities on cloudlets in V.

Let $x_{r,n,v}$ be a binary variable, where $x_{r,n,v} = 1$ means the worker or master digital twin $DT_{r,n} \in \mathbb{DT}^+_r$ is deployed in cloudlet $v \in V$ for request r, and $x_{r,n,v} = 0$ otherwise. Denote by $c_{r,n}$ the amount of demanded computing resource of worker or master digital twin $DT_{r,n} \in \mathbb{DT}^+_r$. Denote by C_v the computing capacity of a cloudlet. The utility maximization problem for a single DTN slicing request is formulated as an Integer Nonlinear Program (INP) as follows.

$$\text{Maximize} \sum_{DT_{r,n} \in \mathbb{DT}_r} \sum_{v \in V} \sum_{v_0 \in \mathbb{V}_r} U_{r,n,v,v_0} \cdot x_{r,n,v} \cdot x_{r,0,v_0}, \tag{8}$$

subject to:

$$\sum\nolimits_{DT_{r,n} \in \mathbb{DT}^+_r} c_{r,n} \cdot x_{r,n,v} \le C_v, \quad \forall v \in V \tag{9}$$

$$\sum\nolimits_{v \in V} x_{r,n,v} = 1, \quad \forall DT_{r,n} \in \mathbb{DT}^+_r \tag{10}$$

$$x_{r,0,v} = 0, \quad \forall v \in V \backslash \mathbb{V}_r \tag{11}$$

$$x_{r,n,v} \in \{0, 1\}, \quad \forall DT_{r,n} \in \mathbb{DT}^+_r, \forall v \in V, \tag{12}$$

where Constraint (9) ensures the capacity constraints on cloudlets. Constraint (10) ensures that each worker or master digital twin is deployed in one cloudlet only. Constraint (11) guarantees that the master digital twin $DT_{r,0}$ will not be deployed in any cloudlet in $V \backslash \mathbb{V}_r$, otherwise, the delay requirement of the user is violated.

Definition 2: Given an MEC network $G = (V, E)$, a set $\mathcal{N}$ of objects, and a sequence R of DTN slicing requests arriving one by one, *the dynamic utility maximization problem for multiple DTN slicing requests* is to maximize the accumulative utility gain of admitted requests without any future knowledge, through dynamic admissions of incoming requests for their DTN slicing, subject to computing capacities on cloudlets in V.

Let y_r be a binary variable, where $y_r = 1$ means admitting request r, and $y_r = 0$ otherwise. The offline version of the dynamic utility maximization problem for multiple DTN slicing requests is formulated as an INP as follows.

$$\text{Maximize} \sum_{r \in R} \sum_{DT_{r,n} \in \mathbb{DT}_r} \sum_{v \in V} \sum_{v_0 \in \mathbb{V}_r} U_{r,n,v,v_0} \cdot x_{r,n,v} \cdot x_{r,0,v_0}, \tag{13}$$

subject to:

$$\sum\nolimits_{r \in R} \sum\nolimits_{DT_{r,n} \in \mathbb{DT}^+_r} c_{r,n} \cdot x_{r,n,v} \le C_v, \quad \forall v \in V \tag{14}$$

$$\sum\nolimits_{v \in V} x_{r,n,v} = y_r, \quad \forall r \in R, \ \forall DT_{r,n} \in \mathbb{DT}^+_r \tag{15}$$

$$x_{r,0,v} = 0, \quad \forall r \in R, \ \forall v \in V \backslash \mathbb{V}_r \tag{16}$$

$$x_{r,n,v} \in \{0, 1\}, \quad \forall r \in R, \quad \forall DT_{r,n} \in \mathbb{DT}^+_r, \ \forall v \in V \tag{17}$$

$$y_r \in \{0, 1\}, \quad \forall r \in R. \tag{18}$$

If request r is admitted ($y_r = 1$), Constraint (15) ensures that each of its worker or the master digital twin is deployed in one cloudlet only. Otherwise ($y_r = 0$), Constraint (15) means that request r is rejected with no digital twins deployed for it.

All symbols adopted are listed in Table I.

Theorem 1: The utility maximization problem for a single DTN slicing request in an MEC network $G = (V, E)$ is NP-hard.

Proof: The NP-hardness of the problem is shown through reducing the NP-hard Generalized Assignment Problem (GAP) [2] to it.

We consider a special case of the problem, where the master digital twin $DT_{r,0}$ has been deployed in cloudlet v_0. We perform the problem reduction as follows. With regard to each cloudlet

TABLE I
TABLE OF SYMBOLS

Notations	Descriptions
$G = (V, E)$	G is an MEC network, V is the set of APs (cloudlets) and E is the set of links connecting APs.
d_e	The communication delay for transmitting a unit of data along link $e \in E$
$\mathcal{N}$ and R	The set of physical objects, and the set of DTN slicing requests issued by users.
$DT_{r,0}$	The master digital twin of request r.
N_r	Each worker digital twin of request r describes a subset of features of one in a set N_r of objects with $N_r \subseteq \mathcal{N}$.
$\mathbb{DT}_r$	The worker digital twins of request r with $\mathbb{DT}_r = \{DT_{r,n_1}, \ldots, DT_{r,n_{\lvert N_r \rvert}}\}$.
$\mathbb{DT}_r^+$	The set of digital twins for deployment for the admission of DTN slicing request r with $\mathbb{DT}_r^+ = \mathbb{DT}_r \cup \{DT_{r,0}\}$.
$L_{r,n} = \{p_{r,n,v} \mid v \in V\}$	The mobility profile of object n during the DTN slicing of request r, and $p_{r,n,v}$ is the probability that object n moves to location v during the DTN slicing.
$s_{r,n}^{raw}$	The volume of the updated raw data of object n sent to its worker digital twin $DT_{r,n}$.
$d_{l_n,v}$	The communication delay for transmitting a unit of data from location l_n of object n to cloudlet v hosting $DT_{r,n}$
$t_{com}^{worker}(r,n,v)$	The expected communication delay of transmitting the raw data of object n to its worker digital twin $DT_{r,n}$ in cloudlet $v \in V$.
$a_{r,n,v}$	The delay of a worker digital twin $DT_{r,n}$ in cloudlet v to process a unit of data.
$t_{proc}^{worker}(r,n,v)$	The processing delay of worker digital twin $DT_{r,n}$ in cloudlet $v \in V$.
$s_{r,n}^{worker}$	The volume of the processed data by the worker digital twin $DT_{r,n}$.
d_{v,v_0}	The communication delay for transmitting a unit of data from cloudlet v to another cloudlet v_0.
$t_{com}^{master}(r,n,v,v_0)$	The communication delay of transmitting the processed data of worker digital twin $DT_{r,n}$ in cloudlet v to its master digital twin $DT_{r,0}$ in cloudlet v_0.
t_n	The update interval t_n of object n.
$t_{AoI}(r,n,v,v_0)$	The expected AoI of the updated data at the master digital twin $DT_{r,0}$ in cloudlet v_0 from the worker digital twin $DT_{r,n}$ in cloudlet v.
u_{r,n,v,v_0}	The utility gain for the master digital twin $DT_{r,0}$ in cloudlet v_0 to collect data from a worker digital twin $DT_{r,n}$ in cloudlet v.
$w_{r,n}$	The constant coefficient associated with each object $n \in N_r$ for request r.
$\mathcal{U}_r$	The utility gain of admitting DTN slicing request r.
Q_r and D_r^{req}	The set of queries of request r, and the delay requirement of request r.
$t_{proc}^{master}(r,q,v_0)$ and $s_{r,q}$	The processing delay of query $q \in Q_r$ by $DT_{r,0}$ in cloudlet v_0, and the volume of the query result of query r.
$\mathcal{D}_{r,v_0}^{max}$	The maximum query service delay for request r.
$\mathbb{V}_r$	The set of cloudlets for potential placements of $DT_{r,0}$ to meet the delay requirement D_r^{req}.
$x_{r,n,v}$	The binary variable indicating whether the worker or master digital twin $DT_{r,n} \in \mathbb{DT}_r^+$ is deployed in cloudlet $v \in V$ for request r.
$c_{r,n}$	The amount of demanded computing resource of worker or master digital twin $DT_{r,n} \in \mathbb{DT}_r^+$.
C_v	The computing capacity of cloudlet v.
y_r	The binary variable indicating whether to admit request r.
z_{r,n,v,v_0}	The binary variable indicating whether the worker digital twin $DT_{r,n} \in \mathbb{DT}_r$ is deployed in cloudlet $v \in V$, while the master digital twin $DT_{r,0}$ is deployed in cloudlet $v_0 \in \mathbb{V}_r$ for request $r \in R$.
α_v, $\beta_{r,n}$, ψ_{r,v_0}, $\gamma_{r,n,v}$, ϕ_{r,n,v_0} and σ_r	The dual variables, with regard to constraints (14), (15), (16), (20), (21) and (24), respectively.

$v \in V$, there is a bin b_v with a budget C_v, i.e., the computing capacity of cloudlet v. For each worker digital twin $DT_{r,n} \in \mathbb{DT}_r$, there is an item $i_{r,n}$ with weight $c_{r,n}$, i.e., the amount of demanded computing resource for deploying $DT_{r,n}$. Assigning item $i_{r,n}$ to bin b_v leads to a profit U_{r,n,v,v_0} that is defined by (6). We observe this special problem is equivalent to a GAP to maximize the collective profit, which is NP-hard. Hence, the problem of concern is NP-hard. ∎

Corollary 1: The dynamic utility maximization problem for multiple DTN slicing requests is NP-hard.

Proof: By Theorem 1, we have proven that the utility maximization problem for a single DTN slicing request is NP-hard, which is a special case of the dynamic utility maximization problem for multiple DTN slicing requests. Therefore, the problem of concern is also NP-hard. Hence, the corollary follows. ∎

IV. APPROXIMATION ALGORITHM FOR THE UTILITY MAXIMIZATION PROBLEM

In this section, we propose an approximation algorithm for the utility maximization problem for a single DTN slicing request. we first propose the approximation algorithm. Then, we analyze its approximation ratio and time complexity.

A. Approximation Algorithm

The basic idea of the proposed approximation algorithm is to devise an approximate solution for the problem, when placing master digital twin $DT_{r,0}$ in each candidate cloudlet in $\mathbb{V}_r$. We then identify the solution with the placement of $DT_{r,0}$ to achieve the maximum utility gain.

Algorithm 1: Approximation Algorithm for the Utility Maximization Problem for a Single DTN Slicing Request.

