Digital Twin-Enabled Service Satisfaction Enhancement in Edge Computing

Jing Li[†], Jianping Wang[‡*], Quan Chen[§], Yuchen Li[¶], and Albert Y. Zomaya[%]
[†] The Hong Kong Polytechnic University, Hong Kong, P. R. China
[‡] City University of Hong Kong, Hong Kong, P. R. China
[§] Guangdong University of Technology, Guangzhou, 510006, P. R. China
[¶] The Australian National University, Canberra, ACT 2601, Australia
[%] The University of Sydney, Sydney, NSW 2006, Australia
Emails: jing5.li@polyu.edu.hk, jianwang@cityu.edu.hk, quan.c@gdut.edu.cn, u6013787@anu.edu.au, albert.zomaya@sydney.edu.au
[*] Corresponding author: Jianping Wang

IEEE INFOCOM 2023 - IEEE Conference on Computer Communications | 979-8-3503-3414-2/23/$31.00 ©2023 IEEE | DOI: 10.1109/INFOCOM53939.2023.10228893

***Abstract*—The emerging digital twin technique enhances the network management efficiency and provides comprehensive insights, through mapping physical objects to their digital twins. The user satisfaction on digital twin-enabled query services relies on the freshness of digital twin data, which is measured by the Age of Information (AoI). Because the remote cloud faces challenges in providing data for users due to long service delays, Mobile Edge Computing (MEC), as a promising technology, offers real-time data communication between physical objects and their digital twins at the edge of the core network. However, the mobility of physical objects and dynamic query arrivals make efficient service provisioning in MEC become challenging. In this paper, we investigate the dynamic digital twin placement for improving user service satisfaction in MEC environments. We focus on two user service satisfaction augmentation problems under both static and dynamic digital twin placement schemes: the static and dynamic utility maximization problems. We first formulate an Integer Linear Programming (ILP) solution to the static utility maximization problem when the problem size is small; otherwise, we propose a performance-guaranteed approximation algorithm for it. We then devise an online algorithm for the dynamic utility maximization problem with a provable competitive ratio. Finally, we evaluate the performance of the proposed algorithms through experimental simulations. Simulation results demonstrate that the proposed algorithms outperform the comparison baseline algorithms, and the performance improvement is no less than 11.6%, compared with the baseline algorithms.**

***Index Terms*—Mobile edge computing, digital twin, age of information, approximation and online algorithms.**

I. Introduction

Driven by the explosive growth of the Internet of Things (IoT) and its applications, unprecedented amounts of data generated by IoT devices are invaluable to businesses, governments and organizations, which are proliferating in the physical world [12]. The emerging digital twin technique has attracted more and more attention in digitizing the physical world through digital representation and analytics of big data [22]. Digital twins mirror the statuses of physical objects via continuous monitoring, along with implementing comprehensive and high-fidelity digital models for physical objects in the virtual world [8], [13]. By leveraging vivid simulations, the digital twin technique provides future insights and perceptual data for users to optimize their decision-making.

Digital twins that are deployed in traditional clouds usually demand real-time data and statuses of their physical objects for timely processing and analysis [11]. The long communication delay between physical devices and the remote cloud results in system performance degrading significantly [4]. The Mobile Edge Computing (MEC) paradigm has been envisioned as a revolutionary solution to provide computing resource in the proximity of users to curtail the communication delay [7], [9], [10]. Empowered by MEC, physical objects feed digital twins in cloudlets (edge servers) with raw real-time data, in the sequel, digital twins provide users with fresh digital twin data in MEC networks [12]. Superior to 5G networks without digital twins, the digital twins established in MEC networks pave the way to attain the marriage of physical and virtual worlds in 6G for immersive Metaverse, optimizing network service provisioning through real-time perception, analytics and prediction [19].

In this paper, we study the dynamic digital twin placement problem in an MEC network, considering the mobility of physical objects and users who request digital twin data within a finite time horizon. To provide fresh data to users, the metric, Age of Information (AoI), is commonly adopted to measure the freshness of data, i.e., the time elapsed from the data generation to its usage [2], [26]. Although digital twins of physical objects can be placed in the remote cloud with abundant computing and storage resource, the query result may not be as fresh as its users expected. Therefore, we consider digital twin placements in cloudlets near users to provide users with fresh data access, where the mobility of physical objects in MEC inevitably happens, and so do dynamic query requests for digital twin data [8], [11], [12], [19], [22].

Efficient service provisioning in MEC built upon digital twin data poses several challenges. Compared with the remote cloud, we can leverage the computing resource in cloudlets to deploy digital twins near physical objects and users, thereby providing query results with small AoI to users. How to quantitatively model user service satisfaction augmentation based on the AoI reduction of the received query results through the digital twins deployed in cloudlets, rather than the remote cloud? It is desirable that the network service provider can obtain the mobility information of physical objects and the user query request arrivals before the time horizon, by adopting the machine learning-based prediction mechanism [16]. With this preparation, the network service provider can deploy digital twins in cloudlets at the beginning of the time horizon to maximize the user service satisfaction augmentation. We refer to this as *the static digital twin placement scheme*. However, the mobility of physical objects and the user query request arrivals can be uncertain during the time horizon [14]. The digital twin placement without future knowledge is likely to result in unacceptable AoI of query results. Therefore, it is necessary to determine whether to replace digital twins at the beginning of each time slot, i.e., remove some existing digital twins and instantiate new digital twins in cloudlets. We refer to this as *the dynamic digital twin placement scheme*. Thus, it is challenging to deploy digital twins in cloudlets to maximize the user service satisfaction augmentation, subject to the computing capacities of cloudlets, under the static and dynamic digital twin placement schemes, respectively.

The novelty of this paper lies in introducing a new metric to measure user satisfactions on AoI-aware query service provisioning in MEC empowered by digital twins. We consider two novel utility maximization problems under static and dynamic digital twin placement schemes, respectively, based on this new metric. We aim to maximize the user service satisfaction augmentation by efficient digital twin placement in cloudlets. Performance-guaranteed approximation and online algorithms for problems are devised too.

The main contributions of this paper are given as follows. We explore the mobility of physical objects and dynamic query request arrivals during a finite time horizon in an MEC network, enabling efficient query service provisioning based on digital twin data. We first formulate two user service satisfaction augmentation problems in the MEC network: the static utility maximization problem and the dynamic utility maximization problem. We show the NP-hardness of both of the problems. We then formulate an Integer Linear Programming (ILP) solution to the static utility maximization problem when the problem size is small; otherwise, we devise an approximation algorithm with a guaranteed approximation ratio for the problem. We also devise an online algorithm with a provable competitive ratio for the dynamic utility maximization problem. We finally evaluate the performance of the proposed algorithms for augmenting user satisfaction on query service provisioning in MEC through experimental simulations. Experimental results demonstrate that the proposed algorithms are promising, and outperform the comparison baseline algorithms, improving by no less than 11.6% of the performance, compared to that of the baseline algorithms.

The rest of the paper is organized as follows. Section II summarizes the related works. Section III introduces problem definitions and shows the NP-hardness of defined problems. Section IV formulates an ILP solution, and devises an approximation algorithm for the static utility maximization problem. Section V proposes an online algorithm for the dynamic utility maximization problem. Section VI evaluates the proposed algorithms empirically, and Section VII concludes the paper.

II. Related work

The role of Mobile Edge Computing (MEC) has been studied in delay-sensitive service provisioning, and the mobility issue in MEC has sparked considerable discussions on providing seamless and real-time network services [4], [7], [14], [16], [18], [20]. Gao *et al.* [4] studied dynamic access network selections and service placements with user mobility in MEC. They developed an online algorithm to improve the service quality through balancing service delays. Considering high mobility and delay requirements of users, Ma *et al.* [14] proposed approximation and online algorithms for seamless network service provisioning in MEC. Polese *et al.* [18] exploited machine learning algorithms to predict the mobility patterns of users at network edges, optimizing the network service provisioning. Recently, the Age of Information (AoI), as a new metric to express the freshness of information, has attracted extensive attention in literature [2], [3], [6], [23], [24], [28]. Plenty of approaches have been proposed to optimize the AoI in service provisioning in MEC. Chen *et al.* [2] addressed a joint scheduling problem of data transmission and energy replenishment with directional chargers in MEC. They devised approximation algorithms to minimize the peak AoI. Xu *et al.* [24] focused on minimizing AoI for big data analytics with uncertain network delays, by proposing efficient algorithms. Zou *et al.* [28] explored the trade-off between the transmission and preprocessing time, and designed algorithms under different queue management schemes to minimize average AoI and average peak AoI. However, none of the above studies ever considered query service provisioning based on digital twin data in MEC.

The mobility-aware service provisioning in MEC empowered by the digital twin technique has also been investigated in existing literature [8], [11], [12], [19], [22]. Li *et al.* [11] established digital twins in a base station to estimate the states of MEC networks with user mobility, and devised a Deep Reinforcement Learning (DRL) algorithm to minimize total energy consumption of Unmanned Aerial Vehicles (UAVs). Lin *et al.* [12] paid much attention to dynamic and stochastic digital twin service demands of users with mobility in MEC. They devised an incentive-based congestion control scheme to maximize the long-term profit of the network service provider. Sun *et al.* [19] utilized digital twins of edge servers to predict their states under uncertain user mobility, along with developing a DRL algorithm to minimize the offloading delay of users. However, the mentioned studies did not consider the user service satisfaction augmentation based on the AoI query results by deploying digital twins in MEC networks.

