AoI-Aware User Service Satisfaction Enhancement in Digital Twin-Empowered Edge Computing

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

***Abstract*— The emerging digital twin technique enhances the network management efficiency and provides comprehensive insights on network performance, through mapping physical objects to their digital twins. The user satisfaction on digital twin-enabled service relies on the freshness of digital twin data, which is measured by the Age of Information (AoI). Due to long service delays, the use of the remote cloud for delay-sensitive service provisioning faces serious challenges. Mobile Edge Computing (MEC), as an ideal paradigm for delay-sensitive services, is able to realize real-time data communication between physical objects and their digital twins at the network edge. However, the mobility of physical objects and dynamics of user query arrivals make seamless service provisioning in MEC become challenging. In this paper, we investigate dynamic digital twin placements for improving user service satisfaction in MEC environments, by introducing a novel metric to measure user service satisfaction based on the AoI concept and formulating two user service satisfaction enhancement problems: the static and dynamic utility maximization problems under static and dynamic digital twin placement schemes. To this end, 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. We then propose an online algorithm with a provable competitive ratio for the dynamic utility maximization problem, by considering dynamic user query services. Finally, we evaluate the performance of the proposed algorithms via simulations. Simulation results demonstrate that the proposed algorithms outperform the comparison baseline algorithms, improving the algorithm performance by at least 10.7%, compared to the baseline algorithms.**

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

Manuscript received 23 March 2023; revised 15 August 2023; accepted 9 October 2023; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor T. Lan. Date of publication 20 October 2023; date of current version 18 April 2024. The work of Jing Li and Song Guo was supported in part by the Key-Area Research and Development Program of Guangdong Province under Grant 2021B0101400003; in part by the Hong Kong Research Grants Council (RGC) Research Impact Fund under Grant R5060-19 and Grant R5034-18; in part by the Areas of Excellence Scheme under Grant AoE/E-601/22-R; in part by the General Research Fund under Grant 152203/20E, Grant 152244/21E, Grant 152169/22E, and Grant 152228/23E; and in part by the Shenzhen Science and Technology Innovation Commission under Grant JCYJ20200109142008673. The work of Weifa Liang was supported by the City University of Hong Kong under Grant 9380137 and Grant 7005845. The work of Jianping Wang was supported by a grant from the Hong Kong Research Grant Council under Grant NSFC/RGC N_CityU 140/20. The work of Quan Chen was supported in part by NSFC under Grant 62372118 and in part by the Guangdong Basic and Applied Basic Research Foundation under Grant 2022A1515011032. The work of Zichuan Xu was supported in part by the National Natural Science Foundation of China under Grant 61802048; and in part by the Xinghai Scholar Program in the Dalian University of Technology, China. The work of Wenzheng Xu was supported by NSFC under Grant 62272328. *(Corresponding author: Weifa Liang.)*

Jing Li is with the Department of Computing, The Hong Kong Polytechnic University, Hong Kong, China (e-mail: jing5.li@polyu.edu.hk).

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

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

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

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/TNET.2023.3324704

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 [21]. The emerging digital twin technique has attracted more and more attention in digitizing the physical world through digital representation and analytics of big data [35]. Digital twins mirror the states of physical objects via continuous monitoring, along with implementing comprehensive and high-fidelity digital models for physical objects in the virtual world [10], [22]. By leveraging vivid simulations, the digital twin technique provides future insights and perceptual data for users in decision-making.

Digital twins deployed in remote clouds usually demand real-time data and states of their physical objects for timely processing and analysis [11]. However, the long communication delay between physical devices and the remote cloud results in system performance degradation significantly [5]. Mobile Edge Computing (MEC) as a promising computing paradigm has been envisioned as a revolutionary solution to offer computing resource in the proximity of users to curtail communication delays [8], [12], [13], [17], [18]. Empowered by MEC, physical objects are enabled to feed their digital twins in cloudlets (edge servers) with raw real-time data, in the sequel, digital twins can provide services to users with fresh digital twin data in MEC networks [21]. Superior to 5G and beyond 5G networks without digital twins, digital twin-enabled 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 [28].

In this paper, we study static and dynamic digital twin placement problems for delay-sensitive service provisioning in an MEC network, considering the mobility of both physical objects and users querying digital twin data for a finite time horizon. The quality of a query result depends on the freshness of the retrieved digital twin data. Each digital twin needs to continuously synchronize with its physical object in real-time to keep its data freshness. To measure the freshness of digital twin data, the Age of Information (AoI) metric is commonly adopted, i.e., the time elapsed from the data generated to its usage [3], [9], [40]. Although digital twins of physical objects can be placed in the remote cloud with abundant computing and storage resources, the query results on them may not be as fresh as users expected. Therefore, in this paper we consider digital twin placements in cloudlets to provide service to users with fresh digital twin data, where the mobility of physical objects in MEC inevitably happens, and so do mobile users [10], [11], [21], [28], [35].

Efficient service provisioning in a digital twin-assisted MEC network with highly mobile objects and users poses several challenges. Compared with the remote cloud, we can leverage the computing resource in cloudlets to deploy digital twins to the locations near their physical objects and users, thereby providing smaller-AoI query results. How to quantitatively model user service satisfaction enhancement based on the AoI reduction of their 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 users prior to the given time horizon, and such information can be obtained by adopting the machine learning-based prediction mechanism [25]. With this preparation, the network service provider can pre-deploy digital twins in cloudlets in the beginning of the time horizon to maximize user service satisfaction augmentation. We refer to this as *the static digital twin placement scheme*. However, the mobility of physical objects and users can be uncertain for a given time horizon [23], and the digital twin placement without the knowledge of future user query arrivals is likely to result in unacceptable AoI query results. Therefore, it is necessary to determine whether to replace existing digital twins in the beginning of each time slot, i.e., remove some existing digital twins and instantiate new digital twins at each time slot, and we refer to this as *the dynamic digital twin placement scheme*. Thus, it is challenging to deploy digital twins in cloudlets under both static and dynamic digital twin placement schemes to maximize user satisfaction augmentation, subject to computing capacities on cloudlets.