Input: An MEC network $G = (V, E)$, a set $\mathcal{N}$ of physical objects, and a single DTN slicing request r.
Output: Maximize the utility gain of the DTN slicing request r.

1: **for** each $v_0 \in \mathbb{V}_r$ with sufficient computing resource for $DT_{r,0}$ **do**
2: Reduce the problem instance when deploying $DT_{r,0}$ in cloudlet v_0 to a GAP instance, via generating an item $i_{r,n}$ with the weight $c_{r,n}$ for each $DT_{r,n} \in \mathbb{DT}_r$, and a bin b_v for each cloudlet with a budget C_v. Assigning item $i_{r,n}$ to bin b_v leads to a profit U_{r,n,v,v_0} by (6);
3: Apply the approximation algorithm from [2] to obtain a solution S_{r,v_0};
4: **end for**
5: **return** Solution $\mathcal{S}_r$ to the problem with the maximum utility gain.

Given the placement of $DT_{r,0}$, we can reduce the utility maximization problem for a single DTN slicing request to a maximum profit Generalized Assignment Problem (GAP), and the approximation algorithm from [2] can be adopted to solve the GAP.

The detailed problem reduction is as follows. Consider each cloudlet $v \in V$, there exists a bin b_v with a budget C_v, i.e., the computing capacity of cloudlet v. Supposing $DT_{r,0}$ is deployed in cloudlet $v_0 \in \mathbb{V}_r$, for each worker digital twin $DT_{r,n} \in \mathbb{DT}_r$, it exists an item $i_{r,n}$ with a weight $c_{r,n}$, i.e., the amount of demanded computing resource of $DT_{r,n}$. A profit U_{r,n,v,v_0} that is defined by (6) will be collected when item $i_{r,n}$ is assigned to bin b_v.

For each cloudlet $v_0 \in \mathbb{V}_r$ with sufficient computing resource for $DT_{r,0}$, we can obtain an approximate solution S_{r,v_0} by [2] for the problem when $DT_{r,0}$ is deployed in cloudlet v_0. We finally identify the solution with the placement of $DT_{r,0}$ to achieve the maximum utility gain. The proposed approximation algorithm is detailed in Algorithm 1.

B. Algorithm Analysis

We now analyze the approximation ratio and time complexity of Algorithm 1 as follows.

Theorem 2: Given an MEC network $G = (V, E)$, a set $\mathcal{N}$ of physical objects, and a request r for a DTN slice consisting of a set $\mathbb{DT}_r$ of worker digital twins and a master digital twin $DT_{r,0}$, there is an approximation algorithm, Algorithm 1, for the utility maximization problem for a single DTN slicing request, with approximation ratio of $\frac{1}{2+\epsilon}$. The time complexity of Algorithm 1 is $O(|\mathbb{V}_r| \cdot |V| \cdot |\mathbb{DT}_r| \cdot \log \frac{1}{\epsilon} + \frac{|\mathbb{V}_r| \cdot |V|}{\epsilon^4})$, where ϵ is a constant with $0 < \epsilon \leq 1$.

Proof: Denote by $\mathcal{V}_r \subseteq \mathbb{V}_r$ the set of cloudlets with sufficient computing resource for $DT_{r,0}$. Denote by OPT_r the optimal solution for the problem, and OPT_{r,v_0} the optimal solution for the problem when $DT_{r,0}$ is deployed in cloudlet v_0. Then $OPT_r = \max\{OPT_{r,v_0} \mid \forall v_0 \in \mathcal{V}_r\}$. For each cloudlet $v_0 \in \mathcal{V}_r$, we obtain an approximate solution S_{r,v_0} for the problem with $DT_{r,0}$ deployed in cloudlet v_0, by the approximation algorithm from [2] with an approximation ratio of $\frac{1}{2+\epsilon}$. Then, we have $S_{r,v_0} \geq \frac{OPT_{r,v_0}}{2+\epsilon}$. Because Algorithm 1 identifies the solution with the placement of $DT_{r,0}$ to achieve the maximum utility gain, its delivered solution is $\max\{S_{r,v_0} \mid \forall v_0 \in \mathcal{V}_r\} \geq \max\{\frac{OPT_{r,v_0}}{2+\epsilon} \mid \forall v_0 \in \mathcal{V}_r\} = \frac{1}{2+\epsilon} \cdot \max\{OPT_{r,v_0} \mid \forall v_0 \in \mathcal{V}_r\} = \frac{OPT_r}{2+\epsilon}$.

The time complexity analysis of Algorithm 1 is given as follows. There are at most $|\mathbb{V}_r|$ cloudlets for the potential deployments of the master digital twin $DT_{r,0}$, and the invoking of the approximation algorithm from [2] for each potential deployment of $DT_{r,0}$ takes $O(|V| \cdot |\mathbb{DT}_r| \cdot \log \frac{1}{\epsilon} + \frac{|V|}{\epsilon^4})$ time. Hence, the time complexity of Algorithm 1 is $O(|\mathbb{V}_r| \cdot |V| \cdot |\mathbb{DT}_r| \cdot \log \frac{1}{\epsilon} + \frac{|\mathbb{V}_r| \cdot |V|}{\epsilon^4})$. ∎

V. Online Algorithm for the Dynamic Utility Maximization Problem

In this section, we consider the dynamic utility maximization problem. We first perform linearization on the INP (13) of the offline version of the problem to obtain an Integer Linear Program (ILP). We then develop an online algorithm for the problem by adopting the primal-dual dynamic updating technique [4].

A. ILP Formulation

To remove the non-linearity of the objective function (13), we introduce another binary variable z_{r,n,v,v_0} to replace $x_{r,n,v} \cdot x_{r,0,v_0}$, i.e., $z_{r,n,v,v_0} = 1$ means that the worker digital twin $DT_{r,n} \in \mathbb{DT}_r$ is deployed in cloudlet $v \in V$, while the master digital twin $DT_{r,0}$ is deployed in cloudlet $v_0 \in \mathbb{V}_r$ for request $r \in R$. Otherwise, $z_{r,n,v,v_0} = 0$. Denote by **P1** the dynamic utility maximization problem for multiple DTN slicing requests. The ILP of its offline version then is presented as follows.

$$\textbf{P1}: \text{Maximize} \sum_{r \in R} \sum_{DT_{r,n} \in \mathbb{DT}_r} \sum_{v \in V} \sum_{v_0 \in \mathbb{V}_r} U_{r,n,v,v_0} \cdot z_{r,n,v,v_0}, \tag{19}$$

subject to:

Constraints (14), (15), (16), (17) and (18)

$$\sum\nolimits_{v_0 \in \mathbb{V}_r} z_{r,n,v,v_0} \leq x_{r,n,v}, \forall r \in R, \forall DT_{r,n} \in \mathbb{DT}_r, \forall v \in V \tag{20}$$

$$\sum\nolimits_{v \in V} z_{r,n,v,v_0} \leq x_{r,0,v_0}, \forall r \in R, \forall DT_{r,n} \in \mathbb{DT}_r, \forall v_0 \in \mathbb{V}_r \tag{21}$$

$$z_{r,n,v,v_0} \in \{0,1\}, \forall r \in R, \forall DT_{r,n} \in \mathbb{DT}_r, \forall v \in V, \forall v_0 \in \mathbb{V}_r, \tag{22}$$

where (20) and (21) guarantee that z_{r,n,v,v_0} is functionally equivalent to $x_{r,n,v} \cdot x_{r,0,v_0}$, which will be shown in Lemma 1.

B. Online Algorithm

Now we adopt the primal-dual dynamic updating technique to develop an online algorithm for the dynamic utility maximization problem.

Denote by **P2** the linear relaxation of **P1**, where binary variables are relaxed into real variables between 0 and 1. An online algorithm for **P1** can be derived from a feasible solution to the dual of **P2**, which is denoted by **P3**.

The Linear Program (LP) for **P2** is formulated as follows.

$$\mathbf{P2}: \text{Maximize} \sum_{r\in R}\sum_{DT_{r,n}\in\mathbb{DT}_r}\sum_{v\in V}\sum_{v_0\in\mathbb{V}_r} U_{r,n,v,v_0}\cdot z_{r,n,v,v_0}, \tag{23}$$

subject to:

Constraints (14), (15), (16), (17), (18), (20) and (21)

$$y_r \le 1, \quad \forall r\in R \tag{24}$$

$$x_{r,n,v}\ge 0, \quad \forall r\in R,\ \ \forall DT_{r,n}\in\mathbb{DT}_r^+,\ \forall v\in V \tag{25}$$

$$y_r\ge 0, \quad \forall r\in R, \tag{26}$$

$$z_{r,n,v,v_0}\ge 0,\ \forall r\in R, \forall DT_{r,n}\in\mathbb{DT}_r, \forall v\in V, \forall v_0\in\mathbb{V}_r, \tag{27}$$

where constraints (15) and (24) indicate $x_{r,n,v}\le 1$. Because $x_{r,n,v}\le 1$, constraints (20) and (21) indicate $z_{r,n,v,v_0}\le 1$.

The dual problem **P3** of **P2** is formulated as follows.

$$\mathbf{P3}: \text{Minimize} \sum_{v\in V} C_v\cdot\alpha_v + \sum_{r\in R}\sigma_r, \tag{28}$$

subject to:

$$\alpha_v\cdot c_{r,n} + \beta_{r,n} - \gamma_{r,n,v}\ge 0, \forall r\in R, \forall DT_{r,n}\in\mathbb{DT}_r, \forall v\in V \tag{29}$$

$$\alpha_{v_0}\cdot c_{r,0} + \beta_{r,0} - \sum_{DT_{r,n}\in\mathbb{DT}_r}\phi_{r,n,v_0}\ge 0, \forall r\in R, \forall v_0\in\mathbb{V}_r \tag{30}$$

$$\alpha_{v_0}\cdot c_{r,0} + \beta_{r,0} + \psi_{r,v_0}\ge 0, \quad \forall r\in R, \forall v_0\in V\backslash\mathbb{V}_r \tag{31}$$

$$\sigma_r - \sum_{DT_{r,n}\in\mathbb{DT}_r^+}\beta_{r,n}\ge 0, \quad \forall r\in R \tag{32}$$

$$\gamma_{r,n,v} + \phi_{r,n,v_0} - U_{r,n,v,v_0}\ge 0,\quad \forall r\in R,\ \forall DT_{r,n}\in\mathbb{DT}_r,\ \forall v\in V,\ \forall v_0\in\mathbb{V}_r \tag{33}$$

$$\alpha_v\ge 0,\ \gamma_{r,n,v}\ge 0,\ \phi_{r,n,v_0}\ge 0,\ \sigma_r\ge 0,\quad \forall r\in R,\ \forall DT_{r,n}\in\mathbb{DT}_r,\ \forall v\in V,\ \forall v_0\in\mathbb{V}_r, \tag{34}$$

where α_v, $\beta_{r,n}$, ψ_{r,v_0}, $\gamma_{r,n,v}$, ϕ_{r,n,v_0} and σ_r are dual variables, with regard to constraints (14), (15), (16), (20), (21) and (24), respectively. α_v, $\gamma_{r,n,v}$, ϕ_{r,n,v_0} and σ_r are non-negative shown in constraint (34), while $\beta_{r,n}$, $\forall r\in R$, $\forall DT_{r,n}\in\mathbb{DT}_r^+$, and ψ_{r,v_0}, $\forall r\in R$, $\forall v_0\in V\backslash\mathbb{V}_r$, are unconstrained.