Unlike the aforementioned studies, in this paper we investigate the mobility-aware query service provisioning based on digital twin data in MEC. We propose a novel metric to quantify the user service augmentation based on the AoI of query results by deploying digital twins in MEC networks. We focus on maximizing the accumulative user service augmentation in MEC under the static digital twin placement scheme and the dynamic digital twin placement scheme, respectively.

III. Preliminaries

A. System model

We consider a Mobile Edge Computing (MEC) network, modelled by an undirected graph $G = (V \cup \{v_0\}, E)$, where V is the set of Access Points (APs) and node v_0 is a remote cloud with abundant computing and storage resources. We assume that each AP is co-located with a cloudlet via an optical fiber cable, and the communication delay between an AP and its co-located cloudlet is negligible [14]. For notation simplicity, we abuse notation $v \in V$ to represent an AP or its co-located cloudlet. Denote by cap_v the available amount of computing resource of cloudlet $v \in V$. Each AP $v \in V$ is assumed to be connected to the remote cloud v_0 through a gateway, and its communication delay through the gateway is far larger than that between any pair of APs in the MEC network. Denote by d_{v,v_0} (resp. $d_{v_0,v}$) the transmission delay of transmitting a unit of data from AP v to the remote cloud (resp. from the remote cloud to AP v) through the gateway. E is the set of links connecting APs. Denote by d_e the transmission delay of transmitting a unit of data along link $e \in E$ [25]. We assume that the MEC network runs in a discrete-time fashion, i.e., the monitoring time horizon is slotted into equal *time slots*. Let $\mathbb{T} = \{0, 1, 2, \ldots, |\mathbb{T}| - 1\}$ be the set of time slots.

B. Physical objects with mobility and their digital twins

Let M be a set of physical objects for providing streaming raw data, and each object $m \in M$ is highly movable during the time horizon. We assume that the network service provider can deploy multiple digital twins for each object in the MEC network to provide digital twin data for user queries. Especially, a digital twin of each object $m \in M$ is established in the remote cloud v_0. Due to the high transmission delay between an AP and the remote cloud, the network service provider also determines the digital twin placement for objects in cloudlets to provide fresh data for user queries, subject to computing capacities in cloudlets. Denote by c_m the computing resource consumption of a digital twin of object $m \in M$ in a cloudlet.

We assume that each object $m \in M$ generates and sends its updated data to its digital twins for synchronization every δ_m time slots with $\delta_m \geq 1$ a positive integer, i.e., at time slot $k \cdot \delta_m$, with $k \geq 0$ a non-negative integer. Denote by a_m the data size of each update of object m, and the initial digital twin placement in the MEC network at time slot 0 is based on the updates of objects generated at time slot 0.

Due to the mobility of objects and limited computing resource in cloudlets, the network service provider can remove some existing digital twins and instantiate new digital twins in cloudlets during the time horizon to keep synchronizations between objects and their digital twins. We assume the delay of removing a digital twin is negligible [21], while instantiating a digital twin of object m leads to a delay of d_m^{ins}. However, frequent removals and instantiations of digital twins in cloudlets will lead to high overheads on service delays, dramatically downgrading the performance of query services.

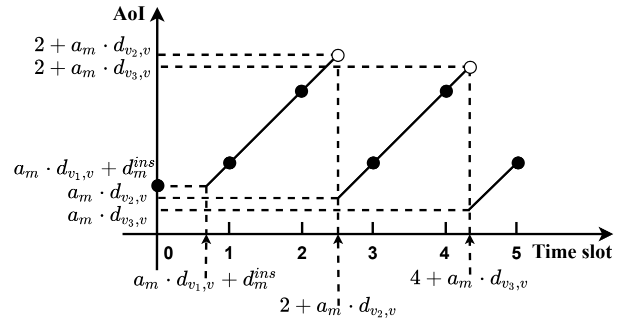


Fig. 1. An illustrative example of the AoI evolution at the digital twin of object m in node $v \in V \cup \{v_0\}$ with the digital twin deployed at time 0.

C. AoI at digital twins

Assuming the current time slot is t, the data at a digital twin is based on the update of its object generated at time slot t'. The *Age of Information* (AoI) at the digital twin is defined as $(t - t')$ [2]. In the following, we show the evolution of the AoI at a digital twin by distinguishing its instantiation time: the digital twin is instantiated at time slot $t = 0$ or $t \geq 1$.

We first assume that a digital twin of object $m \in M$ is deployed in cloudlet or the remote cloud $v \in V \cup \{v_0\}$ at time 0, by adopting the initial update of m. Suppose the location of object m at time slot 0 is v_1 (under the coverage of AP v_1), and the data at the digital twin of m in node v will be available at time $a_m \cdot d_{v_1,v} + d_m^{ins}$ with the AoI of $a_m \cdot d_{v_1,v} + d_m^{ins}$, where $a_m \cdot d_{v_1,v}$ is the delay of transmitting the update of object m along the shortest path between v_1 and v, and d_m^{ins} is the instantiation delay of digital twin in v for object m. We assume that the initial AoI at the digital twin of object m in v at time slot 0 is $a_m \cdot d_{v_1,v} + d_m^{ins}$ for queries issuing at time 0. Then the AoI at the digital twin increases after $a_m \cdot d_{v_1,v} + d_m^{ins}$ until receiving the next update of m at time $\delta_m + a_m \cdot d_{v_2,v}$, and its AoI decreases to $a_m \cdot d_{v_2,v}$, where object m sends its update every δ_m time slots, and v_2 is the location of m at time δ_m. The above procedure continues until the time horizon ends or the digital twin is removed.

For example, let the time horizon be $\mathbb{T} = \{0, 1, 2, 3, 4, 5\}$, and the object m sends its updated data with size a_m to its digital twins every 2 time slots, i.e, at time slots $0, 2, 4$, respectively. Object m is located at AP v_1 initially, then moves to AP v_2 and v_3 at time 2 and 4, respectively. We assume that a digital twin of object m is deployed in a node $v \in V \cup \{v_0\}$ at time slot 0 through adopting the first update of m at time 0, which will be not removed. The data at the digital twin of object m in cloudlet v will be available at time $a_m \cdot d_{v_1,v} + d_m^{ins}$ with the AoI of $a_m \cdot d_{v_1,v} + d_m^{ins}$, and we assume that the AoI of digital twin in node v at time slot 0 is $a_m \cdot d_{v_1,v} + d_m^{ins}$. The data at the digital twin gets stale until time $2 + a_m \cdot d_{v_2,v}$, and its AoI decreases to $a_m \cdot d_{v_2,v}$, as object m moves to AP v_2 at time 2 and sends its update. The AoI of data at the digital twin increases until time $4 + a_m \cdot d_{v_3,v}$, and decreases to $a_m \cdot d_{v_3,v}$,

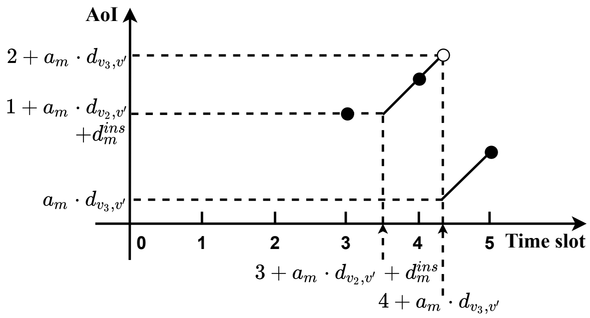


Fig. 2. An illustrative example of the AoI evolution at the digital twin of object m in cloudlet $v' \in V$ with the digital twin deployed at time 3.

with m moving to AP v_3 at time 4 and sending its update. The evolution of AoI of object m at the digital twin deployed in node $v \in V \cup \{v_0\}$ at time slot 0 is demonstrated in Fig. 1.

Moreover, suppose a digital twin of object m is instantiated in cloudlet $v' \in V$ at time $t \geq 1$ with $k \cdot \delta_m \leq t < (k+1) \cdot \delta_m$ and $k \geq 0$ a non-negative integer. Because the latest update of object m at time slot t is generated at time $k \cdot \delta_m$, the AoI of the latest update of m is $t - k \cdot \delta_m$. The data of the instantiated digital twin based on its latest update in v' will be available at time $t + a_m \cdot d_{v_2,v'} + d_m^{ins}$ with the AoI of $t - k \cdot \delta_m + a_m \cdot d_{v_2,v'} + d_m^{ins}$, where v_2 is the location of m at time t. We assume the AoI at this digital twin at time t is $t - k \cdot \delta_m + a_m \cdot d_{v_2,v'} + d_m^{ins}$, and its AoI evolves similarly.