The novelty of this paper lies in the introduction of a new metric to measure user satisfaction on AoI-aware service provisioning in an MEC network empowered by digital twins. We consider two novel utility maximization problems under static and dynamic digital twin placement schemes, respectively, with the aim to maximize the user service satisfaction through efficient digital twin placements. Performance-guaranteed approximation and online algorithms for the defined problems then are devised.

The main contributions of this paper are given as follows.

- We explore the mobility of physical objects and dynamic query request arrivals in an MEC network for a finite time horizon, enabling efficient query service provisioning based on digital twin data. We formulate two user service satisfaction enhancement problems in the MEC network: the static utility maximization problem and the dynamic utility maximization problem, and show their NP-hardness.
- We formulate an Integer Linear Programming (ILP) solution to the static utility maximization problem when the problem size is small; otherwise, we propose an approximation algorithm with a guaranteed approximation ratio for the problem.
- We devise an online algorithm with a provable competitive ratio for the dynamic utility maximization problem.
- We evaluate the performance of the proposed algorithms for user satisfaction on service provisioning in the MEC network via simulations. Simulation results demonstrate that the proposed algorithms are promising, and outperform the comparison baseline algorithms, improving by no less than 10.7% of the performance, compared to that of the baseline algorithms.

The rest of the paper is arranged as follows. The related work is surveyed in Section II. Section III introduces the system model, notions and notations. Section IV introduces problem definitions and shows the NP-hardness of the defined problems. Section V formulates an ILP solution and devises an approximation algorithm for the static utility maximization problem, respectively. Section VI develops an online algorithm for the dynamic utility maximization problem. Section VII performs the performance evaluation of the proposed algorithms. The paper is concluded in Section VIII.

II. Related Work

Mobility-aware delay-sensitive service provisioning in MEC has been extensively studied in the past [2], [5], [8], [15], [23], [25], [27], [29], [31], [32], [42]. For example, Gao et al. [5] studied dynamic access network selections and service placements with user mobility in MEC by developing an online algorithm to improve the service quality through balancing service delays. Hu et al. [8] addressed uncertain channel conditions due to the mobility of wireless user devices in MEC networks. They aimed to alleviate energy consumption and service delay by leveraging Reinforcement Learning (RL) methods for energy and computing resource allocations. Li et al. [16] considered the budget-constrained service satisfaction enhancement problem in MEC by developing performance-guaranteed approximation algorithms. Li et al. [19] also considered delay-sensitive IoT applications in MEC, and developed efficient algorithms under static and dynamic user request settings to enhance accumulative user service satisfaction. Xu et al. [39] dealt with service provisioning in an MEC network to minimize the service cost of IoT applications, via virtualized network function instance placements. Considering the high mobility and delay requirements of users, Ma et al. [23] proposed approximation and online algorithms for enabling seamless services in MEC. Maleki et al. [25] considered user mobility and uncertain application specifications for task offloading in MEC, and devised two efficient online algorithms to mitigate the turnaround time. Polese et al. [27] applied machine learning algorithms to predict user mobility patterns at network edges. Wang et al. [31] investigated the coordination of delay-aware micro-services with user mobility in MEC. They developed a dynamic programming-based offline algorithm to mitigate the overall service delay, and a RL-based online algorithm to obtain a near-optimal solution. However, the above-mentioned studies did not consider the freshness issue of information in network service provisioning in MEC.

Recently, the Age of Information (AoI) as a new metric to capture the freshness of information has attracted lots of attention. Intensive efforts have been taken to minimize the AoI in service provisioning in MEC [3], [4], [7], [14], [34], [36], [37], [43]. For example, Chen et al. [3] addressed a joint scheduling problem on transmitting data and replenishing energy with directional chargers. They designed approximation algorithms to minimize the peak AoI. He et al. [7] considered the system dynamics in monitoring and controlling systems by developing a RL-based online algorithm for AoI minimization. Xu et al. [37] focused on minimizing the AoI of big data analytics with uncertain network delays by proposing efficient algorithms. They developed efficient approximation algorithms for the problem. Wang et al. [34] formulated a minimization problem of the average AoCI on mobile clients by introducing a novel metric, Age-of-Critical-Information (AoCI), and developed a heuristic algorithm for the problem in an offline manner, along with an imitation learning-based algorithm for the online problem. Zou et al. [43] explored the non-trivial trade-off between the transmission delay and preprocessing delay, and designed algorithms under different queue management schemes to minimize the average AoI and the average peak AoI. However, none of the above studies ever considered query service provisioning based on digital twin data in MEC.

Mobility-aware service provisioning in MEC empowered by digital twins has been investigated recently [10], [11], [21], [28], [35]. For instance, Lei et al. [10] established digital twins to monitor the life cycles of Unmanned Aerial Vehicles (UAVs) with high mobility, and proposed a machine learning algorithm to optimize the behaviors of UAVs. 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 the total energy consumption of UAVs. Li et al. [12], [13] addressed digital twin-assisted service provisioning problems in MEC and proposed efficient algorithms for reliable service function chain-enabled network services. Lin et al. [21] 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 collected profit. Sun et al. [28] utilized digital twins of edge servers to predict their states under uncertain user mobility, along with developing a DRL algorithm to mitigate the offloading delay of users. Wang et al. [35] designed a Mobility Digital Twin (MDT) architecture for transportation in MEC, i.e. creating digital twins for humans, vehicles and traffic infrastructures to implement functionalities such as modelling, learning, and prediction, thereby providing efficient mobility services. However, the mentioned studies did not consider the user service satisfaction augmentation based on the AoI of query results by deploying digital twins in MEC networks.

Unlike the aforementioned studies, in this paper we investigate mobility-aware service provisioning in MEC that is leveraged by the digital twin technique. We propose a novel metric to quantify user satisfaction measured by the AoIs of query results. It must be mentioned that this journal version is an extension of a conference paper [20].

III. PRELIMINARIES

In this section, we introduce the system model, and give notions and notations used in this paper.