From (29), we have $\forall r\in R, \forall DT_{r,n}\in\mathbb{DT}_r, \forall v\in V$,

$$\gamma_{r,n,v}\le\alpha_v\cdot c_{r,n}+\beta_{r,n}. \tag{35}$$

From (33), we set $\phi_{r,n,v_0}=0$, and we have $\forall r\in R$, $\forall DT_{r,n}\in\mathbb{DT}_r$, $\forall v\in V$,

$$\gamma_{r,n,v}\ge\frac{1}{|\mathbb{V}_r|}\cdot\sum_{v_0\in\mathbb{V}_r}U_{r,n,v,v_0}. \tag{36}$$

By (35) and (36), we have $\forall r\in R, \forall DT_{r,n}\in\mathbb{DT}_r, \forall v\in V$,

$$\beta_{r,n}\ge\frac{1}{|\mathbb{V}_r|}\cdot\sum_{v_0\in\mathbb{V}_r}U_{r,n,v,v_0}-\alpha_v\cdot c_{r,n}. \tag{37}$$

Therefore, we have $\forall r\in R$,

$$\sum_{DT_{r,n}\in\mathbb{DT}_r}\beta_{r,n}\ge\frac{1}{|V|\cdot|\mathbb{V}_r|}\cdot\sum_{DT_{r,n}\in\mathbb{DT}_r}\sum_{v\in V}\sum_{v_0\in\mathbb{V}_r}U_{r,n,v,v_0} - \frac{1}{|V|}\cdot\sum_{DT_{r,n}\in\mathbb{DT}_r}\sum_{v\in V}\alpha_v\cdot c_{r,n}. \tag{38}$$

As we set $\phi_{r,n,v_0}=0$, by (30), $\forall r\in R$, $\forall v_0\in\mathbb{V}_r$, we have

$$\beta_{r,0}\ge-\alpha_{v_0}\cdot c_{r,0}. \tag{39}$$

From (31), we set $\psi_{r,v_0}=0$. Then $\forall r\in R$, $\forall v_0\in V\backslash\mathbb{V}_r$,

$$\beta_{r,0}\ge-\alpha_{v_0}\cdot c_{r,0}. \tag{40}$$

Combing Ineq. (39) and (40), we have $\forall r\in R$,

$$\beta_{r,0}\ge-\frac{1}{|V|}\sum_{v\in V}\alpha_v\cdot c_{r,0}. \tag{41}$$

Combining Ineq. (38) and (41), we have

$$\sum_{DT_{r,n}\in\mathbb{DT}_r^+}\beta_{r,n}\ge\frac{1}{|V|\cdot|\mathbb{V}_r|}\cdot\sum_{DT_{r,n}\in\mathbb{DT}_r}\sum_{v\in V}\sum_{v_0\in\mathbb{V}_r}U_{r,n,v,v_0} - \frac{1}{|V|}\cdot\sum_{DT_{r,n}\in\mathbb{DT}_r^+}\sum_{v\in V}\alpha_v\cdot c_{r,n}. \tag{42}$$

Considering Ineq. (32) and (42), we have $\forall r\in R$,

$$\sigma_r\ge\frac{1}{|V|\cdot|\mathbb{V}_r|}\cdot\sum_{DT_{r,n}\in\mathbb{DT}_r}\sum_{v\in V}\sum_{v_0\in\mathbb{V}_r}U_{r,n,v,v_0} - \frac{1}{|V|}\cdot\sum_{DT_{r,n}\in\mathbb{DT}_r^+}c_{r,n}\cdot\sum_{v\in V}\alpha_v. \tag{43}$$

For the sake of convenience, we define constants μ_r and ρ_r as follows.

$$\mu_r=\frac{1}{|V|\cdot|\mathbb{V}_r|}\cdot\sum_{DT_{r,n}\in\mathbb{DT}_r}\sum_{v\in V}\sum_{v_0\in\mathbb{V}_r}U_{r,n,v,v_0}. \tag{44}$$

$$\rho_r=\frac{1}{|V|}\cdot\sum_{DT_{r,n}\in\mathbb{DT}_r^+}c_{r,n}. \tag{45}$$

From Ineq. (43), we have

$$\sigma_r\ge\mu_r-\rho_r\cdot\sum_{v\in V}\alpha_v,\quad \forall r\in R. \tag{46}$$

By setting α_v and σ_r subject to Ineq. (46), $\beta_{r,n}$, ψ_{r,v_0}, $\gamma_{r,n,v}$ and ϕ_{r,n,v_0} can be identified to deliver a feasible solution to **P3**, which will be shown in Lemma 2.

The online algorithm for **P1** is proposed by updating variables in the primal and dual programs simultaneously. Especially, we set $\alpha_v, \forall v \in V$, as 0s initially. Considering Ineq. (46), we devise an admission control policy for each arrived request r as follows.

If $\mu_r - \rho_r \cdot \sum_{v\in V} \alpha_v \leq 0$, request r is rejected, while we set $\sigma_r = 0$ and the values of $\alpha_v, \forall v \in V$, do not change. Otherwise, request r is admitted, and we set σ_r as follows.

$$\sigma_r = \mu_r - \rho_r \cdot \sum\nolimits_{v\in V} \alpha_v. \tag{47}$$

In the following, we show how to update α_v for admitting request r. Given residual capacities on cloudlets, we first invoke Algorithm 1 to build the DTN slice of request r, i.e., identify $\{x_{r,n,v}, \mid \forall DT_{r,n} \in \mathbb{DT}_r^+, \forall v \in V\}$, where $x_{r,n,v}$ is a binary variable defined in (17). Then the amount $\mathcal{C}_{r,v}$ $(= \sum_{DT_{r,n}\in\mathbb{DT}_r^+} c_{r,n} \cdot x_{r,n,v})$ of computing resource will be consumed by deploying digital twins in cloudlet v for request r. Notice that Algorithm 1 may fail to deliver a feasible solution for request r, when the cloudlets have insufficient computing resource. In such a case, we invoke Algorithm 1 for request r, assuming cloudlets are idle, i.e., no computing resource is consumed prior to the arrival of request r. Similarly, we can obtain $\mathcal{C}_{r,v}$ in this case, however, it will cause resource violations on cloudlets, and we will bound resource violations on cloudlets later in Lemma 4. We now update α_v for each cloudlet $v \in V$ with $\mathcal{C}_{r,v} > 0$ as follows.

$$\alpha_v = \alpha_v \cdot \left(1 + \frac{\rho_r \cdot \mathcal{C}_{r,v}}{C_v \cdot \sum_{DT_{r,n}\in\mathbb{DT}_r^+} c_{r,n}}\right) + \frac{\rho_r \cdot \mathcal{C}_{r,v}}{C_v \cdot \sum_{DT_{r,n}\in\mathbb{DT}_r^+} c_{r,n}}. \tag{48}$$

The online algorithm is detailed in Algorithm 2.

Algorithm 2: An Online Algorithm for the Dynamic Utility Maximization Problem for Multiple DTN Slicing Requests.

Input: The MEC network $G = (V, E)$, a set $\mathcal{N}$ of objects, and a sequence R of DTN slicing requests arriving one by one without future knowledge.
Output: Maximize the accumulative utility gain of admitted requests.

1: $\alpha_v \leftarrow 0, \forall v \in V$;
2: **while** a DTN slicing request r arrives **do**
3: **if** $\mu_r - \rho_r \cdot \sum_{v\in V} \alpha_v \leq 0$ **then**
4: Reject request r, and $\sigma_r \leftarrow 0$;
5: **else**
6: Admit request r, and update σ_r by (47);
7: Construct a problem instance **P1** for the single request r, and invoke Algorithm 1 to obtain a solution $\mathcal{S}_r$ for the deployment of digital twins for request r, with the residual capacities on cloudlets;
8: **if** the solution $\mathcal{S}_r$ is infeasible **then**
9: Update solution $\mathcal{S}_r$, by invoking Algorithm 1 for request r, assuming cloudlets are idle;
10: **end if**
11: Deploy digital twins on cloudlets by $\mathcal{S}_r$, and identify the amount $\mathcal{C}_{r,v}$ of consumed computing resource on each cloudlet v by $\mathcal{S}_r$;
12: **for** each cloudlet $v \in V$ with $\mathcal{C}_{r,v} > 0$ **do**
13: Update α_v by (48);
14: **end for**;
15: **end if**;
16: **end while**;

C. Algorithm Analysis

Lemma 1: The binary variable z_{r,n,v,v_0} is functionally equivalent to $x_{r,n,v} \cdot x_{r,0,v_0}$, $\forall r \in R$, $\forall DT_{r,n} \in \mathbb{DT}_r$, $\forall v \in V$, $\forall v_0 \in \mathbb{V}_r$, through imposing constraints (20) and (21).

Proof: The lemma is proved by distinguishing into four cases as follows.

Case 1: $x_{r,n,v} = 0$ and $x_{r,0,v_0} = 0$. Then $\sum_{v_0\in\mathbb{V}_r} z_{r,n,v,v_0} \leq 0$ by (20), and $\sum_{v\in V} z_{r,n,v,v_0} \leq 0$ by (21). As z_{r,n,v,v_0} is a binary variable, we have $z_{r,n,v,v_0} = 0$.

Case 2: $x_{r,n,v} = 0$ and $x_{r,0,v_0} = 1$. Then $\sum_{v_0\in\mathbb{V}_r} z_{r,n,v,v_0} \leq 0$ by (20), and we have $z_{r,n,v,v_0} = 0$.

Case 3: $x_{r,n,v} = 1$ and $x_{r,0,v_0} = 0$. Then $\sum_{v\in V} z_{r,n,v,v_0} \leq 0$ by (21), and we have $z_{r,n,v,v_0} = 0$.