For example, object m moves to v_2 at time slot 2, and a digital twin is deployed in cloudlet v' for object m at time slot 3. Assuming the latest update of object m at time 3 is generated at time 2, the data at the digital twin in v' will be available at time $3 + a_m \cdot d_{v_2,v'} + d_m^{ins}$ with the AoI of $1 + a_m \cdot d_{v_2,v'} + d_m^{ins}$. Suppose m moves to v_3 and sends its next update at time 4. Then the AoI at the digital twin increases after time $3 + a_m \cdot d_{v_2,v'} + d_m^{ins}$ until time $4 + a_m \cdot d_{v_3,v}$, and decreases to $a_m \cdot d_{v_3,v}$. Fig. 2 shows the AoI evolution of the digital twin of object m deployed in node v' at time slot 3.

D. AoI of a query result and the utility gain

There is a set Q of queries demanding the digital twin data of different objects at different time slots within a given time horizon. We assume that each query $q \in Q$ is represented by a tuple $\langle t_q, loc_q, m_q, s_q \rangle$, where t_q is its issuing time, loc_q is its issuing location (under the coverage of AP loc_q), m_q is the requested object, and s_q is the data size of its query result.

Denote by $\mathcal{V}_{m,t} \subseteq V \cup \{v_0\}$ the set of nodes, and the digital twins of object $m \in M$ are deployed in nodes $\mathcal{V}_{m,t}$ at time slot t. Let $w_{m,v,t}$ be the AoI at the digital twin of object m in node $v \in \mathcal{V}_{m,t}$ at time slot t. Each query $q \in Q$ will retrieve the data from the digital twin in node $v_q \in \mathcal{V}_{m,t}$, such that the AoI of the query result is minimized, i.e., $v_q = \arg\min_{v \in \mathcal{V}_{m_q,t_q}} \{w_{m_q,v,t_q} + s_q \cdot d_{v,loc_q}\}$, where d_{v,loc_q} is the delay of transmitting a unit data along the shortest path between v and loc_q. The AoI of the query result of query q thus is $W_{q,v_q} = w_{m_q,v_q,t_q} + s_q \cdot d_{v_q,loc_q}$.

Queries can always retrieve the data at digital twins in remote cloud v_0, which, however, usually leads to high AoI of query results. Therefore, digital twins are deployed in cloudlets V to provide fresher data for queries, thereby augmenting user service satisfaction. We define *the utility gain* u_q of such user service satisfaction augmentation for query $q \in Q$ as follows.

$$u_q = W_{q,v_0} - W_{q,v_q}, \tag{1}$$

where $W_{q,v_0} = w_{m_q,v_0,t_q} + s_q \cdot d_{v_0,v_q}$ is the AoI of the query result of query q when retrieving the data at digital twin in remote cloud v_0. Utility gain u_q implies the AoI reduction of query result through deploying digital twins in cloudlets V.

E. Problem definitions

We consider two digital twin placement schemes in cloudlets V with the aim to maximize the total utility gain of queries Q : *the static digital twin placement scheme* and *the dynamic digital twin placement scheme*, respectively.

Under the static digital twin placement scheme, we assume that all queries in Q and the mobility of all objects in M during the time horizon $\mathbb{T}$ can be obtained by historical traces or prediction mechanisms based on machine learning methods [16]. Without removing any existing digital twin or instantiating new digital twins during time horizon $\mathbb{T}$, we determine the digital twin placement in cloudlets at the beginning of time horizon $\mathbb{T}$ under the static digital twin placement scheme, subject to computing capacities in cloudlets, with the aim to maximize the total utility gain of queries.

Definition 1: Given an MEC network $G = (V \cup \{v_0\}, E)$, a time horizon $\mathbb{T}$, a set Q of queries, a set M of objects with given mobility during the time horizon, and a set V of cloudlets, the *static utility maximization problem* in G is to maximize the total utility gain of queries in Q, by deploying digital twins for each object in M in cloudlets under the static digital twin placement scheme, subject to the computing capacity on each cloudlet in G.

For the dynamic digital twin placement scheme, we assume that at the beginning of each time slot $t \in \mathbb{T}$, only a set $Q_t \subseteq Q$ of queries issued at time t are revealed, as well as the mobility of objects at time t. We then dynamically deploy or replace digital twins in cloudlets at the beginning of each time slot t, by removing existing digital twins or instantiating new digital twins, subject to computing capacities of cloudlets. Note that instantiating new digital twins will lead to instantiation delays, and we aim to maximize the accumulative utility gain of queries in Q under the dynamic digital twin placement scheme for the given time horizon.

Definition 2: Given an MEC network $G = (V \cup \{v_0\}, E)$, a time horizon $\mathbb{T}$, a set Q_t of queries coming at the beginning of each time slot $t \in \mathbb{T}$, a set M of objects are highly movable during the time horizon, and a set V of cloudlets, the *dynamic utility maximization problem* in G is to maximize the accumulative utility gain of queries in $Q = \cup_{t \in \mathbb{T}} Q_t$, by deploying digital twins for each object in M in cloudlets under the dynamic digital twin placement scheme, subject to the computing capacity on each cloudlet in G.

F. NP-hardness of the proposed problems

Theorem 1: The static utility maximization problem is NP-hard.

Proof We show the NP-hardness of the problem through a reduction from a well-known NP-hard problem - the knapsack problem [17]. Consider a special case of the static utility maximization problem, where there is a single AP v co-located with a cloudlet with computing capacity of cap_v. All objects are located at AP v without any mobility, while all queries are issued by AP v. The reduction is as follows. In a knapsack problem instance, there is a bin with capacity of cap_v and $|M|$ items. Each item $m \in M$ has a weight c_m, i.e., the computing resource consumption of a digital twin of object m. Let Q_m be the set of queries requesting the data of object m, and each item $m \in M$ has a profit $\sum_{q \in Q_m}(W_{q,v_0} - \min\{W_{q,v_0}, W_{q,v}\})$, i.e., the utility gain of deploying digital twin of m in cloudlet v. The knapsack problem is to maximize the total profit by packing as many items as possible into the bin, subject to the bin capacity. It can be seen that this knapsack problem is equivalent to the special case of the static utility maximization problem. The static utility maximization problem is NP-hard, because the knapsack problem is NP-hard [17]. ■

Theorem 2: The dynamic utility maximization problem is NP-hard.

The proof is omitted, due to space limitation.

IV. Approximation algorithm for the static utility maximization problem

In this section, we formulate an Integer Linear Programming (ILP) solution to the static utility maximization problem. We then devise an approximation algorithm for the problem.

A. ILP solution

Under the static digital twin placement scheme, the deployed digital twins will not be removed during the time horizon. For each digital twin of object m deployed in node $v \in V \cup \{v_0\}$, we can obtain its AoI $w_{m,v,t}$ at each time slot t. Recall that $W_{q,v} = w_{m_q,v,t_q} + s_q \cdot d_{v,v_q}$ is the AoI of query result when query q retrieves the data at the digital twin in v.

Let $x_{m,v}$ be a binary decision variable, where $x_{m,v} = 1$ indicates that a digital twin of object m is placed in cloudlet $v \in V$; and $x_{m,v} = 0$ otherwise. Let $y_{q,v}$ be a binary decision variable, where $y_{q,v} = 1$ indicates that query q retrieves the data at digital twin of object m_q in node $v \in V \cup \{v_0\}$ with the minimum AoI of query result; and $y_{q,v} = 0$ otherwise.

The ILP solution to the problem is given as follows.

$$\text{Maximize } \sum_{q \in Q} \sum_{v \in V \cup \{v_0\}} y_{q,v} \cdot (W_{q,v_0} - W_{q,v}), \quad (2)$$

subject to:

$$\sum_{m \in M} c_m \cdot x_{m,v} \leq cap_v, \quad \forall v \in V \quad (3)$$

$$\sum_{v \in V \cup \{v_0\}} y_{q,v} = 1, \quad \forall q \in Q \quad (4)$$

$$y_{q,v} \leq x_{m_q,v}, \quad \forall q \in Q,\ \forall v \in V \quad (5)$$

$$x_{m,v} \in \{0,1\}, \quad \forall m \in M,\ \forall v \in V \quad (6)$$

$$y_{q,v} \in \{0,1\}, \quad \forall q \in Q,\ \forall v \in V \cup \{v_0\}, \quad (7)$$

where objective (2) is the total utility gain of queries by utility gain definition (1). Constraint (3) is the computing capacity constraint on each cloudlet. Constraint (4) indicates each query q selects one digital twin of object m_q in node $v \in V \cup \{v_0\}$ to minimize the AoI of query result. Constraint (5) shows query q can retrieve the data at digital twin of object m_q in cloudlet v only if the digital twin of m_q is deployed in cloudlet v.