A. System Model

Consider an 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. Assume that each AP is co-located with a cloudlet via an optical fiber cable, and the communication delay between the AP and its co-located cloudlet is negligible [23]. Denote by v a cloudlet or its co-located AP with $v \in V$ for simplicity. Denote by cap_v the available amount of computing resource in 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 via the gateway is far larger than that between any pair of APs. 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) via the gateway. E is the set of links connecting APs. Denote by d_e the transmission delay of transmitting a unit of data through link $e \in E$ [38].

We assume that the MEC network runs in a discrete-time manner, i.e., a given 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. We further assume that the values of delays mentioned in this paper (e.g., d_e) are normalized by the length of each time slot. E.g., supposing the length of each time slot is 50 ms, a transmission delay of 10 ms is treated as 0.2 time slots. An illustrative example of the system model is shown in Fig. 1.

B. Submodular Function, Approximation Ratio and Competitive Ratio

Definition 1: 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 submodular if (i) $f(\emptyset) = 0$, and (ii) for every two sets $A, B \subseteq \Omega$ with $A \subseteq B$ and every $s \in \Omega \setminus B$, $f(A \cup \{s\}) - f(A) \geq f(B \cup \{s\}) - f(B)$.

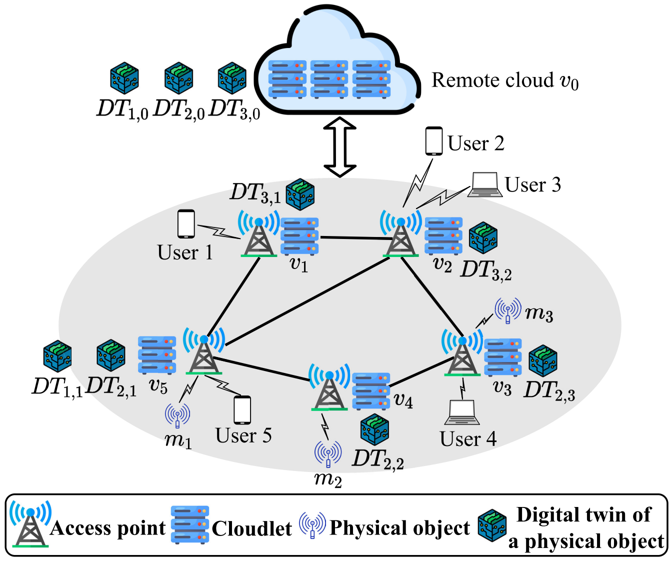


Fig. 1. An MEC network consists of a remote cloud v_0 and 5 APs with each AP co-located with a cloudlet. There are 5 users issuing queries for digital twin data of physical objects m_1, m_2 and m_3, which have digital twins $DT_{1,0}$, $DT_{2,0}$ and $DT_{3,0}$ deployed in the remote cloud v_0, respectively. Moreover, physical object m_1 has a digital twin $DT_{1,1}$ deployed in cloudlet v_5. Physical object m_2 has digital twins $DT_{2,1}$, $DT_{2,2}$ and $DT_{2,3}$ deployed in cloudlets v_5, v_4 and v_3, respectively. Physical object m_3 has digital twins $DT_{3,1}$ and $DT_{3,2}$ deployed in cloudlets v_1 and v_2, respectively.

Definition 2: An approximation algorithm is with an approximation ratio γ for a maximization problem instance if the solution delivered by the algorithm is no less than $\gamma \cdot OPT$, where OPT is the optimal solution of the problem instance and $0 < \gamma \leq 1$.

Definition 3: An online algorithm is with a competitive ratio α for a dynamic maximization problem instance if the solution delivered by the algorithm is no less than $\alpha \cdot OPT_{off}$ with $0 < \alpha \leq 1$, where OPT_{off} is the optimal solution of the offline instance of the dynamic maximization problem.

All symbols used in this paper are listed in Table I.

IV. Problem Formulations

In this section, we first introduce cost modelling of data freshness of digital twins, which is measured by the Age of Information (AoI). We then introduce the concepts of digital twin state updating and synchronization. We finally define problems and show the NP-hardness of the defined problems.

A. 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, there is a digital twin for each object $m \in M$ 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 on cloudlets. Denote by c_m the amount of computing resource consumed by 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, where $\delta_m \geq 1$ is 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 maintain synchronizations between objects and their digital twins. We assume that the delay of removing a digital twin is negligible [33], while instantiating a digital twin for an object m has a delay of d_m^{ins}. However, it must be mentioned that frequent removals and instantiations of digital twins will result in high service delays, dramatically downgrading the quality of query services.

B. AoI at Digital Twins

Assuming that 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', and the AoI of the digital twin is defined as $(t-t')$ [3]. In the following, we show the evolution of the AoI at a digital twin by distinguishing from its instantiation time: the digital twin is instantiated at time slot $t = 0$ or $t \geq 1$.

We 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 object 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 a shortest path in G between v_1 and v, where d_m^{ins} is the instantiation delay of the digital twin in v for object m. We assume that the initial AoI of a 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. The AoI of the digital twin increases from time $a_m \cdot d_{v_1,v} + d_m^{ins}$ to time $\delta_m + a_m \cdot d_{v_2,v} + \sigma_m$, and the AoI decreases to $a_m \cdot d_{v_2,v} + \sigma_m$ if object m sends its update every δ_m time slots, where v_2 is the location of m at time δ_m, and σ_m is the update delay of a digital twin of object m after receiving its update data. The AoI evolution of the digital twin procedure continues until the end of the time horizon or it is removed from the system.