Case 4: $x_{r,n,v} = 1$ and $x_{r,0,v_0} = 1$. Then $\sum_{v_0\in\mathbb{V}_r} z_{r,n,v,v_0} \leq 1$ by (20), and $\sum_{v\in V} z_{r,n,v,v_0} \leq 1$ by (21). The former means that given a worker digital twin $DT_{r,n}$ deployed in cloudlet v $(x_{r,n,v} = 1)$, its data is sent to at most one cloudlet $v_0 \in \mathbb{V}_r$ to build the master digital twin $DT_{r,0}$. The latter means that given $DT_{r,0}$ in cloudlet v_0 $(x_{r,0,v_0} = 1)$, it retrieves the data of a worker digital twin $DT_{r,n}$ from at most one cloudlet $v \in V$. Because the objective (19) is to maximize the accumulative utility gain, it will implicitly set $z_{r,n,v,v_0} = 1$, i.e., the worker digital twin $DT_{r,n}$ is deployed in a cloudlet v, while $DT_{r,0}$ is deployed in cloudlet v_0. ■

Lemma 2: (1) With $\alpha_v \geq 0$ and $\sigma_r \geq 0$ subject to Ineq. (46), there always exist feasible $\beta_{r,n}$, ψ_{r,v_0}, $\gamma_{r,n,v}$ and ϕ_{r,n,v_0} to deliver a feasible solution to **P3**; and (2) Algorithm 2 delivers a feasible solution to **P3**.

Proof: (1) Recall that constraint (34) shows $\alpha_v, \gamma_{r,n,v}, \phi_{r,n,v_0}$ and σ_r are non-negative, while $\beta_{r,n}, \forall r \in R$, $\forall DT_{r,n} \in \mathbb{DT}_r^+$, and $\psi_{r,v_0}, \forall r \in R$, $\forall v_0 \in V\backslash\mathbb{V}_r$, are unconstrained.

Ineq. (46) is derived from constraint (32) and Ineq. (42), while Ineq. (42) is derived from Ineq. (38) and (41). Therefore, given feasible α_v and σ_r subject to Ineq. (46), there always exists feasible $\beta_{r,n}$ meeting constraint (32), Ineq. (38) and (41). Because of feasible α_v and $\beta_{r,n}$ to meet Ineq. (41), we set $\phi_{r,n,v_0} = 0$, $\forall r \in R$, $\forall DT_{r,n} \in \mathbb{DT}_r$, $\forall v_0 \in \mathbb{V}_r$, to meet constraint (30), and we set $\psi_{r,v_0} = 0$, $\forall r \in R$, $\forall v_0 \in V\backslash\mathbb{V}_r$, to meet constraint (31). Similarly, because Ineq. (38) is derived from constraint (29) and (33), given feasible α_v and $\beta_{r,n}$ to meet Ineq. (38), there always exists feasible $\gamma_{r,n,v}$, $\forall r \in R, \forall DT_{r,n} \in \mathbb{DT}_r$, $\forall v \in V$, $\forall v_0 \in \mathbb{V}_r$, to meet constraint (29) and (33).

Having obtained α_v and σ_r, subject to Ineq. (46), there always exist feasible $\beta_{r,n}$, ψ_{r,v_0}, $\gamma_{r,n,v}$ and ϕ_{r,n,v_0} to meet constraints (29), (30), (31), (32), (33), and (34), thereby delivering a feasible solution to **P3**.

(2) The rest is to show that Ineq. (46) always holds when updating α_v and σ_r for each incoming request r. If request r is

rejected, i.e., $\mu_r - \rho_r \cdot \sum_{v\in V} \alpha_v \leq 0$, we set $\sigma_r = 0$. Otherwise (request r is admitted), σ_r is updated by the update function (47). Because of the non-decreasing nature of the update function (48) of α_v, the function $\mu_r - \rho_r \cdot \sum_{v\in V} \alpha_v$ is also non-increasing with the updating of α_v. This means Ineq. (46) always holds when updating α_v and σ_r for each incoming request r. Therefore, when we update α_v and σ_r subject to Ineq. (46), $\beta_{r,n}$, ψ_{r,v_0}, $\gamma_{r,n,v}$ and ϕ_{r,n,v_0} can be obtained, and a feasible solution to **P3** is obtained, i.e., Algorithm 2 delivers a feasible solution to **P3**. ■

Let $\mu_{\max}$ and $\mu_{\min}$ be the maximum and minimum values of μ_r by (44) and $\rho_{\max}$ and $\rho_{\min}$ the maximum and minimum values of ρ_r by (45), respectively. Let $C_{\max}$ and $C_{\min}$ be the maximum and minimum computing capacities on any cloudlet, respectively. Let $c_{\max}$ and $c_{\min}$ be the maximum and minimum amounts of computing resources consumed by any digital twin, and $|\mathbb{DT}^+|_{\max}$ is the maximum number of digital twins in a DTN slice.

Denote by $\hat{x}_{r,v}$ a binary variable. When $\hat{x}_{r,v} = 1$, it means at least a worker or master digital twin is deployed in cloudlet v for request r; otherwise $\hat{x}_{r,v} = 0$.

Lemma 3: When updating the dual variable α_v by Algorithm 2, we have, $\forall v \in V$,

$$\alpha_v \geq \left(1 + \frac{\rho_{\min} \cdot c_{\min}}{C_{\max} \cdot |\mathbb{DT}^+|_{\max} \cdot c_{\max}}\right)^{\sum_{r\in R} \hat{x}_{r,v}} - 1; \quad (49)$$

$$\alpha_v < \frac{\mu_{\max}}{\rho_{\min}} \cdot \left(1 + \frac{\rho_{\max}}{C_{\min}}\right) + \frac{\rho_{\max}}{C_{\min}}. \quad (50)$$

Proof: We first show that Ineq. (49) holds by induction. The right hand side of Ineq. (49) is 0 before requests arrive. Therefore, the induction hypothesis holds initially, with $\alpha_v = 0$ initially. Let $\alpha_v(before)$ and $\alpha_v(after)$ indicate α_v before and after the arrival of request r'. We now show that Ineq. (49) holds by the induction, considering whether α_v is updated or not.

Case 1. α_v is not updated, i.e., $\hat{x}_{r',v} = 0$. This is due to rejecting request r', or admitting request r' with no digital twin deployed in cloudlet v. Because α_v is not updated, i.e., $\alpha_v(after) = \alpha_v(before)$, then

$$\begin{aligned}
&\alpha_v(after) = \alpha_v(before)\\
&\geq \left(1 + \frac{\rho_{\min} \cdot c_{\min}}{C_{\max} \cdot |\mathbb{DT}^+|_{\max} \cdot c_{\max}}\right)^{\sum_{r\in R\setminus\{r'\}} \hat{x}_{r,v}} - 1\\
&\geq \left(1 + \frac{\rho_{\min} \cdot c_{\min}}{C_{\max} \cdot |\mathbb{DT}^+|_{\max} \cdot c_{\max}}\right)^{\sum_{r\in R\setminus\{r'\}} \hat{x}_{r,v}}\\
&\quad \cdot \left(1 + \frac{\rho_{\min} \cdot c_{\min}}{C_{\max} \cdot |\mathbb{DT}^+|_{\max} \cdot c_{\max}}\right)^{\hat{x}_{r',v}} - 1, \text{by } \hat{x}_{r',v} = 0\\
&= \left(1 + \frac{\rho_{\min} \cdot c_{\min}}{C_{\max} \cdot |\mathbb{DT}^+|_{\max} \cdot c_{\max}}\right)^{\sum_{r\in R} \hat{x}_{r,v}} - 1. \quad (51)
\end{aligned}$$

Case 2. α_v is updated, i.e., $\hat{x}_{r',v} = 1$. This is due to admitting request r' with at least a digital twin deployed in cloudlet v. By the update function (48),

$$\begin{aligned}
\alpha_v(after) &= \alpha_v(before) \cdot \left(1 + \frac{\rho_{r'} \cdot \mathcal{C}_{r,v}}{C_v \cdot \sum_{DT_{r,n}\in\mathbb{DT}_r^+} c_{r,n}}\right)\\
&\quad + \frac{\rho_{r'} \cdot \mathcal{C}_{r,v}}{C_v \cdot \sum_{DT_{r,n}\in\mathbb{DT}_r^+} c_{r,n}}\\
&\geq \alpha_v(before) \cdot \left(1 + \frac{\rho_{\min} \cdot c_{\min}}{C_{\max} \cdot |\mathbb{DT}^+|_{\max} \cdot c_{\max}}\right)\\
&\quad + \frac{\rho_{\min} \cdot c_{\min}}{C_{\max} \cdot |\mathbb{DT}^+|_{\max} \cdot c_{\max}}\\
&\geq \left(\left(1 + \frac{\rho_{\min} \cdot c_{\min}}{C_{\max} \cdot |\mathbb{DT}^+|_{\max} \cdot c_{\max}}\right)^{\sum_{r\in R\setminus\{r'\}} \hat{x}_{r,v}} - 1\right)\\
&\quad \cdot \left(1 + \frac{\rho_{\min} \cdot c_{\min}}{C_{\max} \cdot |\mathbb{DT}^+|_{\max} \cdot c_{\max}}\right)\\
&\quad + \frac{\rho_{\min} \cdot c_{\min}}{C_{\max} \cdot |\mathbb{DT}^+|_{\max} \cdot c_{\max}}, \quad \text{by Ineq. (49)}\\
&\geq \left(1 + \frac{\rho_{\min} \cdot c_{\min}}{C_{\max} \cdot |\mathbb{DT}^+|_{\max} \cdot c_{\max}}\right)^{\sum_{r\in R\setminus\{r'\}} \hat{x}_{r,v}}\\
&\quad \cdot \left(1 + \frac{\rho_{\min} \cdot c_{\min}}{C_{\max} \cdot |\mathbb{DT}^+|_{\max} \cdot c_{\max}}\right)^{\hat{x}_{r',v}} - 1, \text{by } \hat{x}_{r',v} = 1\\
&= \left(1 + \frac{\rho_{\min} \cdot c_{\min}}{C_{\max} \cdot |\mathbb{DT}^+|_{\max} \cdot c_{\max}}\right)^{\sum_{r\in R} \hat{x}_{r,v}} - 1. \quad (52)
\end{aligned}$$

We then show that Ineq. (50) holds. α_v is 0 before requests arrive, and a new request r is admitted with $\mu_r - \rho_r \cdot \sum_{v\in V} \alpha_v > 0$. Then

$$\mu_r - \rho_r \cdot \sum_{v\in V} \alpha_v > 0 \Rightarrow \sum_{v\in V} \alpha_v < \frac{\mu_r}{\rho_r} \Rightarrow \alpha_v < \frac{\mu_{\max}}{\rho_{\min}}.$$