B. The submodular function and the utility function

Denote by $\mathcal{A}$ the set of deployed digital twins for objects in M in cloudlets V, i.e., if a digital twin of object m is deployed in cloudlet v, such a digital twin, denoted by $n_{m,v}$, is added into set $\mathcal{A}$. Let $\mathbb{V}_m(\mathcal{A}) \subseteq V \cup \{v_0\}$ be the set of nodes with digital twins placed for object m by $\mathcal{A}$, with $v_0 \in \mathbb{V}_m(\mathcal{A})$. Recall that query $q \in Q$ requests the digital twin data of object m_q. By utility gain definition (1), we define *the utility function* $f(\mathcal{A})$ of deploying digital twins $\mathcal{A}$ into cloudlets V as follows.

$$f(\mathcal{A}) = \sum_{q \in Q} W_{q,v_0} - \sum_{q \in Q} \min_{v \in \mathbb{V}_{m_q}(\mathcal{A})} \{W_{q,v}\}, \quad (8)$$

Definition 3: Given a finite set Ω and a non-negative real number set $\mathbb{R}^{\geq 0}$, *a submodular function* on Ω is a set function $f : 2^{\Omega} \longmapsto \mathbb{R}^{\geq 0}$. Function $f(\cdot)$ is a submodular function if (i) $f(\emptyset) = 0$, and (ii) for every two sets $\mathbb{A}, \mathbb{B} \subseteq \Omega$ with $\mathbb{A} \subseteq \mathbb{B}$ and every $s \in \Omega \setminus \mathbb{B}$, $f(\mathbb{A} \cup \{s\}) - f(\mathbb{A}) \geq f(\mathbb{B} \cup \{s\}) - f(\mathbb{B})$.

Lemma 1: $f(\cdot)$ in Eq (8) is a submodular function.

Proof We can see $\mathbb{V}_{m_q}(\emptyset) = \{v_0\}$, $\forall q \in Q$, then $f(\emptyset) = 0$. Denote by $\mathbb{A}$ and $\mathbb{B}$ two sets of digital twins placed in cloudlets V with $\mathbb{A} \subseteq \mathbb{B}$, then $\mathbb{V}_m(\mathbb{A}) \subseteq \mathbb{V}_m(\mathbb{B}), \forall m \in M$, i.e., more digital twins of each object are deployed by $\mathbb{B}$ than that by $\mathbb{A}$.

Let $n_{m',v'} \notin \mathbb{B}$ be a digital twin of object m' in cloudlet v'. Let $\mathbb{A}' = \mathbb{A} \cup \{n_{m',v'}\}$ and $\mathbb{B}' = \mathbb{B} \cup \{n_{m',v'}\}$, then $\mathbb{A}' \subseteq \mathbb{B}'$. Let $Q_m \subseteq Q$ be the query set for m with $Q = \cup_{m \in M} Q_m$. Let $Q_{m'}^{v'}(\mathbb{A}') \subseteq Q_{m'}$ be the set of queries with $Q_{m'} = \cup_{v \in \mathbb{V}_{m'}(\mathbb{A}')} Q_m^v(\mathbb{A}')$, and each $q \in Q_{m'}^{v'}(\mathbb{A}')$ retrieves the data at the digital twin of m' in v' with the minimum AoI of its query result by $\mathbb{A}'$. For each $q \in Q_{m'}^{v'}(\mathbb{A}')$, we have $\min_{v \in \mathbb{V}_{m'}(\mathbb{A}')}\{W_{q,v}\} = W_{q,v'}$ and $\min_{v \in \mathbb{V}_{m'}(\mathbb{A})}\{W_{q,v}\} \geq W_{q,v'}$. Similarly, we have $\min_{v \in \mathbb{V}_{m'}(\mathbb{B}')}\{W_{q,v}\} = W_{q,v'}, \forall q \in Q_{m'}^{v'}(\mathbb{B}')$. Since $\mathbb{A}' \subseteq \mathbb{B}'$, we have $Q_{m'}^{v'}(\mathbb{B}') \subseteq Q_{m'}^{v'}(\mathbb{A}')$. Then

$$\begin{aligned} f(\mathbb{A}') - f(\mathbb{A}) &= \sum_{q \in Q} \min_{v \in \mathbb{V}_{m_q}(\mathbb{A})}\{W_{q,v}\} - \sum_{q \in Q} \min_{v \in \mathbb{V}_{m_q}(\mathbb{A}')}\{W_{q,v}\} \\ &= \sum_{q \in Q_{m'}} \big(\min_{v \in \mathbb{V}_{m'}(\mathbb{A})}\{W_{q,v}\} - \min_{v \in \mathbb{V}_{m'}(\mathbb{A}')}\{W_{q,v}\}\big) \\ &= \sum_{q \in Q_{m'}^{v'}(\mathbb{A}')} \big(\min_{v \in \mathbb{V}_{m'}(\mathbb{A})}\{W_{q,v}\} - W_{q,v'}\big), \end{aligned} \quad (9)$$

because $\min_{v \in \mathbb{V}_{m'}(\mathbb{A}')}\{W_{q,v}\} = W_{q,v'}$, $\forall q \in Q_{m'}^{v'}(\mathbb{A}')$, and $\min_{v \in \mathbb{V}_{m'}(\mathbb{A})}\{W_{q,v}\} = \min_{v \in \mathbb{V}_{m'}(\mathbb{A}')}\{W_{q,v}\}$, $\forall q \in Q_{m'} \setminus Q_{m'}^{v'}(\mathbb{A}')$, i.e., $n_{m',v'}$ has no impact on queries in $Q_{m'} \setminus Q_{m'}^{v'}(\mathbb{A}')$. Similarly, we have

$$f(\mathbb{B}') - f(\mathbb{B}) = \sum_{q \in Q_{m'}^{v'}(\mathbb{B}')} \big(\min_{v \in \mathbb{V}_{m'}(\mathbb{B})}\{W_{q,v}\} - W_{q,v'}\big). \quad (10)$$

By Eq. (9), $\min_{v \in \mathbb{V}_{m'}(\mathbb{A})}\{W_{q,v}\} \geq W_{q,v'}, \forall q \in Q_{m'}^{v'}(\mathbb{A}')$, and $Q_{m'}^{v'}(\mathbb{B}') \subseteq Q_{m'}^{v'}(\mathbb{A}')$, we have

$$f(\mathbb{A}') - f(\mathbb{A}) \geq \sum_{q \in Q_{m'}^{v'}(\mathbb{B}')} \big(\min_{v \in \mathbb{V}_{m'}(\mathbb{A})}\{W_{q,v}\} - W_{q,v'}\big). \quad (11)$$

We thus have $(f(\mathbb{A}') - f(\mathbb{A})) - (f(\mathbb{B}') - f(\mathbb{B})) \geq \sum_{q\in Q^{v'}_{m'}(\mathbb{B}')}(\min_{v\in\mathbb{V}_{m'}(\mathbb{A})}\{W_{q,v}\} - \min_{v\in\mathbb{V}_{m'}(\mathbb{B})}\{W_{q,v}\})$.

Since $\mathbb{A} \subseteq \mathbb{B}$, we have $\min_{v\in\mathbb{V}_{m'}(\mathbb{A})}\{W_{q,v}\} \geq \min_{v\in\mathbb{V}_{m'}(\mathbb{B})}\{W_{q,v}\}$ and $f(\mathbb{A}') - f(\mathbb{A}) \geq f(\mathbb{B}') - f(\mathbb{B})$. ■

C. Approximation algorithm

Recall $n_{m,v}$ is a digital twin of object $m \in M$ deployed in cloudlet $v \in V$. Let $N = \{n_{m,v} \mid m \in M, v \in V\}$ be the set of potential digital twin deployments in cloudlets. Let $n^l \in N$ be the lth deployed digital twin in cloudlets. Denote by $\mathcal{A}^{l-1}$ the set of the first $l-1$ digital twins deployed prior to the deployment of the lth digital twin, with $n^l \notin \mathcal{A}^{l-1}$, and denote by $\mathcal{A}^l = \mathcal{A}^{l-1} \cup \{n^l\}$. Assuming digital twins in $\mathcal{A}^{l-1}$ have been deployed in cloudlets, we define *the marginal utility gain* of deploying digital twin n^l in a cloudlet as follows.

$$\Delta f(n^l \mid \mathcal{A}^{l-1}) = f(\mathcal{A}^{l-1} \cup \{n^l\}) - f(\mathcal{A}^{l-1}). \tag{12}$$

Let $c(n^l)$ be the computing resource consumption of digital twin n^l. To guide the deployment of digital twins, we define the ratio $\rho(n^l)$ of the marginal utility gain of deploying n^l to its computing resource consumption as follows.

$$\rho(n^l) = \Delta f(n^l \mid \mathcal{A}^{l-1})/c(n^l). \tag{13}$$

Let $\mathcal{C}_v(\mathcal{A}^{l-1})$ be the accumulative computing resource consumption of cloudlet v, by deploying digital twins in $\mathcal{A}^{l-1}$.