We use an example in Fig. 2 to illustrate the evolution of AoI of object m at the digital twin deployed in node $v \in V \cup \{v_0\}$ at time slot 0 as follows. Given time horizon $\mathbb{T} = \{0, 1, 2, 3, 4, 5\}$, let object m send 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 slot 2 and 4, respectively. We assume that a digital twin of object m is deployed in node $v \in V \cup \{v_0\}$ at time slot 0 through adopting the first update of m at time slot 0, which will not be removed. The data of the digital twin of object m in cloudlet v will be available at

TABLE I
TABLE OF SYMBOLS

Notations	Descriptions
$G=(V\cup\{v_0\},E)$	An MEC network, where V is the set of APs with each AP co-located with a cloudlet (V also denotes the set of cloudlets), node v_0 is a remote cloud, and E is the set of links connecting APs
v and cap_v	An AP (or a cloudlet), and the available amount of computing resource in cloudlet v
d_{v,v_0} and $d_{v_0,v}$	The transmission delay of transmitting a unit of data from AP v to the remote cloud v_0 via the gateway, and the transmission delay of transmitting a unit of data from the remote cloud v_0 to AP v via the gateway
d_e	The transmission delay of transmitting a unit of data through link $e\in E$
$\mathbb{T}=\{0,1,2,\ldots,\lvert\mathbb{T}\rvert-1\}$	The monitoring time horizon
M and m	A set of physical objects, and an object $m\in M$
c_m	The computing resource consumption of a digital twin of object $m\in M$ in a cloudlet
δ_m	The digital twin of object $m\in M$ is synchronized every δ_m time slots with $\delta_m\geq 1$ a positive integer
a_m	The data size of each update of object m
d_m^{ins}	The instantiation delay of the digital twin for object m
σ_m	The update delay of a digital twin of object m after receiving its update data
Q, Q_t and q	A set Q of queries demanding the digital twin data, a set $Q_t\subseteq Q$ of queries issued at time t, and a query $q\in Q$
t_q, loc_q, m_q and s_q	t_q is the issuing time of query q, loc_q is the issuing location of query q (under the coverage of AP loc_q), m_q is the requested object of query q, and s_q is the data size of the query result of query q
$\mathcal{V}_{m,t}$	The set of nodes $\mathcal{V}_{m,t}\subseteq V\cup\{v_0\}$, and the digital twins of object $m\in M$ are deployed in nodes $\mathcal{V}_{m,t}$ at time slot t
$w_{m,v,t}$	The AoI at the digital twin of object m in node $v\in\mathcal{V}_{m,t}$ at time slot t
v_q	Query q retrieves 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
W_{q,v_0} and W_{q,v_q}	The AoI of the query result of query q when retrieving the data at the digital twin in remote cloud v_0 and node v_q
u_q	The utility gain u_q of user service satisfaction augmentation for query $q\in Q$ defined by Eq. (1)
$x_{m,v}$	Binary decision variable, showing whether a digital twin of object m is deployed in cloudlet $v\in V$
$y_{q,v}$	Binary decision variable, showing whether 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
$f(\mathcal{A})$	The utility function of deploying digital twins $\mathcal{A}$ into cloudlets of V
$n_{m,v}$	A digital twin of object $m\in M$ deployed in cloudlet $v\in V$
N	$N=\{n_{m,v}\mid m\in M, v\in V\}$ is the set of potential digital twin deployments in cloudlets
n^l	The lth deployed digital twin in cloudlets with $n^l\in N$
$\mathcal{A}^{l-1}$	The set of the first $l-1$ digital twins deployed prior to the deployment of the lth digital twin
$\Delta f(n^l\mid\mathcal{A}^{l-1})$	The marginal utility gain of deploying digital twin n^l in a cloudlet by Eq. (9)
$\rho(n^l)$	The ratio of the marginal utility gain of deploying n^l to its computing resource consumption by Eq. (10)
$\mathcal{C}_v(\mathcal{A}^{l-1})$	The accumulative computing resource consumption of cloudlet v, via placing digital twins in $\mathcal{A}^{l-1}$
S_1 and S_2	Two disjoint sets of digital twins with $\mathcal{A}=S_1\cup S_2$
$\mathcal{W}_t^D$ and $\mathcal{W}_t^S$	The dynamic and static AoI of queries in Q_t issued at time t
β	A control parameter to bound the accumulative instantiation delays

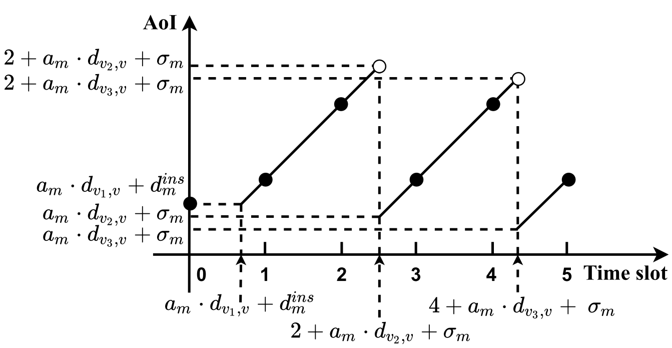


Fig. 2. An illustrative example of the AoI evolution at the digital twin of object m in node $v\in V\cup\{v_0\}$ with the digital twin deployed at time 0.

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 the digital twin in node v at time slot 0 is $a_m\cdot d_{v_1,v}+d_m^{ins}$. The data of the digital twin becomes stale until time $2+a_m\cdot d_{v_2,v}+\sigma_m$, and its AoI decreases to $a_m\cdot d_{v_2,v}+\sigma_m$ as object m moves to AP v_2 at time slot 2 and sends its update. The AoI of the digital twin increases until time $4+a_m\cdot d_{v_3,v}+\sigma_m$, and decreases to $a_m\cdot d_{v_3,v}+\sigma_m$, with object m moving to v_3 at time slot 4 and sending its update.

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

We use an example in Fig. 3 to show the AoI evolution of the digital twin of object m deployed in cloudlet v' at time slot 3. Assuming object m moves to v_2 at time slot 2 and the latest update of object m at time slot 3 is generated at time slot 2, the data at the digital twin in v' will be available at time

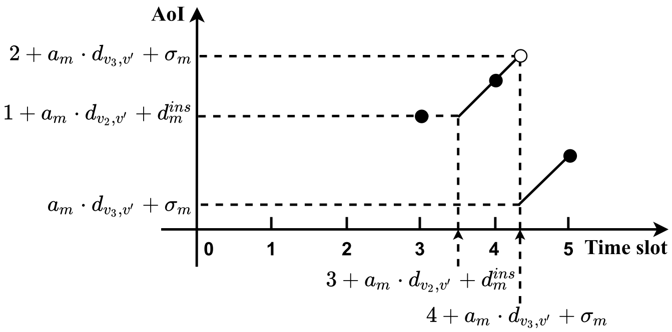


Fig. 3. An illustrative example of the AoI evolution at the digital twin of object m in cloudlet $v' \in V$ with the digital twin deployed at time 3.

slot $3+a_m \cdot d_{v_2,v'}+d_m^{ins}$ 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 slot 4. Then the AoI at the digital twin increases after time $3 + a_m \cdot d_{v_2,v'} + d_m^{ins}$ until time $4 + a_m \cdot d_{v_3,v'} + \sigma_m$, and decreases to $a_m \cdot d_{v_3,v'} + \sigma_m$.