This means that if $\alpha_v \geq \frac{\mu_{\max}}{\rho_{\min}}$, α_v will not be updated; otherwise, α_v is likely to be updated by function (48).

$$\begin{aligned}
\alpha_v &= \alpha_v \cdot \left(1 + \frac{\rho_r \cdot \mathcal{C}_{r,v}}{C_v \cdot \sum_{DT_{r,n}\in\mathbb{DT}_r^+} c_{r,n}}\right)\\
&\quad + \frac{\rho_r \cdot \mathcal{C}_{r,v}}{C_v \cdot \sum_{DT_{r,n}\in\mathbb{DT}_r^+} c_{r,n}} < \frac{\mu_{\max}}{\rho_{\min}}\\
&\quad \cdot \left(1 + \frac{\rho_{\max}}{C_{\min}}\right) + \frac{\rho_{\max}}{C_{\min}}, \quad (53)
\end{aligned}$$

where Ineq. (53) holds, because the consumed computing resource $\mathcal{C}_{r,v}$ on a cloudlet v of deploying digital twins of request r is at most that of all digital twins of request r, i.e., $\mathcal{C}_{r,v} \leq \sum_{DT_{r,n}\in\mathbb{DT}_r^+} c_{r,n}$. ■

Lemma 4: Given a solution delivered by Algorithm 2 for **P1**, the resource violation on each cloudlet is upper bounded by ξ, where $\xi = \frac{|\mathbb{DT}^+|_{\max} \cdot c_{\max}}{C_{\min}} \cdot \frac{\ln(\frac{\mu_{\max}}{\rho_{\min}} \cdot (1+\frac{\rho_{\max}}{C_{\min}}) + \frac{\rho_{\max}}{C_{\min}} + 1)}{\ln(1+\frac{\rho_{\min} \cdot c_{\min}}{C_{\max} \cdot |\mathbb{DT}^+|_{\max} \cdot c_{\max}})} - 1$.

Proof: From Lemma 3, we have

$$\left(1 + \frac{\rho_{\min} \cdot c_{\min}}{C_{\max} \cdot |\mathbb{DT}^+|_{\max} \cdot c_{\max}}\right)^{\sum_{r\in R} \hat{x}_{r,v}} - 1$$

$$< \frac{\mu_{\max}}{\rho_{\min}} \cdot \left(1 + \frac{\rho_{\max}}{C_{\min}}\right) + \frac{\rho_{\max}}{C_{\min}}.$$

We then have

$$\sum\nolimits_{r \in R} \hat{x}_{r,v} < \frac{\ln\left(\frac{\mu_{\max}}{\rho_{\min}} \cdot (1 + \frac{\rho_{\max}}{C_{\min}}) + \frac{\rho_{\max}}{C_{\min}} + 1\right)}{\ln\left(1 + \frac{\rho_{\min} \cdot c_{\min}}{C_{\max} \cdot |\mathbb{DT}^+|_{\max} \cdot c_{\max}}\right)}$$

The computing resource utilized in cloudlet v is

$$\sum_{r \in R} \sum_{DT_{r,n} \in \mathbb{DT}_r^+} c_{r,n} \cdot x_{r,n,v} \le |\mathbb{DT}^+|_{\max} \cdot c_{\max} \cdot \sum_{r \in R} \hat{x}_{r,v}$$

$$< |\mathbb{DT}^+|_{\max} \cdot c_{\max} \cdot \frac{\ln\left(\frac{\mu_{\max}}{\rho_{\min}} \cdot \left(1 + \frac{\rho_{\max}}{C_{\min}}\right) + \frac{\rho_{\max}}{C_{\min}} + 1\right)}{\ln\left(1 + \frac{\rho_{\min} \cdot c_{\min}}{C_{\max} \cdot |\mathbb{DT}^+|_{\max} \cdot c_{\max}}\right)}.$$

By constraint (14), the resource violation on a cloudlet is no more than ξ, which is defined in this lemma. ■

With the utility definition (6), let $U_{\max}$ and $U_{\min}$ be the maximum and minimum values of U_{r,n,v,v_0}, respectively. Let $|\mathbb{V}|_{\max}$ be the maximum number of candidate cloudlets for deploying the master digital twin of a request.

Lemma 5: Given a solution delivered by Algorithm 2 for **P1**, suppose a DTN slicing request r is admitted by Algorithm 2 with the utility gain $\mathcal{U}_r$ defined in (7), we have $\mu_r \le \frac{U_{\max}}{U_{\min}} \cdot \mathcal{U}_r$.

Proof: Considering the admission of request r by Algorithm 2, suppose each of its worker digital twin $DT_{r,n}$ is deployed in cloudlet $v_{r,n}$, respectively, while its master digital twin $DT_{r,0}$ is deployed in cloudlet $v_{r,0}$, by Algorithm 2. By (7), the utility gain is $\mathcal{U}_r = \sum_{DT_{r,n} \in \mathbb{DT}_r} U_{r,n,v_{r,n},v_{r,0}}$. With regard to the definition of μ_r by (44), we have

$$\mu_r = \frac{1}{|V| \cdot |\mathbb{V}_r|} \cdot \sum_{DT_{r,n} \in \mathbb{DT}_r} \sum_{v \in V} \sum_{v_0 \in \mathbb{V}_r} U_{r,n,v,v_0}$$

$$\le \frac{1}{|V| \cdot |\mathbb{V}_r|} \cdot \sum_{DT_{r,n} \in \mathbb{DT}_r} \sum_{v \in V} \sum_{v_0 \in \mathbb{V}_r} \frac{U_{\max}}{U_{\min}} \cdot U_{r,n,v_{r,n},v_{r,0}}$$

$$= \frac{U_{\max}}{U_{\min}} \cdot \sum_{DT_{r,n} \in \mathbb{DT}_r} U_{r,n,v_{r,n},v_{r,0}} = \frac{U_{\max}}{U_{\min}} \cdot \mathcal{U}_r. \quad ■$$

Lemma 6: Denote by $\mathbb{S}_1$ and $\mathbb{S}_3$ the solution values by Algorithm 2 for **P1** and **P3**, respectively. We have $\frac{U_{\max}}{U_{\min}} \cdot (1 + \frac{\rho_{\max}}{\mu_{\min}}) \cdot \mathbb{S}_1 \ge \mathbb{S}_3$, where $\mu_{\min}$ is the minimum value of μ_r by (44), and $\rho_{\max}$ is the maximum value of ρ_r by (45).

Proof: Because $\mathbb{S}_1 = \mathbb{S}_3 = 0$, the claim holds initially. Let $\Delta\mathbb{S}_1$ and $\Delta\mathbb{S}_3$ be the value differences of $\mathbb{S}_1$ and $\mathbb{S}_3$ after the rejection or admission of an incoming request r, respectively. We will show that the claim holds through showing $\frac{U_{\max}}{U_{\min}} \cdot (1 + \frac{\rho_{\max}}{\mu_{\min}}) \cdot \Delta\mathbb{S}_1 \ge \Delta\mathbb{S}_3$ for each request r as follows.

Case 1. Request r is rejected. Because $\Delta\mathbb{S}_1 = \Delta\mathbb{S}_3 = 0$, we have $\frac{U_{\max}}{U_{\min}} \cdot (1 + \frac{\rho_{\max}}{\mu_{\min}}) \cdot \Delta\mathbb{S}_1 \ge \Delta\mathbb{S}_3$.

Case 2. Request r is admitted. We have $\Delta\mathbb{S}_1 = \mathcal{U}_r$, i.e., the utility gain defined in (6), and $\Delta\mathbb{S}_3 = \sum_{v \in V} C_v \cdot \Delta\alpha_v + \sigma_r$ by the objective function (28), where $\Delta\alpha_v$ is the value difference of α_v before and after updating α_v. By update functions (48) and (47) of α_v and ρ_r, we have

$$\Delta\mathbb{S}_3 = \sum\nolimits_{v \in V} C_v \cdot \left(\frac{\rho_r \cdot \mathcal{C}_{r,v}}{C_v \cdot \sum_{DT_{r,n} \in \mathbb{DT}_r^+} c_{r,n}} \cdot \alpha_v + \frac{\rho_r \cdot \mathcal{C}_{r,v}}{C_v \cdot \sum_{DT_{r,n} \in \mathbb{DT}_r^+} c_{r,n}}\right) + \sigma_r$$

$$= \rho_r \cdot \frac{\sum_{v \in V} \mathcal{C}_{r,v} \cdot \alpha_v}{\sum_{DT_{r,n} \in \mathbb{DT}_r^+} c_{r,n}} + \rho_r \cdot \frac{\sum_{v \in V} \mathcal{C}_{r,v}}{\sum_{DT_{r,n} \in \mathbb{DT}_r^+} c_{r,n}} + \mu_r - \rho_r \cdot \sum\nolimits_{v \in V} \alpha_v$$

$$\le \mu_r + \rho_r \tag{54}$$

$$\le \mu_r \cdot (1 + \frac{\rho_r}{\mu_r}) \le \frac{U_{\max}}{U_{\min}} \cdot \mathcal{U}_r \cdot \left(1 + \frac{\rho_{\max}}{\mu_{\min}}\right)$$

$$\le \frac{U_{\max}}{U_{\min}} \cdot \left(1 + \frac{\rho_{\max}}{\mu_{\min}}\right) \cdot \Delta\mathbb{S}_1, \tag{55}$$

where (54) holds as $\sum_{v \in V} \mathcal{C}_{r,v} = \sum_{DT_{r,n} \in \mathbb{DT}_r^+} c_{r,n}$, i.e., $\rho_r \cdot \frac{\sum_{v \in V} \mathcal{C}_{r,v}}{\sum_{DT_{r,n} \in \mathbb{DT}_r^+} c_{r,n}} = \rho_r$. As $\mathcal{C}_{r,v} \le \sum_{DT_{r,n} \in \mathbb{DT}_r^+} c_{r,n}$, then $\rho_r \cdot \frac{\sum_{v \in V} \mathcal{C}_{r,v} \cdot \alpha_v}{\sum_{DT_{r,n} \in \mathbb{DT}_r^+} c_{r,n}} \le \rho_r \cdot \sum_{v \in V} \alpha_v$. (55) holds by Lemma 5. ■

Theorem 3: Given an MEC network $G = (V, E)$, a set $\mathcal{N}$ of objects, and a sequence R of DTN slicing requests arriving one by one, there is an online algorithm, Algorithm 2, with the competitive ratio of $(\frac{U_{\max}}{U_{\min}} \cdot (1 + \frac{\rho_{\max}}{\mu_{\min}}))$ for the dynamic utility maximization problem for multiple DTN slicing requests. The resource violation on a cloudlet is no larger than ξ defined in Lemma 4, and it takes $O(|\mathbb{V}_r| \cdot |V| \cdot |\mathbb{DT}_r| \cdot \log \frac{1}{\epsilon} + \frac{|\mathbb{V}_r| \cdot |V|}{\epsilon^4})$ time to admit each request r.