The approximation algorithm proceeds greedily, and $\mathcal{A}^0 = \emptyset$ initially. To deploy the lth digital twin n^l, we identify a digital twin $n_{m',v'} \in N \backslash \mathcal{A}^{l-1}$ with the largest $\rho(n_{m',v'})$, and the computing capacity of cloudlet v' is not fully used prior to its deployment, i.e., $\mathcal{C}_{v'}(\mathcal{A}^{l-1}) < C_{v'}$, where $C_{v'}$ is the computing capacity of cloudlet v'. The chosen $n^l (= n_{m',v'})$ is added to set $\mathcal{A}^{l-1}$ to form $\mathcal{A}^l = \mathcal{A}^{l-1} \cup \{n^l\}$. Note that the computing capacity of cloudlet v' may be violated after deploying n^l, i.e., $\mathcal{C}_{v'}(\mathcal{A}^l) = \mathcal{C}_{v'}(\mathcal{A}^{l-1}) + c(n^l) > C_{v'}$. We construct two sets S_1 and S_2 of digital twins to avoid resource violation, with $S_1 = \emptyset$ and $S_2 = \emptyset$ initially. If deploying n^l results in no resource violation, n^l is added to set S_1; otherwise, n^l is added to set S_2. This procedure continues until no more digital twins can be deployed in cloudlets. Suppose the solution obtained is $\mathcal{A}$, then S_1 and S_2 are two disjoint sets with $\mathcal{A} = S_1 \cup S_2$. We claim that deploying digital twins in either S_1 or S_2 in cloudlets will not result in any computing resource violation, and this claim will be shown rigorously in Theorem 3. The proposed approximation algorithm chooses one of S_1 and S_2 with the larger utility gain as the problem solution. The detailed algorithm is given in `Algorithm` 1.

D. Algorithm analysis

Given solution $\mathcal{A} = S_1 \cup S_2$ delivered by `Algorithm` 1, denote by $\mathcal{A}_v$ the set of digital twins deployed in cloudlet v by $\mathcal{A}$, i.e., $\mathcal{A} = \cup_{v\in V}\mathcal{A}_v$. Let $\mathcal{A}^{opt}$ be the set of deployed digital twins in the optimal solution to the static utility maximization problem. Similarly, denote by $\mathcal{A}^{opt}_v$ the set of digital twins deployed in cloudlet v by $\mathcal{A}^{opt}$ with $\mathcal{A}^{opt} = \cup_{v\in V}\mathcal{A}^{opt}_v$.

Lemma 2: $\rho(n^l) \geq \frac{\Delta f(n^* \mid \mathcal{A})}{c(n^*)}, \forall v \in V, \forall n^l \in \mathcal{A}_v, \forall n^* \in \mathcal{A}^{opt}_v \setminus \mathcal{A}_v$.

Algorithm 1 Approximation algorithm for the static utility maximization problem

---

Input: An MEC network $G = (V \cup \{v_0\}, E)$, a time horizon $\mathbb{T}$, a set Q of queries, a set M of objects with given mobility during the time horizon, and a set V of cloudlets.

Output: Maximize the total utility gains of all queries under the static digital twin placement scheme.

```
Identify the shortest path between each pair of nodes in V ∪ {v_0};
A^0 ← ∅;  S_1 ← ∅;  S_2 ← ∅;  C_v(A^0) ← 0, ∀v ∈ V;
N ← {n_{m,v} | m ∈ M, v ∈ V};  l ← 1;
while N\A^{l-1} ≠ ∅ and it exists a cloudlet v' with C_{v'}(A^{l-1}) < C_{v'} do
    Identity a digital twin n_{m',v'} ∈ N\A^{l-1} as n^l with the largest ρ(n^l)
    in Eq. (13) and C_{v'}(A^{l-1}) < C_{v'};  n^l ← n_{m',v'};
    A^l ← A^{l-1} ∪ {n^l};  C_{v'}(A^l) ← C_{v'}(A^{l-1}) + c(n^l);
    if C_{v'}(A^l) > C_{v'} then
        S_2 ← S_2 ∪ {n^l};
    else
        S_1 ← S_1 ∪ {n^l};
    end if;
    l ← l + 1;
end while;
if f(S_1) ≥ f(S_2) then
    return S_1 and f(S_1);
else
    return S_2 and f(S_2);
end if;
```

Proof If $\mathcal{A}^{opt}_v \backslash \mathcal{A}_v = \emptyset$, then $\Delta f(n^* \mid \mathcal{A}) = 0$, and the lemma follows. Otherwise, $\forall v \in V, \forall n^l \in \mathcal{A}_v, \forall n^* \in \mathcal{A}^{opt}_v \setminus \mathcal{A}_v$,

$$\rho(n^l) = \frac{\Delta f(n^l \mid \mathcal{A}^{l-1})}{c(n^l)} \geq \frac{\Delta f(n^* \mid \mathcal{A}^{l-1})}{c(n^*)} \tag{14}$$

$$\geq \frac{\Delta f(n^* \mid \mathcal{A})}{c(n^*)}, \tag{15}$$

where Ineq. (14) holds because we identify $n_{m,v} \in N \backslash \mathcal{A}^{l-1}$ as n^l with the largest $\rho(n^l)$ and $\mathcal{C}_v(\mathcal{A}^{l-1}) < C_v$ prior to deploying n^l. Also, $n^* \in \mathcal{A}^{opt}_v \setminus \mathcal{A}_v$, then $n^* \in N \backslash \mathcal{A}^{l-1}$. Ineq.(15) holds because $f(\cdot)$ is submodular by Lemma 1. ■

Lemma 3: $f(\mathcal{A}) \geq \sum_{n^*\in\mathcal{A}^{opt}\setminus\mathcal{A}} \Delta f(n^* \mid \mathcal{A})$.

Proof Let $n^*_{v,max} = \arg\max_{n^*\in\mathcal{A}^{opt}_v\setminus\mathcal{A}_v} \frac{\Delta f(n^* \mid \mathcal{A})}{c(n^*)}, \ \forall v \in V$.

$$f(\mathcal{A}) = \sum_{l=1}^{|\mathcal{A}|} \Delta f(n^l \mid \mathcal{A}^{l-1}) = \sum_{v\in V}\sum_{n^l\in\mathcal{A}_v} \Delta f(n^l \mid \mathcal{A}^{l-1})$$

$$= \sum_{v\in V}\sum_{n^l\in\mathcal{A}_v} \rho(n^l)\cdot c(n^l) \geq \sum_{v\in V} \frac{\Delta f(n^*_{v,max} \mid \mathcal{A})}{c(n^*_{v,max})} \sum_{n^l\in\mathcal{A}_v} c(n^l) \tag{16}$$

$$\geq \sum\nolimits_{v\in V} \frac{\Delta f(n^*_{v,max} \mid \mathcal{A})}{c(n^*_{v,max})} \cdot \sum\nolimits_{n^*\in\mathcal{A}^{opt}_v\setminus\mathcal{A}_v} c(n^*) \tag{17}$$

$$\geq \sum\nolimits_{v\in V} \sum\nolimits_{n^*\in\mathcal{A}^{opt}_v\setminus\mathcal{A}_v} \frac{\Delta f(n^* \mid \mathcal{A})}{c(n^*)} \cdot c(n^*) \tag{18}$$

$$= \sum_{v\in V} \sum_{n^*\in\mathcal{A}^{opt}_v\setminus\mathcal{A}_v} \Delta f(n^* \mid \mathcal{A}) = \sum_{n^*\in\mathcal{A}^{opt}\setminus\mathcal{A}} \Delta f(n^* \mid \mathcal{A}), \tag{19}$$

where Ineq. (16) holds by Lemma 2 and definition of $n^*_{v,max}$. Ineq. (17) holds because the consumed computing resource of each cloudlet v by $\mathcal{A}$ is no less than its computing capacity, while no cloudlet has its computing capacity violated in the optimal solution. Ineq. (18) holds by definition of $n^*_{v,max}$. ■

Theorem 3: Given an MEC network $G = (V \cup \{v_0\}, E)$, a set Q of queries, a set M of objects with given mobility during time horizon $\mathbb{T}$, and cloudlets V, there is a $\frac{1}{4}$-approximation algorithm, `Algorithm` 1, for the static utility maximization problem, which takes $O(|M|^2 \cdot |V|^2 \cdot |Q| + |V|^3)$ time.

Proof Recall that $\mathcal{A}^{opt}$ is the set of deployed digital twins in the optimal solution, and $f(\mathcal{A}^{opt})$ is its total utility gain. With $\mathcal{A} = S_1 \cup S_2$ the solution delivered by `Algorithm` 1, then

$$f(\mathcal{A}^{opt}) \le f(\mathcal{A} \cup \mathcal{A}^{opt}) \le f(\mathcal{A}) + \sum_{n^* \in \mathcal{A}^{opt} \setminus \mathcal{A}} \Delta f(n^* \mid \mathcal{A}) \quad (20)$$

$$\le 2 \cdot f(\mathcal{A}). \quad (21)$$

Ineq. (20) holds because $f(\cdot)$ is a submodular function by Lemma 1, and Ineq. (21) holds by Lemma 3. Because $\mathcal{A}$ is partitioned into two disjoint sets S_1 and S_2, i.e., $S_1 \cup S_2 = \mathcal{A}$,

$$f(S_1) + f(S_2) \ge f(\mathcal{A}) \ge \frac{f(\mathcal{A}^{opt})}{2}, \quad \text{by Ineq. (21)} \quad (22)$$

We then choose a set with the larger utility gain between S_1 and S_2 as the final solution to the problem, and

$$\max\{f(S_1), f(S_2)\} \ge \frac{f(\mathcal{A}^{opt})}{4}. \quad (23)$$

Now we show that the deployment of digital twins in either S_1 or S_2 onto the MEC network does not result in any computing resource violations. A digital twin is added to S_1 only if its deployment causes no resource violation. Meanwhile, a digital twin is added to S_2 if its deployment violates the computing capacity of a cloudlet, and the cloudlet is excluded for further digital twin placement. Therefore, each cloudlet is assigned with at most one digital twin in S_2.