C. AoI of a Query Result and the Utility Gain

Given a set Q of user service queries that demand the digital twin data of different objects at different time slots for 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 its requested object, and s_q is the size of its query result.

Denote by $\mathcal{V}_{m,t} \subseteq V \cup \{v_0\}$ the set of nodes, in which the digital twins of object $m \in M$ are deployed at time slot t. Let $w_{m,v,t}$ be the AoI of 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 of data through a shortest path in G 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 of digital twins in remote cloud v_0. This however leads to high AoIs of the query results. Therefore, digital twins are deployed in cloudlets of V to provide fresher data for user queries, thereby enhancing user service satisfaction. We use *the utility gain* u_q to capture user service satisfaction enhancement 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 of the digital twin in remote cloud v_0. Utility gain u_q implies the AoI reduction of the query result through deploying digital twins in cloudlets of V.

D. Problem Definitions

We consider two digital twin placement schemes in cloudlets of V with the aim to maximize the total utility gain of queries in a set 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 [25]. 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 $\mathbb{T}$ under the static digital twin placement scheme, subject to computing capacities on cloudlets, with the aim to maximize the total utility gain of queries.

Definition 4: Given an MEC network $G = (V \cup \{v_0\}, E)$, a given time horizon $\mathbb{T}$, a set Q of queries, a set M of objects with each having a 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 admitted queries in Q, by deploying digital twins for each object in M in cloudlets under the static digital twin placement scheme, subject to computing capacities on cloudlets.

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 the query result when query q retrieves the data of the digital twin in v.

Let $x_{m,v}$ be a binary variable, where $x_{m,v} = 1$ means 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 variable, where $y_{q,v} = 1$ means that query q retrieves the data of the digital twin of object m_q in node $v \in V \cup \{v_0\}$ with the minimum AoI of the query result; and $y_{q,v} = 0$ otherwise. The Integer Linear Programming (ILP) solution to the static utility maximization problem is given as follows.

$$\textbf{ILP}: \text{Maximize} \sum\nolimits_{q \in Q} \sum\nolimits_{v \in V \cup \{v_0\}} y_{q,v} \cdot (W_{q,v_0} - W_{q,v}), \tag{2}$$

$$\text{subject to}: \quad \sum\nolimits_{m \in M} c_m \cdot x_{m,v} \le cap_v, \quad \forall v \in V \tag{3}$$

$$\sum\nolimits_{v \in V \cup \{v_0\}} y_{q,v} = 1, \quad \forall q \in Q \tag{4}$$

$$y_{q,v} \le x_{m_q,v}, \quad \forall q \in Q,\ \forall v \in V \tag{5}$$

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

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

where objective (2) is the total utility gain of admitted queries by the utility gain definition (1). Constraint (3) ensures that no computing capacity on any cloudlet is violated. Constraint (4) enforces that each query q selects only one digital twin of object m_q in node $v \in V \cup \{v_0\}$ to minimize the AoI of its query result. Constraint (5) shows query q can retrieve the data of the digital twin of object m_q in cloudlet v only if the digital twin of m_q has been deployed in cloudlet v already.

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

Definition 5: Given an MEC network $G = (V \cup \{v_0\}, E)$, a given time horizon $\mathbb{T}$, a set Q_t of queries in the beginning of each time slot $t \in \mathbb{T}$, a set M of highly mobile objects, and a set V of cloudlets, the *dynamic utility maximization problem* in G is to maximize the accumulative utility gain of admitted 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 computing capacities on cloudlets.

E. NP-Hardness of the Defined Problems

Theorem 1: The static utility maximization problem in an MEC network $G = (V \cup \{v_0\}, E)$ is NP-hard.

Proof: We show the NP-hardness of the problem through a reduction from a NP-hard problem - the knapsack problem [26]. 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, while all user queries are issued through AP v. The reduction is as follows.

In a knapsack problem instance, there is a bin with capacity of cap_v and there are $|M|$ 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 placing a digital twin of m in cloudlet v. The knapsack problem is to maximize the total profit by packing items into the bin, subject to the bin capacity. The knapsack problem [26] is equivalent to this special static utility maximization problem. The static utility maximization problem thus is NP-hard. ■

Corollary 1: The dynamic utility maximization problem in an MEC network $G = (V \cup \{v_0\}, E)$ is NP-hard.

Proof: We consider a special dynamic utility maximization problem where all queries for the given finite time horizon are known in advance. All digital twins are deployed in the beginning of the time horizon, and there is no digital twin replacement for the time horizon. This special dynamic utility maximization problem becomes a static utility maximization problem, which has been shown to be NP-hard by Theorem 1. Thus, the dynamic utility maximization problem is NP-hard too. ■

V. Approximation Algorithm for the Static Utility Maximization Problem

In this section, we propose an approximation algorithm for the static utility maximization problem.

A. Approximation Algorithm

Denote by $\mathcal{A}$ the set of digital twins placed for objects in M in cloudlets of 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 to set $\mathcal{A}$. Let $\mathbb{V}_m(\mathcal{A}) \subseteq V \cup \{v_0\}$ be the set of nodes in which the digital twins of object m are placed. 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 in $\mathcal{A}$ to cloudlets in V as follows.

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

We aim to identify a set $\mathcal{A}$ of digital twins deployed in cloudlets in the beginning of the time horizon to maximize the total utility gain of queries.