Proof: Let $\mathcal{OPT}(P1)$, $\mathcal{OPT}(P2)$ and $\mathcal{OPT}(P3)$ be the optimal solution values of **P1**, **P2**, and **P3**, respectively. Let $\mathbb{S}_1$ and $\mathbb{S}_3$ be the solution values by Algorithm 2 for **P1** and **P3**, respectively. We have $\frac{U_{\max}}{U_{\min}} \cdot (1 + \frac{\rho_{\max}}{\mu_{\min}}) \cdot \mathbb{S}_1 \ge \mathbb{S}_3$ by Lemma 6. Because Algorithm 2 delivers a feasible solution to the minimization problem **P3** by Lemma 2, then $\mathbb{S}_3 \ge \mathcal{OPT}(P3)$. Because **P3** is the dual of **P2**, then $\mathcal{OPT}(P3) \ge \mathcal{OPT}(P2)$ by weak duality. As **P2** is the relaxation of the maximization problem **P1**, we have $\mathcal{OPT}(P2) \ge \mathcal{OPT}(P1)$. Thus, $\mathbb{S}_3 \ge \mathcal{OPT}(P1)$. Then, $\mathbb{S}_1 \ge \mathbb{S}_3 / (\frac{U_{\max}}{U_{\min}} \cdot (1 + \frac{\rho_{\max}}{\mu_{\min}})) \ge \mathcal{OPT}(P1) / (\frac{U_{\max}}{U_{\min}} \cdot (1 + \frac{\rho_{\max}}{\mu_{\min}}))$.

The resource violation analysis is in Lemma 4. The time complexity of Algorithm 2 for examining each incoming request r is dominated by invoking Algorithm 1, which is $O(|\mathbb{V}_r| \cdot |V| \cdot |\mathbb{DT}_r| \cdot \log \frac{1}{\epsilon} + \frac{|\mathbb{V}_r| \cdot |V|}{\epsilon^4})$, due to Theorem 2. ■

VI. Performance Evaluation

In this section, we evaluated the performance of the proposed algorithms for the (dynamic) utility maximization problem in an MEC network. We also investigated the impacts of important parameters on the performance of the proposed algorithms.

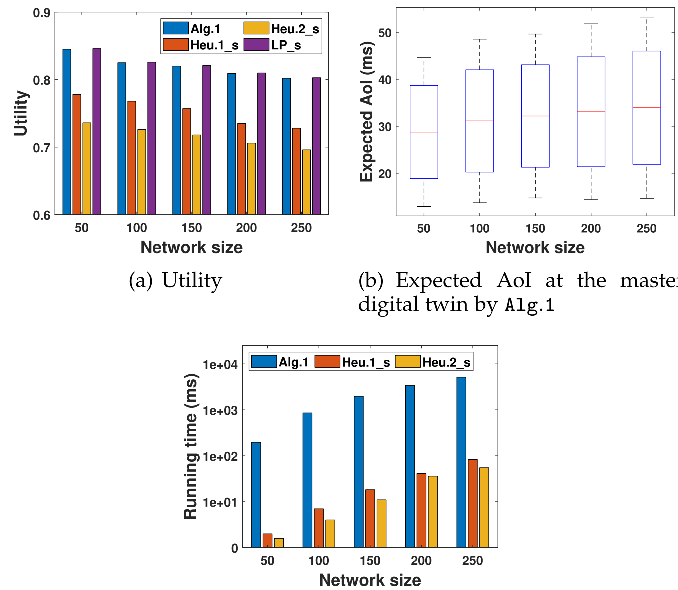
TABLE II
TABLE OF EXPERIMENTAL PARAMETERS

Parameter	Value	Parameter	Value
$\lvert V\rvert$	$[50, 250]$	$\lvert R\rvert$	500
C_v	$[4000, 8000]$ MHz	$\lvert\mathbb{DT}_r\rvert$	$[5, 15]$
$c_{r,n}$	$[50, 500]$ MHz	$t_{proc}^{master}(r, q, v_0)$	$[10, 20]$ ms
d_e	$[0.2, 1]$ ms	$s_{r,q}$	$[1, 10]$ MB
$\lvert\mathcal{N}\rvert$	200	D_r^{req}	$[25, 50]$ ms
t_n	$[20, 60]$ ms	$\xi_{r,n}$	$[50, 150]$ ms
$s_{r,n}^{raw}$	$[5, 25]$ MB	$\lambda_{r,n}$	0.1
$s_{r,n}^{worker}$	$[1, 5]$ MB	$w_{r,n}$	$\frac{1}{\lvert\mathbb{DT}_r\rvert}$

A. Parameter Settings

We considered an MEC network that consists of the number V of APs (cloudlets) from 50 to 250, which is generated by the GT-ITM tool [5]. The computing capacities C_v on cloudlets are within $[4000, 8000]$ MHz [31]. The amount $c_{r,n}$ of computing resource demanded by a worker or master digital twin is within $[50, 500]$ MHz [15], [20]. The transmission delay d_e for transmitting a unit of data (one MB) along a link $e \in E$ is drawn in $[0.2, 1]$ ms [39]. There are 200 objects. The number of possible moving locations of an object is no more than 10% of the number of APs in the network [24]. The update interval t_n of each object is within $[20, 60]$ ms [34], and the volume $s_{r,n}^{raw}$ of the update data generated by the object is within $[5, 25]$ MB [34]. The data processing rate by a worker digital twin in a cloudlet is within $[0.5, 2]$ MB per ms [38], and the size $s_{r,n}^{worker}$ of its processed data is within $[1, 5]$ MB. There are 50 DTN slices with the number $\lvert\mathbb{DT}_r\rvert$ of worker digital twins from 5 to 15. There are 500 incoming DTN slicing requests, with each requesting a random preset DTN slice. The query processing delay $t_{proc}^{master}(r, q, v_0)$ by a master digital twin in a cloudlet is within $[10, 20]$ ms, and the volume $s_{r,q}$ of a query result is within $[1, 10]$ MB [20]. Each user stays under the coverage of a random AP, and the service delay requirement D_r^{req} is within $[25, 50]$ ms [39]. For the utility function (5), we adopt $f_{r,n}(t_{AoI}(r, n, v, v_0)) = 1 - t_{AoI}(r, n, v, v_0)/\xi_{r,n} + \lambda_{r,n}$, where the AoI threshold $\xi_{r,n}$ is within $[40, 200]$ ms [34] and $\lambda_{r,n} = 0.1$. The coefficient $w_{r,n}$ in (6) is set as $\frac{1}{\lvert\mathbb{DT}_r\rvert}$. The value in each figure is the mean over 30 different MEC network topologies with the same size. The running time of algorithms is based on a desktop with a 3.60GHz Intel 8-Core i7 CPU and 16 GB RAM. Unless otherwise specified, these parameters are adopted by default. The experimental parameters are summarized in Table II.

We evaluated Algorithm 1, referred to as Algorithm 1, for the utility maximization problem against the following benchmarks: (1) `Heu.1_s`: it deploys the master digital twin of each request in a cloudlet with the least query service delay, and each worker digital twin is deployed in a cloudlet to minimize the expected AoI of its sent data at the master digital twin. (2) `Heu.2_s`: which is similar to `Heu.1_s`, each worker digital twin is deployed in a cloudlet to minimize the expected AoI of the collected data at the worker digital twin from the object. (3) `LP_s`: the Linear Program (LP) (23), which will serve as the upper bound on the optimal solution of the problem.



(a) Utility (b) Expected AoI at the master digital twin by `Alg.1`

(c) Running time

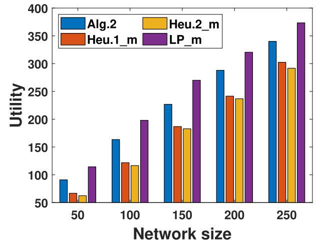
Fig. 2. Algorithm performance for the utility maximization problem for a single DTN slicing request.

We evaluated Algorithm 2, referred to as Algorithm 2, for the dynamic utility maximization problem, against benchmark algorithms: `Heu.1_m`, `Heu.2_m`, and `LP_m`. Especially, `Heu.1_m` and `Heu.2_m` try to admit each incoming DTN slicing request through invoking `Heu.1_s` and `Heu.2_s` for its admission, respectively.

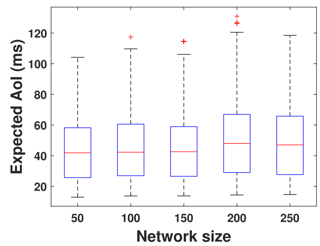
B. Algorithm Performance Evaluation

We first investigated the performance of Algorithm 1 against `Heu.1_s`, `Heu.2_s` and `LP_s` for the utility maximization problem for a single DTN slicing request, by varying the network size from 50 to 250. We observed from Fig. 2(a) that Algorithm 1 achieves near-optimal performance, compared with `LP_s`, and the utility gain by Algorithm 1 is 10.2% and 15.2% more than that by `Heu.1_s` and `Heu.2_s` respectively when the network size reaches 250. This is because Algorithm 1 jointly deploys the master digital twin near users to meet their delay requirements, whilst deploying worker digital twins to maximize the utility gain. Fig. 2(b) shows that the expected AoI of the master digital twin from different worker digital twins of an admitted request by Algorithm 1. Fig. 2(c) shows that Algorithm 1 takes the most running time because of invoking the algorithm in [2] repeatedly.