The analysis of the time complexity of `Algorithm` 1 is omitted due to space limitation. ■

V. Online algorithm for the dynamic utility maximization problem

In this section, we devise an online algorithm with a provable competitive ratio for the problem.

A. Online algorithm

Under the dynamic digital twin placement scheme, the mobility of objects in M at time slot t and the set $Q_t \subseteq Q$ of queries issued at time slot t are revealed at the beginning of each time slot t. Intuitively, we can replace digital twins in cloudlets at each time slot t to maximize the utility gain of Q_t without any future knowledge. However, the accumulative instantiation delays of queries retrieving the newly instantiated digital twins can be prohibitively large during the time horizon, which will dramatically downgrade user service satisfactions.

Inspired by the work in [27], we introduce a digital twin replacement control policy to bound the accumulative instantiation delays. For each time slot $t \ge 1$, we define the *dynamic AoI* $\mathcal{W}_t^D$ of queries in Q_t as the accumulative instantiation delays of queries in Q_t, i.e., let $Q_t^D \subseteq Q_t$ be the set of queries retrieving the data at newly instantiated digital twins at time slot t, we have $\mathcal{W}_t^D = \sum_{q \in Q_t^D} d_{m_q}^{ins}$, where $d_{m_q}^{ins}$ is the instantiation delay of a digital twin of object m_q for query q. We let $\mathcal{W}_0^D = 0$ because cloudlets are empty

Algorithm 2 Online algorithm for the dynamic utility maximization problem

Input: An MEC network $G = (V \cup \{v_0\}, E)$, a time horizon $\mathbb{T}$, a set Q_t of queries at each time slot t, a set M of objects with high mobility, and there is no future information.

Output: Maximize the accumulative utilities of all queries under the dynamic digital twin placement scheme.

Obtain and deploy the set $\mathbb{S}_0$ of digital twins in cloudlets V at time slot 0 by invoking `Algorithm` 1 for queries Q_0;
$t \leftarrow 1$; $\hat{t} \leftarrow 0$;
while $t \le |\mathbb{T}| - 1$ **do**
    Obtain the set $\mathbb{S}_t$ of digital twins in cloudlets V at time slot t by invoking `Algorithm` 1 for queries Q_t;
    Calculate the incurred dynamic AoI $\mathcal{W}_t^D$ by $\mathbb{S}_t$ at time slot t.
    if $\mathcal{W}_t^D \le \frac{1}{\beta} \cdot \sum_{t'=\hat{t}}^{t-1} (\sum_{q \in Q_{t'}} W_{q,v_0} - \mathcal{W}_{t'}^S)$ **then**
        Apply set $\mathbb{S}_t$ for digital twin replacement; $\hat{t} \leftarrow t$;
    end if;
    $t \leftarrow t + 1$;
end while;

initially, and there is no need to consider the dynamic AoI at time slot 0. Recall that W_{q,v_q} is the minimum AoI of query result of query q. We also define the *static AoI* $\mathcal{W}_t^S$ of queries in Q_t with $\mathcal{W}_t^S = \sum_{q \in Q_t} W_{q,v_q} - \mathcal{W}_t^D$, i.e., the accumulative AoI of queries in Q_t without considering the dynamic AoI.

By the utility gain definition (1), the accumulative utility gain at each time slot t is $\sum_{q \in Q_t} W_{q,v_0} - \mathcal{W}_t^D - \mathcal{W}_t^S$.

The proposed online algorithm is an iterative algorithm with each iteration in one time slot. At time slot 0, all cloudlets in V are empty initially, and we first adopt `Algorithm` 1 to obtain set $\mathbb{S}_0$ of digital twins for deployment in cloudlets for queries Q_0, with given Q_0 and locations of objects at time slot 0. At each time slot $t \ge 1$, we can calculate the AoI at each potential digital twin $n_{m,v}$, based on whether $n_{m,v}$ has been deployed in node v at time $t-1$ or not, as shown in Section III-C. Then we can also invoke `Algorithm` 1 to obtain set $\mathbb{S}_t$ of digital twins for queries Q_t at each time slot $t \ge 1$. However, we cannot apply $\mathbb{S}_t$ for digital twin replacement at each time slot $t \ge 1$ directly, and we design a digital twin replacement control policy to avoid large instantiation delays as follows. Let $\hat{t}$ be the time slot when the last digital twin replacement occurred, with $\hat{t} = 0$ initially. Let $\beta > 1$ be a control parameter to bound the accumulative instantiation delays. If the dynamic AoI of the query results in current time slot t is no greater than $\frac{1}{\beta}$ times the accumulative utility gain of queries from time slot $\hat{t}$ to time slot $(t-1)$ without considering the dynamic AoI, i.e., $\mathcal{W}_t^D \le \frac{1}{\beta} \cdot \sum_{t'=\hat{t}}^{t-1} (\sum_{q \in Q_{t'}} W_{q,v_0} - \mathcal{W}_{t'}^S)$, we apply $\mathbb{S}_t$ for digital twin replacement at time slot t; otherwise, we keep the digital twin placement in last time slot $t-1$. It is noted that a smaller value of β implies more tolerance on instantiation delays, therefore, more frequent digital twin replacements are needed. Otherwise, a larger β indicates tolerating less instantiation delays, thereby performing less frequent digital twin replacements. The detailed online algorithm for the dynamic utility maximization problem is given in `Algorithm` 2.

B. Algorithm analysis

Lemma 4: The total dynamic AoI of query results is no larger than $\frac{1}{\beta}$ times the total utility gain of queries in

Q over time horizon $\mathbb{T}$ by Algorithm 2, without considering the dynamic AoI of queries, i.e., $\sum_{t=0}^{|\mathbb{T}|-1} \mathcal{W}_t^D \leq \frac{1}{\beta} \cdot \sum_{t=0}^{|\mathbb{T}|-1}(\sum_{q \in Q_t} W_{q,v_0} - \mathcal{W}_t^S)$.

Proof Let $\hat{t}_i$ be the time slot of the ith digital twin replacement, with $\hat{t}_0 = 0$ and $\mathcal{W}_{\hat{t}_0}^D = 0$. Let $\hat{t}_L \leq |\mathbb{T}| - 1$ be the time slot that the last digital twin replacement occurs during $\mathbb{T}$. By the digital twin replacement control policy, we have

$$\sum_{t=0}^{|\mathbb{T}|-1} \mathcal{W}_t^D = \sum_i \mathcal{W}_{\hat{t}_i}^D \leq \frac{1}{\beta} \cdot \sum_{t=0}^{\hat{t}_L - 1}(\sum_{q \in Q_t} W_{q,v_0} - \mathcal{W}_t^S) \leq \frac{1}{\beta} \cdot \sum_{t=0}^{|\mathbb{T}|-1}(\sum_{q \in Q_t} W_{q,v_0} - \mathcal{W}_t^S), \quad (24)$$

because $\sum_{q \in Q_t} W_{q,v_0} - \mathcal{W}_t^S \geq 0$, $\forall t \in \mathbb{T}$. ■

Lemma 5: The optimal solution to the dynamic utility maximization problem is no greater than λ times the total static AoI of query results in solution by Algorithm 2 , i.e., $U^* \leq \lambda(\sum_{t=0}^{|\mathbb{T}|-1} \sum_{q \in Q_t} W_{q,v_0} - \sum_{t=0}^{|\mathbb{T}|-1} \mathcal{W}_t^S)$, where U^* is the utility gain of the optimal solution to the problem and $\lambda = \max_{t \in \mathbb{T}}\{\frac{\max\{\sum_{q \in Q_t} W_{q,v_0} - \mathcal{W}_t^S\}}{\min\{\sum_{q \in Q_t} W_{q,v_0} - \mathcal{W}_t^S\}}\}$, i.e., the maximum ratio of the largest to the smallest utility gain without considering the dynamic AoI of query results at any time slot [27].