Recall that $n_{m,v}$ 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 that digital twins in $\mathcal{A}^{l-1}$ have been deployed in cloudlets, *the marginal utility gain* of deploying digital twin n^l in a cloudlet is defined as follows.

$$\Delta f(n^l \mid \mathcal{A}^{l-1}) = f(\mathcal{A}^{l-1} \cup \{n^l\}) - f(\mathcal{A}^{l-1}). \tag{9}$$

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.

$$\rho(n^l) = \Delta f(n^l \mid \mathcal{A}^{l-1})/c(n^l). \tag{10}$$

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

The approximation algorithm proceeds iteratively, 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 $\rho(n_{m',v'})$ by Eq. (10), and the computing capacity of cloudlet v' has not fully consumed prior to the deployment of this digital twin, i.e., $\mathcal{C}_{v'}(\mathcal{A}^{l-1}) < cap_{v'}$, where $cap_{v'}$ is the computing capacity of 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 capacity of cloudlet v' may be violated after deploying n^l, i.e., $\mathcal{C}_{v'}(\mathcal{A}^l) = \mathcal{C}_{v'}(\mathcal{A}^{l-1}) + c(n^l) > cap_{v'}$. Two sets S_1 and S_2 of digital twins are constructed to avoid resource violations, where $S_1 = \emptyset$ and $S_2 = \emptyset$ initially. If deploying n^l causes no resource violation, n^l is added to set S_1; otherwise, n^l is added to set S_2. This digital twin placement procedure continues until no more digital twins can be placed 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 to cloudlets will not cause any resource violation by Theorem 2, which will be shown later.

The proposed approximation algorithm chooses one of S_1 and S_2 with the larger utility gain as the problem solution. The proposed algorithm is detailed in `Algorithm 1`.

B. Algorithm Analysis

The rest is to analyze the performance of the proposed approximation algorithm and its time complexity.

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 gain of all queries under the static digital twin placement scheme.

Identify the shortest path between each pair of nodes in $V \cup \{v_0\}$;
$\mathcal{A}^0 \leftarrow \emptyset$; $S_1 \leftarrow \emptyset$; $S_2 \leftarrow \emptyset$;
$\mathcal{C}_v(\mathcal{A}^0) \leftarrow 0, \forall v \in V$;
$N \leftarrow \{n_{m,v} \mid m \in M, v \in V\}$;
$l \leftarrow 1$;
while $N \backslash \mathcal{A}^{l-1} \neq \emptyset$ **and** it exists a cloudlet v' with $\mathcal{C}_{v'}(\mathcal{A}^{l-1}) < cap_{v'}$ **do**
  Identity a digital twin $n_{m',v'} \in N \backslash \mathcal{A}^{l-1}$ as n^l with the largest $\rho(n^l)$ by Eq. (10) and $\mathcal{C}_{v'}(\mathcal{A}^{l-1}) < cap_{v'}$;
  $\mathcal{A}^l \leftarrow \mathcal{A}^{l-1} \cup \{n^l\}$;
  $\mathcal{C}_{v'}(\mathcal{A}^l) \leftarrow \mathcal{C}_{v'}(\mathcal{A}^{l-1}) + c(n^l)$;
  if $\mathcal{C}_{v'}(\mathcal{A}^l) > cap_{v'}$ **then**
    $S_2 \leftarrow S_2 \cup \{n^l\}$;
  else
    $S_1 \leftarrow S_1 \cup \{n^l\}$;
  end if;
  $l \leftarrow l + 1$;
end while;
if $f(S_1) \geq f(S_2)$ **then**
  return S_1 and $f(S_1)$;
else
  return S_2 and $f(S_2)$;
end if;

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 A and B two sets of digital twins placed in cloudlets of V with $A \subseteq B$, then $\mathbb{V}_m(A) \subseteq \mathbb{V}_m(B), \forall m \in M$, i.e., more digital twins of each object are deployed by B than that by A. Let $n_{m',v'} \notin B$ be a digital twin of object m' in cloudlet v'. Let $A' = A \cup \{n_{m',v'}\}$ and $B' = B \cup \{n_{m',v'}\}$, then $A' \subseteq B'$. Following the submodular function definition 1, we show $f(A') - f(A) \geq f(B') - f(B)$ as follows.

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

$$\begin{aligned} f(A') - f(A) &= \sum_{q \in Q} \min_{v \in \mathbb{V}_{m_q}(A)}\{W_{q,v}\} - \sum_{q \in Q} \min_{v \in \mathbb{V}_{m_q}(A')}\{W_{q,v}\} \\ &= \sum_{q \in Q_{m'}} \Big(\min_{v \in \mathbb{V}_{m'}(A)}\{W_{q,v}\} - \min_{v \in \mathbb{V}_{m'}(A')}\{W_{q,v}\} \Big) \\ &= \sum_{q \in Q^{v'}_{m'}(A')} \Big(\min_{v \in \mathbb{V}_{m'}(A)}\{W_{q,v}\} - W_{q,v'} \Big), \end{aligned} \tag{11}$$

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

$$f(B') - f(B) = \sum_{q \in Q^{v'}_{m'}(B')} \Big(\min_{v \in \mathbb{V}_{m'}(B)}\{W_{q,v}\} - W_{q,v'} \Big). \tag{12}$$

By Eq. (11), $\min_{v \in \mathbb{V}_{m'}(A)}\{W_{q,v}\} \geq W_{q,v'}, \forall q \in Q^{v'}_{m'}(A')$, and $Q^{v'}_{m'}(B') \subseteq Q^{v'}_{m'}(A')$, we have

$$f(A') - f(A) \geq \sum_{q \in Q^{v'}_{m'}(B')} \Big(\min_{v \in \mathbb{V}_{m'}(A)}\{W_{q,v}\} - W_{q,v'} \Big). \tag{13}$$

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

Lemma 2: Given a 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, let $\mathcal{A}^{opt}_v$ be the set of digital twins placed in cloudlet v by $\mathcal{A}^{opt}$ with $\mathcal{A}^{opt} = \cup_{v \in V} \mathcal{A}^{opt}_v$. Then, (i) $\rho(n^l) \geq \frac{\Delta f(n^* \mid \mathcal{A})}{c(n^*)}, \forall v \in V, \forall n^l \in \mathcal{A}_v$, and $\forall n^* \in \mathcal{A}^{opt}_v \setminus \mathcal{A}_v$; and (ii) $f(\mathcal{A}) \geq \sum_{n^* \in \mathcal{A}^{opt} \setminus \mathcal{A}} \Delta f(n^* \mid \mathcal{A})$.