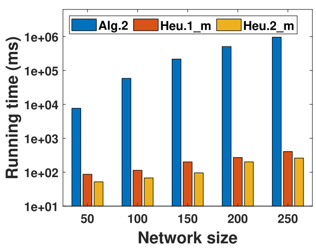
We then studied the performance of Algorithm 2 against `Heu.1_m` and `LP_m` for the dynamic utility maximization problem for multiple DTN slicing requests, by varying the network size from 50 to 250. Observed from Fig. 3(a), the accumulative utility gain by Algorithm 2 is 79.5% of that by `LP_m`, and Algorithm 2 outperforms `Heu.1_m` and `Heu.2_m` by 36.1% and 45.9% respectively when the network size is set at 50. Fig. 3(b) shows that the expected AoI among the master digital twins from their different worker digital twins of admitted requests by Algorithm 2. Fig. 3(c) illustrates that Algorithm 2 takes the most



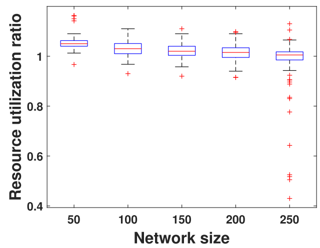
(a) Accumulative utility



(b) Expected AoI at master digital twins by `Alg.2`

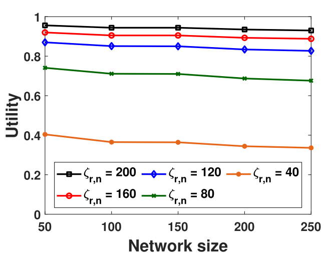


(c) Running time

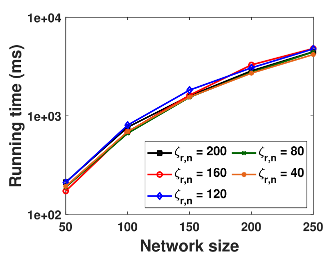


(d) Resource utilization ratios of cloudlets by `Alg.2`

Fig. 3. Algorithm performance for the dynamic utility maximization problem for multiple DTN slicing requests.



(a) Accumulative utility



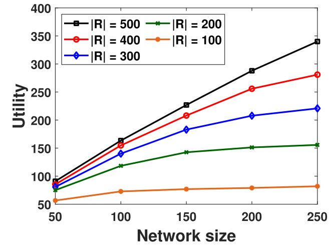
(b) Running time

Fig. 4. Impact of the AoI threshold $\lambda_{r,n}$ on the performance of Algorithm 1.

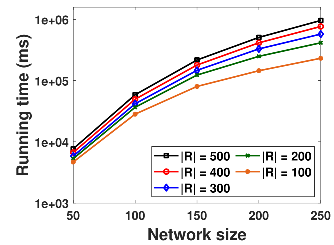
running time due to invoking Algorithm 1 to admit requests. Fig. 3(d) plots the resource utilization ratios on cloudlets, and the computing capacity on any cloudlet is violated by no more than 16.3% by Algorithm 2. Fig. 3 demonstrates that Algorithm 2 is promising, due to adopting a smart admission control policy for the admission of each incoming request.

C. Parameter Impacts on Algorithm Performance

We first evaluated the impact of the AoI threshold $\lambda_{r,n}$ on the performance of Algorithm 1. Fig. 4 shows the accumulative utility gain and running time of Algorithm 1 when the value range of $\lambda_{r,n}$ is drawn from 80 to 200. Fig. 4(a) demonstrates that the accumulative utility gain of Algorithm 1 with $\lambda_{r,n} = 80$ is 36.1% of that by itself with $\lambda_{r,n} = 200$ when the network size is set at 250. This is because a larger $\lambda_{r,n}$ means more tolerance for the staleness of data from objects, and more utility is obtained. Fig. 4(b) shows the value changes of $\lambda_{r,n}$ are insignificant on the running time of Algorithm 1.



(a) Accumulative utility



(b) Running time

Fig. 5. Impact of the number of requests on the performance of Algorithm 2.

We then evaluated the impact of the number $|R|$ of DTN slicing requests on the performance of Algorithm 2. Fig. 5 depicts the performance curves by varying the number of requests from 100 to 500. We observed that Algorithm 2 with $|R| = 100$ takes the least running time, and its accumulative utility gain is 24.1% of that by itself with $|R| = 500$ when the network size is set at 250. The rationale behind is the large number of requests leads to more utility gains.

VII. Conclusion

In this paper, we investigated AoI-aware service provisioning in MEC environments via DTN slicing. We measured the utility gain of a DTN slicing request based on the DTN model quality impacted by the expected AoI at its master digital twin. We formulated two optimization problems: the utility maximization problem for a single DTN slicing request; and the dynamic utility maximization problem for multiple DTN slicing requests. We developed a performance-guaranteed approximation algorithm for the former, and an online algorithm with a provable competitive ratio for the latter. We also evaluated the performance of the proposed algorithms via simulations. The simulation results showed that the proposed algorithms are promising and achieve near-optimal performance, outperforming their counterparts by at least 10.2%. Also, the algorithm performance can be impacted by a series of factors, such as the network size, the AoI threshold, and the number of requests.

Acknowledgment

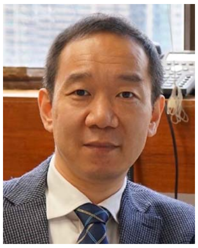
The authors appreciate the three anonymous referees and the Associate Editor for their constructive comments and invaluable suggestions, which help us to improve the quality and presentation of the paper greatly.

References

[1] I. F. Akyildiz and W. Wang, "The predictive user mobility profile framework for wireless multimedia networks," *IEEE/ACM Trans. Netw.*, vol. 12, no. 6, pp. 1021–1035, Dec. 2004.
[2] R. Cohen, L. Katzir, and D. Raz, "An efficient approximation for the generalized assignment problem," *Inf. Process. Lett.*, vol. 100, pp. 162–166, 2006.
[3] Y. Dai, K. Zhang, S. Maharjan, and Y. Zhang, "Deep reinforcement learning for stochastic computation offloading in digital twin networks," *IEEE Trans. Ind. Inform.*, vol. 17, no. 7, pp. 4968–4977, Jul. 2021.