Proof By the definition of λ, we have $\sum_{q \in Q_t} W_{q,v_0} - \mathcal{W}_t^{*S} \leq \lambda \cdot (\sum_{q \in Q_t} W_{q,v_0} - \mathcal{W}_t^S)$, where $\mathcal{W}_t^{*S}$ is the static AoI in the optimal solution at time slot t with $0 \leq t \leq |\mathbb{T}| - 1$. Then,

$$\sum_{t=0}^{|\mathbb{T}|-1} \sum_{q \in Q_t} W_{q,v_0} - \sum_{t=0}^{|\mathbb{T}|-1} \mathcal{W}_t^{*S} \leq \lambda(\sum_{t=0}^{|\mathbb{T}|-1} \sum_{q \in Q_t} W_{q,v_0} - \sum_{t=0}^{|\mathbb{T}|-1} \mathcal{W}_t^S). \quad (25)$$

$$U^* = \sum_{t=0}^{|\mathbb{T}|-1} \sum_{q \in Q_t} W_{q,v_0} - \sum_{t=0}^{|\mathbb{T}|-1} \mathcal{W}_t^{*S} - \sum_{t=0}^{|\mathbb{T}|-1} \mathcal{W}_t^{*D} \leq \sum_{t=0}^{|\mathbb{T}|-1} \sum_{q \in Q_t} W_{q,v_0} - \sum_{t=0}^{|\mathbb{T}|-1} \mathcal{W}_t^{*S} \leq \lambda(\sum_{t=0}^{|\mathbb{T}|-1} \sum_{q \in Q_t} W_{q,v_0} - \sum_{t=0}^{|\mathbb{T}|-1} \mathcal{W}_t^S), \text{ by Ineq.(25).} \quad (26)$$

Hence, the lemma follows. ■

Theorem 4: Given an MEC network $G = (V \cup \{v_0\}, E)$, a finite time horizon $\mathbb{T}$, a set Q_t of queries coming at the beginning of each time slot t with $Q = \cup_{t \in \mathbb{T}} Q_t$, a set M of highly mobile objects, and a set V of cloudlets, there is an online algorithm, Algorithm 2, with a competitive ratio of $\frac{1}{\lambda}(1 - \frac{1}{\beta})$ for the dynamic utility maximization problem, which takes $O(|M|^2 \cdot |V|^2 \cdot |Q| \cdot |\mathbb{T}| + |V|^3)$ time over $\mathbb{T}$, where $\beta > 1$ is a control parameter and $\lambda > 1$ is defined in Lemma 5.

Proof Denote by U^* and U the accumulative utility gain by the optimal solution and the solution delivered by Algorithm 2 to the problem, respectively. We have

$$U = \sum_{t=0}^{|\mathbb{T}|-1} \sum_{q \in Q_t} W_{q,v_0} - \sum_{t=0}^{|\mathbb{T}|-1} \mathcal{W}_t^S - \sum_{t=0}^{|\mathbb{T}|-1} \mathcal{W}_t^D \quad (27)$$
$$\geq \sum_{t=0}^{|\mathbb{T}|-1} \sum_{q \in Q_t} W_{q,v_0} - \sum_{t=0}^{|\mathbb{T}|-1} \mathcal{W}_t^S - \frac{1}{\beta} \cdot \sum_{t=0}^{|\mathbb{T}|-1}(\sum_{q \in Q_t} W_{q,v_0} - \mathcal{W}_t^S), \text{ by Lemma 4}$$
$$\geq (1 - \frac{1}{\beta})(\sum_{t=0}^{|\mathbb{T}|-1} \sum_{q \in Q_t} W_{q,v_0} - \sum_{t=0}^{|\mathbb{T}|-1} \mathcal{W}_t^S) \geq \frac{1}{\lambda}(1 - \frac{1}{\beta})U^*, \text{ by Lemma 5.} \quad (28)$$

The analysis of the time complexity of Algorithm 2 is omitted due to space limitation. ■

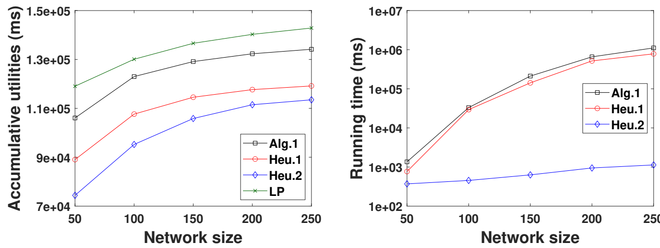
VI. Performance Evaluation

A. Experimental environment setting

We consider an MEC network, consisting of from 50 to 250 APs, and each AP is co-located with a cloudlet. We generate the topology of each MEC network by the GT-ITM tool [5]. The finite time horizon consists of 20 time slots [4], and the length of a time slot is set as 50 ms. There are 200 objects, and we simulate the mobility of objects during the finite time horizon by the BonnMotion tool [1]. The computing capacity on each cloudlet is drawn from $4,000$ MHz to $8,000$ MHz [24]. We further assume that the amount of computing resource demanded by a digital twin ranges from 200 MHz to $2,000$ MHz. There are 500 queries issued at the beginning of each time slot, with each query requesting the digital twin data of a random object. We assume that each object generates and sends its updated data to its digital twins every 1 or 2 time slots. The data size of each update of an object is set within $[2, 5]$ MB, while the size of a query result is set within $[0.5, 2]$ MB. The instantiation delay of a digital twin ranges from 20 ms to 40 ms [15]. The transmission delay for transmitting a unit of data (one MB) along a link between each pair of APs is set within $[0.2, 1]$ ms [25]. The transmission delay for transmitting a unit of data (one MB) between the remote cloud and an AP through the gateway is set within $[2, 10]$ ms [20]. The parameter β is set as 4. The value of each figure is the average of the results over 30 different topologies of MEC networks of the same size. The actual running time of each algorithm is obtained on a desktop with a 3.60GHz Intel 8-Core i7 CPU and 16GB RAM. Unless otherwise specified, these parameters will be adopted by default.

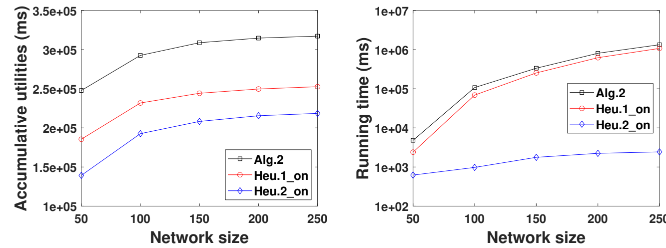
We evaluated the proposed algorithm Algorithm 1, referred to as `Alg.1`, for the static utility maximization problem, against the following benchmarks: (1) `Heu.1`: we iteratively identify a digital twin deployment with the largest utility improvement in a cloudlet with sufficient residual computing resource. This procedure ends until no cloudlet can accommodate more digital twins; (2) `Heu.2`: we consider the cloudlets one by one randomly. For the first cloudlet, we identify the digital twin deployment in this cloudlet with the largest utility improvement until the cloudlet can accommodate no more digital twins. This procedure ends until all cloudlets are examined; (3) `LP`: the Linear Programming solution (2) to the problem, as an upper bound on its optimal solution.

We evaluated algorithm Algorithm 2, referred to as `Alg.2`, for the dynamic utility maximization problem, against benchmark algorithms: `Heu.1_on` and `Heu.2_on`, which invoke algorithms `Heu.1` and `Heu.2` for digital twin replacement at each time slot, respectively.



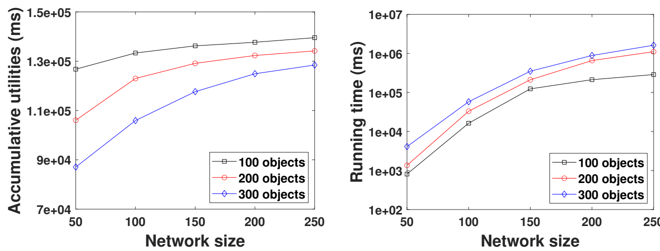
(a) The accumulative utilities (b) The running time

Fig. 3. Algorithm performance for the static utility maximization problem.



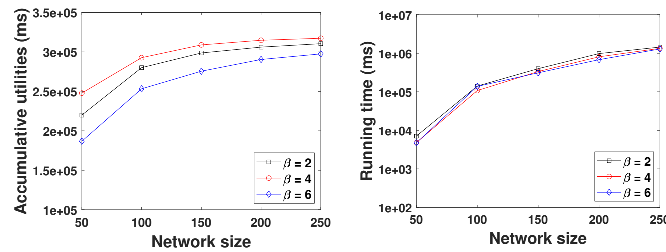
(a) The accumulative utilities (b) The running time

Fig. 5. Algorithm performance for the dynamic utility maximization problem.



(a) The accumulative utilities (b) The running time

Fig. 4. Impact of the number of objects on the performance of `Alg.1`.



(a) The accumulative utilities (b) The running time

Fig. 6. Impact of the parameter β on the performance of `Alg.2`.

B. Algorithm performance for the static utility maximization problem

We first studied the performance of algorithm `Alg.1` for the static utility maximization problem against benchmark algorithms `Heu.1`, `Heu.2` and `LP`, by varying the network size from 50 to 250. Fig. 3 illustrates the accumulative utilities and running time of different algorithms. It can be seen from Fig. 3(a) that when the network size reaches 250, algorithm `Alg.1` outperforms `Heu.1` and `Heu.2` by 11.6% and 17.2%, respectively, while the performance of algorithm `Alg.1` is 93.1% of that of algorithm `LP`. The reason is that algorithm `Alg.1` provides fresh query results to users by jointly considering the utility gain of digital twin deployment and the resulted computing resource consumption. From Fig. 3(b), we can see that algorithm `Heu.2` takes the least running time because of examining cloudlets one by one for digital twin deployment.