Proof: (i) If $\mathcal{A}^{opt}_v \setminus \mathcal{A}_v = \emptyset$, then $\Delta f(n^* \mid \mathcal{A}) = 0$, and the lemma follows. Otherwise by the definition of $\rho(n^l)$ in Eq. (10), $\forall v \in V, \forall n^l \in \mathcal{A}_v, \forall n^* \in \mathcal{A}^{opt}_v \setminus \mathcal{A}_v$, we have

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

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

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 $\mathcal{C}_v(\mathcal{A}^{l-1}) < cap_v$ 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.

(ii) Let $n^*_{v,max} = \arg\max_{n^* \in \mathcal{A}^{opt}_v \setminus \mathcal{A}_v} \frac{\Delta f(n^* \mid \mathcal{A})}{c(n^*)}, \ \forall v \in V$. Recall that set $\mathcal{A}$ is delivered by `Algorithm` 1, then

$$\begin{aligned} f(\mathcal{A}) &= \sum_{l=1}^{|\mathcal{A}|} \Delta f(n^l \mid \mathcal{A}^{l-1}) = \sum_{v \in V} \sum_{n^l \in \mathcal{A}_v} \Delta f(n^l \mid \mathcal{A}^{l-1}) \\ &= \sum_{v \in V} \sum_{n^l \in \mathcal{A}_v} \rho(n^l) \cdot c(n^l) \geq \sum_{v \in V} \frac{\Delta f(n^*_{v,max} \mid \mathcal{A})}{c(n^*_{v,max})} \sum_{n^l \in \mathcal{A}_v} c(n^l) \end{aligned} \tag{16}$$

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

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

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

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

Theorem 2: Given an MEC network $G = (V \cup \{v_0\}, E)$, a set Q of queries, a set M of objects with their mobility profiles, a finite time horizon $\mathbb{T}$, and a set V of cloudlets, there is a $\frac{1}{4}$-approximation algorithm, `Algorithm 1`, for the static utility maximization problem in G, 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 with the total utility gain $f(\mathcal{A}^{opt})$. With $\mathcal{A} = S_1 \cup S_2$ the solution delivered by `Algorithm 1`, then

$$f(\mathcal{A}^{opt}) \leq f(\mathcal{A} \cup \mathcal{A}^{opt})$$

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

$$\leq 2 \cdot f(\mathcal{A}). \tag{21}$$

Ineq. (20) holds because $f(\cdot)$ is a submodular function by Lemma 1, and Ineq. (21) holds by Lemma 2.

Since $\mathcal{A}$ is divided into two sets S_1 and S_2, we have

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

We then choose one from S_1 and S_2 with a larger utility gain as the final solution to the problem, i.e.,

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

We finally show that the deployment of digital twins in S_1 or S_2 does not result in any computing resource violations of cloudlets. 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 on a cloudlet, and that cloudlet then is excluded for further digital twin placements. Therefore, each cloudlet is assigned with at most one digital twin in S_2.

The rest is to analyze the time complexity of `Algorithm 1` as follows. It takes $O(|V|^3)$ time to identify the shortest path in G between each pair of nodes in $V \cup \{v_0\}$. The construction of $\mathcal{A}$ examines at most $|M| \cdot |V|$ digital twins for deployments. At each iteration, we identify a digital twin n^l with the largest $\rho(n^l)$, considering all queries for the object of n^l, which takes the time $O(|M| \cdot |V| \cdot |Q|)$. Thus, the running time of `Algorithm 1` is $O(|M|^2 \cdot |V|^2 \cdot |Q| + |V|^3)$. The theorem then follows. ■

VI. Online Algorithm for the Dynamic Utility Maximization Problem

In this section, we devise an online algorithm for the dynamic utility maximization problem. We also analyze the competitive ratio of the online algorithm.

A. Online Algorithm

Under the dynamic digital twin placement scheme, the mobility of objects in M and a set $Q_t \subseteq Q$ of queries issued at time slot t are revealed in the beginning of 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 the knowledge of future query arrivals. However, the accumulative instantiation delay of queries that retrieve newly instantiated digital twins is prohibitively large for the given time horizon, which will dramatically downgrade user satisfaction on query services.

Inspired by the work in [41], we introduce a digital twin replacement control policy to upper bound the accumulative instantiation delay as follows.

Given time slot $t \geq 1$, we define the *dynamic AoI* $\mathcal{W}_t^D$ of query results of queries in Q_t as the accumulative instantiation delay 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, and 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. Let $\mathcal{W}_0^D = 0$ because cloudlets are empty initially, and it is unnecessary to take the dynamic AoI into consideration at time slot 0. Recall that W_{q,v_q} is the minimum AoI of the query result of query q. We 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 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 proceeds iteratively. Specifically, at time slot 0, all cloudlets in V are empty initially, and we first apply `Algorithm 1` to obtain set $\mathbb{S}_0$ of digital twins for deployment in cloudlets for queries Q_0, with the given locations of objects at time slot 0. At each time slot $t \geq 1$, we calculate the AoI of each potential digital twin $n_{m,v}$, based on whether $n_{m,v}$ was deployed in node v at time $t-1$ or not, as shown in Section IV-B. We then invoke `Algorithm 1` to obtain set $\mathbb{S}_t$ of digital twins for queries Q_t at time slot $t \geq 1$. However, $\mathbb{S}_t$ cannot be applied for the digital twin replacement at time slot $t \geq 1$ directly, and a digital twin replacement control policy is designed to avoid large instantiation delays as follows.