[4] M. X. Goemans and D. P. Williamson, "The primal-dual method for approximation algorithms and its application to network design problems," in *Book Chapter of Approximation Algorithms for NP-Hard Problems*. Worcester, U.K.: PWS Publishing Company, 1997, pp. 144–191.
[5] GT-ITM, 2019. [Online]. Available: http://www.cc.gatech.edu/projects/gtitm/
[6] R. Han, J. Wang, L. Bai, J. Liu, and J. Choi, "Age of information and performance analysis for UAV-aided IoT systems," *IEEE Internet Things J.*, vol. 8, no. 19, pp. 14447–14457, Oct. 2021.
[7] A. Hossain and N. Ansari, "5G multi-band numerology-based TDD RAN slicing for throughput and latency sensitive services," *IEEE Trans. Mobile Comput.*, vol. 22, no. 3, pp. 1263–1274, Mar. 2023.
[8] Y. Hui et al., "Collaboration as a service: Digital-twin-enabled collaborative and distributed autonomous driving," *IEEE Internet Things J.*, vol. 9, no. 19, pp. 18607–18619, Oct. 2022.
[9] P. Jia, X. Wang, and X. Shen, "Accurate and efficient digital twin construction using concurrent end-to-end synchronization and multi-attribute data resampling," *IEEE Internet Things J.*, vol. 10, no. 6, pp. 4857–4870, Mar. 2023.
[10] S. Jošilo and G. Dán, "Joint wireless and edge computing resource management with dynamic network slice selection," *IEEE/ACM Trans. Netw.*, vol. 30, no. 4, pp. 1865–1878, Aug. 2022.
[11] B. Li, Y. Liu, L. Tan, H. Pan, and Y. Zhang, "Digital twin assisted task offloading for aerial edge computing and networks," *IEEE Trans. Veh. Technol*, vol. 71, no. 10, pp. 10863–10877, Oct. 2022.
[12] J. Li, S. Guo, W. Liang, Q. Chen, Z. Xu, and W. Xu, "SFC-enabled reliable service provisioning in mobile edge computing via digital twins," in *Proc. IEEE 19th Int. Conf. Mobile Ad Hoc Smart Syst.*, 2022, pp. 311–317.
[13] J. Li et al., "Digital twin-assisted, SFC-enabled service provisioning in mobile edge computing," *IEEE Trans. Mobile Comput.*, vol. 23, no. 1, pp. 393–408, Jan. 2024.
[14] J. Li et al., "Digital twin-enabled service provisioning in edge computing via continual learning," *IEEE Trans. Mobile Comput.*, vol. 23, no. 6, pp. 7335–7350, Jun. 2024.
[15] J. Li et al., "AoI-aware user service satisfaction enhancement in digital twin-empowered edge computing," *IEEE/ACM Trans. Netw.*, vol. 32, no. 2, pp. 1677–1690, Apr. 2024.
[16] J. Li et al., "Wait for fresh data? Digital twin empowered IoT services in edge computing," in *Proc. IEEE 20th Int. Conf. Mobile Ad Hoc Smart Syst.*, 2023, pp. 397–405.
[17] J. Li et al., "AoI-aware, digital twin-empowered IoT query services in mobile edge computing," *IEEE/ACM Trans. Netw.*, vol. 32, no. 4, pp. 3636–3650, Aug. 2024.
[18] J. Li et al., "Mobility-aware utility maximization in digital twin-enabled serverless edge computing," *IEEE Trans. Comput.*, vol. 73, no. 7, pp. 1837–1851, Jul. 2024.
[19] J. Li, W. Liang, J. Wang, and X. Jia, "Accumulative fidelity maximization of inference services in DT-assisted edge computing," in *Proc. 1st IEEE Int. Conf. Meta Comput.*, 2024.
[20] J. Li, J. Wang, Q. Chen, Y. Li, and A. Y. Zomaya, "Digital twin-enabled service satisfaction enhancement in edge computing," in *Proc. IEEE Conf. Comput. Commun.*, 2023, pp. 1–10.
[21] L. Li, L. Tang, Q. Liu, Y. Wang, X. He, and Q. Chen, "DTN-assisted dynamic cooperative slicing for delay-sensitive service in MEC-enabled IoT via deep deterministic policy gradient with variable action," *IEEE Internet Things J.*, vol. 10, no. 12, pp. 10708–10724, Jun. 2023.
[22] X. Liang, W. Liang, Z. Xu, Y. Zhang, and X. Jia, "Multiple service model refreshments in digital twin-empowered edge computing," *IEEE Trans. Services Comput.*, early access, Dec. 12, 2023, doi: 10.1109/TSC.2023.3341988.
[23] X. Lin, J. Wu, J. Li, W. Yang, and M. Guizani, "Stochastic digital-twin service demand with edge response: An incentive-based congestion control approach," *IEEE Trans. Mobile Comput.*, vol. 22, no. 4, pp. 2402–2416, Apr. 2023.
[24] Y. Ma, W. Liang, J. Li, X. Jia, and S. Guo, "Mobility-aware and delay-sensitive service provisioning in mobile edge-cloud networks," *IEEE Trans. Mobile Comput.*, vol. 21, no. 1, pp. 196–210, Jan. 2022.
[25] R. Prasad and A. Sunny, "Scheduling slice requests in 5G networks," *IEEE/ACM Trans. Netw.*, vol. 31, no. 6, pp. 3025–3036, Dec. 2023.
[26] Y. Ren, S. Guo, B. Cao, and X. Qiu, "End-to-end network SLA quality assurance for C-RAN: A closed-loop management method based on digital twin network," *IEEE Trans. Mobile Comput.*, vol. 23, no. 5, pp. 4405–4422, May 2024.
[27] H. Shen, Y. Tian, T. Wang, and G. Bai, "Slicing-based task offloading in space-air-ground integrated vehicular networks," *IEEE Trans. Mobile Comput.*, vol. 23, no. 5, pp. 4009–4024, May 2024.
[28] X. Shang, Y. Mao, Y. Liu, Y. Huang, Z. Liu, and Y. Yang, "Online container scheduling for data-intensive applications in serverless edge computing," in *Proc. IEEE Conf. Comput. Commun.*, 2023, pp. 1–10.
[29] Y. Shu, Z. Wang, H. Liao, Z. Zhou, N. Nasser, and M. Imran, "Age-of-information-aware digital twin assisted resource management for distributed energy scheduling," in *Proc. IEEE Glob. Commun. Conf.*, 2022, pp. 5705–5710.
[30] W. Sun, H. Zhang, R. Wang, and Y. Zhang, "Reducing offloading latency for digital twin edge networks in 6G," *IEEE Trans. Veh. Technol*, vol. 69, no. 10, pp. 12240–12251, Oct. 2020.
[31] L. Tang, Y. Du, Q. Liu, J. Li, S. Li, and Q. Chen, "Digital twin assisted resource allocation for network slicing in Industry 4.0 and beyond using distributed deep reinforcement learning," *IEEE Internet Things J.*, vol. 10, no. 19, pp. 16989–17006, Oct. 2023.
[32] F. Tao, B. Xiao, Q. Qi, J. Cheng, and P. Ji, "Digital twin modeling," *J. Manuf. Syst.*, vol. 64, pp. 372–389, 2022.
[33] V. Tripathi, R. Talak, and E. Modiano, "Age optimal information gathering and dissemination on graphs," *IEEE Trans. Mobile Comput.*, vol. 22, no. 1, pp. 54–68, Jan. 2023.
[34] M. Vaezi, K. Noroozi, T. D. Todd, D. Zhao, and G. Karakostas, "Digital twin placement for minimum application request delay with data age targets," *IEEE Internet Things J.*, vol. 10, no. 13, pp. 11547–11557, Jul. 2023.
[35] M. Vaezi et al., "Digital twins from a networking perspective," *IEEE Internet Things J.*, vol. 9, no. 23, pp. 23525–23544, Dec. 2022.
[36] H. Wang, Y. Wu, G. Min, and W. Miao, "A graph neural network-based digital twin for network slicing management," *IEEE Trans. Ind. Inform.*, vol. 18, no. 2, pp. 1367–1376, Feb. 2022.
[37] Y. Wu, K. Zhang, and Y. Zhang, "Digital twin networks: A survey," *IEEE Internet Things J.*, vol. 8, no. 18, pp. 13789–13804, Sep. 2021.
[38] Z. Xu, Y. Fu, Q. Xia, and H. Li, "Enabling age-aware big data analytics in serverless edge clouds," in *Proc. IEEE Conf. Comput. Commun.*, 2023, pp. 1–10.
[39] Z. Xu et al., "Stateful serverless application placement in MEC with function and state dependencies," *IEEE Trans. Comput.*, vol. 72, no. 9, pp. 2701–2716, Sep. 2023.
[40] R. Zhang, Z. Xie, D. Yu, W. Liang, and X. Cheng, "Digital twin-assisted federated learning service provisioning over mobile edge networks," *IEEE Trans. Comput.*, vol. 73, no. 2, pp. 586–598, Feb. 2024.
[41] Y. Zhang and W. Liang, "Cost-aware digital twin migration in mobile edge computing via deep reinforcement learning," in *Proc. 23rd IFIP/IEEE Netw. Conf.*, 2024, pp. 441–447.
[42] Y. Zhang, W. Liang, Z. Xu, and X. Jia, "Mobility-aware service provisioning in edge computing via digital twin replica placements," *IEEE Trans. Mobile Comput.*, early access, Apr. 24, 2024, doi: 10.1109/TMC.2024.3394839.
[43] Y. Zhang, W. Liang, W. Xu, Z. Xu, and X. Jia, "Cost minimization of digital twin placements in mobile edge computing," *ACM Trans. Sensor Netw.*, vol. 20, no. 3, pp. 1–26, 2024.
[44] L. Zhao, Z. Bi, A. Hawbani, K. Yu, Y. Zhang, and M. Guizani, "ELITE: An intelligent digital twin-based hierarchical routing scheme for softwarized vehicular networks," *IEEE Trans. Mobile Comput.*, vol. 22, no. 9, pp. 5231–5247, Sep. 2023.
[45] J. Zheng, A. Banchs, and G. de Veciana, "Constrained network slicing games: Achieving service guarantees and network efficiency," *IEEE/ACM Trans. Netw.*, vol. 31, no. 6, pp. 2698–2713, Dec. 2023.

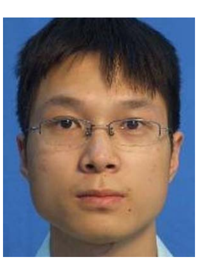
Jing Li received the BSc and PhD degrees with the first class Honours from The Australian National University in 2018 and 2022, respectively. He is currently a postdoctoral fellow with the City University of Hong Kong. His research interests include edge computing, Internet of Things, digital twin, network function virtualization, and combinatorial optimization.



Song Guo (Fellow, IEEE) is a full professor with the Department of Computer Science and Engineering, the Hong Kong University of Science and Technology. He also holds a Changjiang chair Professorship awarded by the Ministry of Education of China. He is a fellow of the Canadian Academy of Engineering. His research interests are mainly in edge AI, machine learning, mobile computing, and distributed systems. He published many papers in top venues with wide impact in these areas and was recognized as a Highly Cited researcher (Clarivate Web of Science). He is the recipient of more than a dozen Best Paper Awards from IEEE/ACM conferences, journals, and technical committees. He is the editor-in-chief of IEEE Open Journal of the Computer Society and the chair of IEEE Communications Society (ComSoc) Space and Satellite Communications Technical Committee. He was an IEEE ComSoc distinguished lecturer and a member of IEEE ComSoc Board of Governors. He has served for IEEE Computer Society on fellow Evaluation Committee, and been named on editorial board of a number of prestigious international journals like *IEEE Transactions on Parallel and Distributed Systems*, *IEEE Transactions on Cloud Computing*, *IEEE Transactions on Emerging Topics in Computing*, etc. He has also served as chairs of organizing and technical committees of many international conferences.

Weifa Liang (Senior Member, IEEE) received the BSc degree in computer science from Wuhan University, China in 1984, the ME degree in computer science from the University of Science and Technology of China in 1989, and the PhD degree in computer science from the Australian National University in 1998. He is a full professor with the Department of Computer Science, City University of Hong Kong. Prior to that, he was a full professor with the Research School of Computer Science, the Australian National University. His research interests include design and analysis of energy efficient routing protocols for wireless ad hoc and sensor networks, mobile edge computing (MEC), network function virtualization (NFV), Internet of Things and digital twins, design and analysis of parallel and distributed algorithms, approximation algorithms, and graph theory. He currently serves as an editor of *IEEE Transactions on Communications*.

Jianping Wang (Fellow, IEEE) received the BS and MS degrees in computer science from Nankai University, Tianjin, China, in 1996 and 1999, respectively, and the PhD degree in computer science from the University of Texas at Dallas, in 2003. She is currently a professor with the Department of Computer Science, City University of Hong Kong. Her research interests include cloud computing, service oriented networking, edge computing, and network performance analysis.



Quan Chen (Member, IEEE) received the BS, master's and PhD degrees from the School of Computer Science and Technology, Harbin Institute of Technology, China. He is currently an associate professor with the School of Computers, Guangdong University of Technology. He once worked as a postdoctoral research fellow with the Department of Computer Science at Georgia State University. His research interests include routing and scheduling algorithms in wireless networks and sensor networks.

Zicong Hong received the BEng degree in software engineering from the School of Data and Computer Science, Sun Yat-sen University. He is currently working toward the PhD degree with the Department of Computing in The Hong Kong Polytechnic University. His current research interest includes blockchain, edge/cloud computing, and federated learning.

Zichuan Xu (Member, IEEE) received the BSc and ME degrees in computer science from the Dalian University of Technology in China in 2008 and 2011, respectively, and the PhD degree in computer science from the Australian National University in 2016. From 2016 to 2017, he was a research associate with the Department of Electronic and Electrical Engineering, University College London, U.K. He is currently a full professor and PhD advisor with the School of Software, Dalian University of Technology. His research interests include mobile edge computing, serverless computing, network function virtualization, algorithmic game theory, and optimization problems.

Wenzheng Xu (Member, IEEE) received the BSc, ME, and PhD degrees in computer science from Sun Yat-Sen University, Guangzhou, China, in 2008, 2010, and 2015, respectively. He currently is an associate professor with Sichuan University, China. Also, he was a visitor with both the Australian National University, Australia and the Chinese University of Hong Kong, Hong Kong. His research interests include wireless ad hoc and sensor networks, mobile computing, approximation algorithms, combinatorial optimization, online social networks, and graph theory.

Bin Xiao (Fellow, IEEE) received the BSc and MSc degrees in electronics engineering from Fudan University, China, and the PhD degree in computer science from the University of Texas at Dallas, USA. He is a full professor with the Department of Computing, The Hong Kong Polytechnic University, Hong Kong. After his PhD graduation, he joined the Department of Computing of the Hong Kong Polytechnic University as an assistant professor. His research interests include AI and network security, data privacy, and blockchain systems. He published more than 180 technical papers in international top journals and conferences. Currently, he is the associate editor for *IEEE Internet of Things Journal*, *IEEE Transactions on Cloud Computing*, and *IEEE Transactions on Network Science and Engineering*. He is the vice chair of IEEE ComSoc CISTC committee. He has been the track co-chair of IEEE ICDCS 2022, symposium track co-chair of IEEE ICC 2020, ICC 2018 and Globecom 2017, and the general chair of IEEE SECON 2018. He is the member of ACM and CCF.