We then investigated the impact of the number of objects on the performance of algorithm `Alg.1`. Fig. 4 shows the performance curves of algorithm `Alg.1` with 100, 200 and 300 objects, respectively. From Fig. 4(a), the accumulative utilities by algorithm `Alg.1` with 100 objects is 45.6% higher than that by itself with 300 objects when the network size is 50. The justification is that a smaller number of objects leads to a higher probability that a deployed digital twin can provide the requested data for a query. Fig. 4(b) shows that `Alg.1` with 300 objects takes the most running time because `Alg.1` examines each potential digital twin deployment at each iteration until no more digital twin can be deployed in cloudlets.

C. Algorithm performance for the dynamic utility maximization problem

We evaluated the performance of algorithm `Alg.2` for the dynamic utility maximization problem against benchmark algorithms `Heu.1_on` and `Heu.2_on`, by varying the network size from 50 to 250. Fig. 5 plots the accumulative utilities and running time of different algorithms. Fig. 5(a) demonstrates that algorithm `Alg.2` outperforms algorithms `Heu.1_on` and `Heu.2_on` by 25.6% and 45.2% with the network size of 250. The rationale behind is that `Alg.2` establishes an efficient digital twin replacement control policy to guarantee the freshness of query results without any future knowledge.

We finally studied the impact of the parameter β on the performance of `Alg.2`. Fig. 6 shows the performance curves of `Alg.2` with $\beta = 2$, 4 and 6, respectively. We can see from Fig. 6(a) that when the network size is 50, the performance of `Alg.2` with $\beta = 4$ is 12.7% and 32.6% higher than that by itself with $\beta = 2$ and $\beta = 6$, respectively. This is because a larger β implies it intends to tolerate less instantiation delays and avoid digital twin replacement. Fig. 6(b) demonstrates that the value of β has little impact on the running time of `Alg.2`.

VII. Conclusion

In this paper, we studied two novel user service satisfaction enhancement problems in a digital twin-enabled MEC environment. Specifically, for the static utility maximization problem, we proposed an ILP solution and an approximation algorithm with a provable approximation ratio. For the dynamic utility problem, we devised an online algorithm with a provable competitive ratio. We also evaluated the performance of the proposed algorithms by simulations. Simulation results demonstrated that the proposed algorithms outperform the comparison baseline algorithms, improving the performance by no less than 11.6% compared to baseline algorithms.

Acknowledgement

The work is supported by a grant from Hong Kong Research Grant Council under NSFC/RGC N_CityU 140/20.

REFERENCES

[1] N. Aschenbruck, R. Ernst, E. Gerhards-Padilla, and M. Schwamborn. BonnMotion: a mobility scenario generation and analysis tool. *Proceedings of the 3rd International ICST Conference on Simulation Tools and Techniques*, pp. 1 – 10, 2010.

[2] Q. Chen, Z. Cai, L. Cheng, F. Wang, and H. Gao. Joint near-optimal age-based data transmission and energy replenishment scheduling at wireless-powered network edge. *Proc. of INFOCOM'22*, IEEE, pp. 770 – 779, 2022.

[3] Q. Chen, S. Guo, W. Xu, Z. Cai, L. Cheng, and H. Gao, AoI minimization charging at wireless-powered network edge. *Proc. of ICDCS'22*, IEEE, pp. 713 – 723, 2022.

[4] B. Gao, Z. Zhou, F. Liu, and F. Xu. Winning at the starting line: joint network selection and service placement for mobile edge computing. *Proc. of INFOCOM'19*, IEEE, pp. 1459 – 1467, 2019.

[5] GT-ITM. http://www.cc.gatech.edu/projects/gtitm/, 2019.

[6] X. He, S. Wang, X. Wang, S. Xu, and J. Ren. Age-based scheduling for monitoring and control applications in mobile edge computing systems. *Proc. of INFOCOM'22*, IEEE, pp. 1009 – 1018, 2022.

[7] Z. Hu, J. Niu, T. Ren, B. Dai, Q. Li, M. Xu, and S. K. Das. An efficient online computation offloading approach for large-scale mobile edge computing via deep reinforcement learning. *IEEE Transactions on Services Computing*, vol. 15, no. 2, pp. 669 – 683, 2022.

[8] L. Lei, G. Shen, L. Zhang, and Z. Li. Toward intelligent cooperation of UAV swarms: when machine learning meets digital twin. *IEEE Network*, vol. 35, no. 1, pp. 386 – 392, 2021.

[9] 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. *Proc of MASS'22*, IEEE, pp. 311 – 317, 2022.

[10] J. Li, S. Guo, W. Liang, Q. Chen, Z. Xu, W. Xu, and A. Y. Zomaya. Digital twin-assisted, SFC-enabled service provisioning in edge computing. To appear in *IEEE Transactions on Mobile Computing*, 2022, doi: 10.1109/TMC.2022.3227248.

[11] B. Li, Y. Liu, L. Tan, H. Pan, and Y. Zhang. Digital twin assisted task offloading for aerial edge computing and networks. *IEEE Transactions on Vehicular Technology*, vol. 71, no. 10, pp. 10863 – 10877, 2022.

[12] 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. To appear in *IEEE Transactions on Mobile Computing*, 2021, doi: 10.1109/TMC.2021.3122013.

[13] T. Liu, L. Tang, W. Wang, Q. Chen, and X. Zeng. Digital-twin-assisted task offloading based on edge collaboration in the digital twin edge network. *IEEE Internet of Things Journal*, vol. 9, no. 2, pp. 1427 – 1444, 2022.

[14] Y. Ma, W. Liang, J. Li, X. Jia, and S. Guo. Mobility-aware and delay-sensitive service provisioning in mobile edge-cloud networks. *IEEE Transactions on Mobile Computing*, vol. 21, no, 1, pp. 196 – 210, 2022.

[15] J. Martins, M. Ahmed, C. Raiciu, V. Olteanu, M. Honda, R. Bifulco, and F. Huici. ClickOS and the art of network function virtualization. *Proc. of USENIX NSDI'14*, 2014.

[16] E. F. Maleki, L. Mashayekhy, and S. M. Nabavinejad. Mobility-aware computation offloading in edge computing using machine learning. To appear in *IEEE Transactions on Mobile Computing*, 2021, doi: 10.1109/TMC.2021.3085527.

[17] D. Pisinger. Algorithms for knapsack problems. *Citeseer*, 1995.

[18] M. Polese, R. Jana, V. Kounev, K. Zhang, S. Deb, and M. Zorzi. Machine learning at the edge: a data-driven architecture with applications to 5G cellular networks. *IEEE Transactions on Mobile Computing*, vol. 20, no. 12, pp. 3367 – 3382, 2021.

[19] W. Sun, H. Zhang, R. Wang, and Y. Zhang. Reducing offloading latency for digital twin edge networks in 6G. *IEEE Transactions on Vehicular Technology*, vol. 69, no. 10, pp. 12240 – 12251, 2020.

[20] S. Wang, Y. Guo, N. Zhang, P. Yang, A. Zhou, and X. Shen. Delay-aware microservice coordination in mobile edge computing: a reinforcement learning approach. *IEEE Transactions on Mobile Computing*, vol. 20, no. 3, pp. 939 – 951, 2021.

[21] L. Wang, L. Jiao, J. Li, J. Gedeon and M. Mühlhäuser. MOERA: mobility-agnostic online resource allocation for edge computing. *IEEE Transactions on Mobile Computing*, vol. 18, no. 8, pp. 1843 – 1856, 2019.

[22] Z. Wang, R. Gupta, K. Han, H. Wang, A. Ganlath, N. Ammar, and P. Tiwari. Mobility digital twin: concept, architecture, case study, and future challenges. *IEEE Internet of Things Journal*, vol. 9, no. 18, pp. 17452 – 17467, 2022.

[23] C. Xu, Q. Xu, J. Wang, K. Wu, K. Lu, and C. Qiao. AoI-centric task scheduling for autonomous driving systems. *Proc. of INFOCOM'22*, IEEE, pp. 1019 – 1028, 2022.

[24] Z. Xu, W. Ren, W. Liang, W. Xu, Q. Xia, P. Zhou, and M. Li. Schedule or wait: age-minimization for IoT big data processing in MEC via online learning *Proc. of INFOCOM'22*, IEEE, pp. 1809 – 1818, 2022.

[25] Z. Xu, L. Zhou, H. Dai, W. Liang, W. Zhou, P. Zhou, W. Xu, and G. Wu. Energy-aware collaborative service caching in a 5G-enabled MEC with uncertain payoffs. *IEEE Transactions on Communications*, vol. 70, no. 2, pp. 1058 – 1071, 2022.

[26] R. D. Yates, Y. Sun, D. R. Brown, S. K. Kaul, E. Modiano, and S. Ulukus. Age of information: an introduction and survey. *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 5, pp. 1183 – 1210, 2021.

[27] L. Zhang, C. Wu, Z. Li, C. Guo, M. Chen, and F. Lau. Moving big data to the cloud: an online cost-minimizing approach. *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 12, pp. 2710 – 2721, 2013.

[28] P. Zou, O. Ozel, and S. Subramaniam. Optimizing information freshness through computation–transmission tradeoff and queue management in edge computing. *IEEE/ACM Transactions on Networking*, vol. 29, no. 2, pp. 949 – 963, 2021.