Denote by $\hat{t}$ 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 delay. If the dynamic AoI of the query results at time slot t is no greater than $\frac{1}{\beta}$ times the accumulative utility gain of queries from time slot $\hat{t}$ to $(t-1)$ without considering the dynamic AoI, i.e., $\mathcal{W}_t^D \leq \frac{1}{\beta} \cdot \sum_{t'=\hat{t}}^{t-1} (\sum_{q\in Q_{t'}} W_{q,v_0} - \mathcal{W}_{t'}^S)$, we apply $\mathbb{S}_t$ for digital twin replacements at time slot t. Otherwise, we keep the digital twin placement in the last time slot $t-1$. It is noted that a smaller β implies more tolerance on the instantiation delay,

Algorithm 2 Online Algorithm for the Dynamic Utility Maximization Problem

Input: An MEC network $G = (V \cup \{v_0\}, E)$, a given finite time horizon $\mathbb{T}$, a set Q_t of queries at each time slot t, and a set M of objects with high mobility without any future information.

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

Deploy digital twins of set $\mathbb{S}_0$ to cloudlets in V at time slot 0, where $\mathbb{S}_0$ can be obtained by invoking `Algorithm 1` for queries in 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 in 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**
    Replace digital twins by applying set $\mathbb{S}_t$;
    $\hat{t} \leftarrow t$;
  end if;
  $t \leftarrow t + 1$;
end while.

leading to more frequent digital twin replacements. Otherwise, a larger β indicates less tolerance on the instantiation delay, thereby performing less frequent digital twin replacements. The proposed algorithm is detailed in `Algorithm 2`.

B. Algorithm Analysis

The rest is to analyze the competitive ratio of `Algorithm 2` for the dynamic utility maximization problem as follows.

Lemma 3: The total dynamic AoI of query results is no larger than $\frac{1}{\beta}$ times the total utility gain of queries in Q for time horizon $\mathbb{T}$ by `Algorithm 2` without considering the dynamic AoI of queries, i.e.,

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

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 $\hat{t}_L \le |\mathbb{T}|-1$ be the time slot that the last digital twin replacement occurs during $\mathbb{T}$. By the digital twin replacement control policy, we have

$$\begin{aligned}\sum_{t=0}^{|\mathbb{T}|-1} \mathcal{W}_t^D &= \sum_i \mathcal{W}_{\hat{t}_i}^D \le \frac{1}{\beta} \cdot \sum_{t=0}^{\hat{t}_L - 1} (\sum_{q \in Q_t} W_{q,v_0} - \mathcal{W}_t^S) \\ &\le \frac{1}{\beta} \cdot \sum_{t=0}^{|\mathbb{T}|-1} (\sum_{q \in Q_t} W_{q,v_0} - \mathcal{W}_t^S), \end{aligned} \quad (25)$$

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

Lemma 4: The optimal solution to the dynamic utility maximization problem is no greater than λ times the total static AoI of query results in the solution by `Algorithm 2`, i.e.,

$$U^* \le \lambda (\sum_{t=0}^{|\mathbb{T}|-1} \sum_{q \in Q_t} W_{q,v_0} - \sum_{t=0}^{|\mathbb{T}|-1} \mathcal{W}_t^S), \quad (26)$$

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 [41].

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

$$\sum_{t=0}^{|\mathbb{T}|-1} \sum_{q \in Q_t} W_{q,v_0} - \sum_{t=0}^{|\mathbb{T}|-1} \mathcal{W}_t^{*S} \le \lambda (\sum_{t=0}^{|\mathbb{T}|-1} \sum_{q \in Q_t} W_{q,v_0} - \sum_{t=0}^{|\mathbb{T}|-1} \mathcal{W}_t^S). \quad (27)$$

Recall that U^* is the optimal utility gain of the problem. Then

$$\begin{aligned} U^* &= \sum_{t=0}^{|\mathbb{T}|-1} \sum_{q \in Q_t} W_{q,v_0} - \sum_{t=0}^{|\mathbb{T}|-1} \mathcal{W}_t^{*S} - \sum_{t=0}^{|\mathbb{T}|-1} \mathcal{W}_t^{*D} \\ &\le \sum_{t=0}^{|\mathbb{T}|-1} \sum_{q \in Q_t} W_{q,v_0} - \sum_{t=0}^{|\mathbb{T}|-1} \mathcal{W}_t^{*S} \\ &\le \lambda (\sum_{t=0}^{|\mathbb{T}|-1} \sum_{q \in Q_t} W_{q,v_0} - \sum_{t=0}^{|\mathbb{T}|-1} \mathcal{W}_t^S), \text{ by Ineq.(27).} \end{aligned} \quad (28)$$

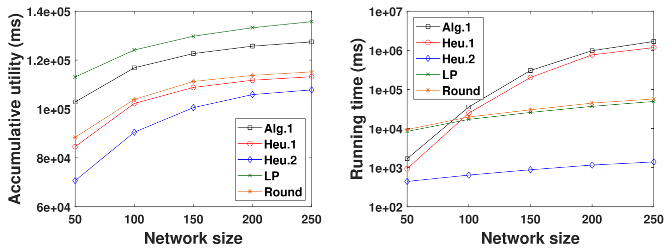
The lemma then follows. ■

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

Proof: We start by analyzing the competitive ratio of `Algorithm 2`. Let U^* and U be the accumulative utility gains by the optimal solution and the solution by `Algorithm 2` for the dynamic utility maximization problem, respectively. We have

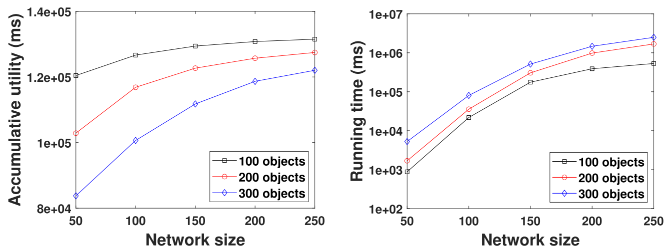
$$\begin{aligned} U &= \sum_{t=0}^{|\mathbb{T}|-1} \sum_{q \in Q_t} W_{q,v_0} - \sum_{t=0}^{|\mathbb{T}|-1} \mathcal{W}_t^S - \sum_{t=0}^{|\mathbb{T}|-1} \mathcal{W}_t^D \\ &\ge \sum_{t=0}^{|\mathbb{T}|-1} \sum_{q \in Q_t} W_{q,v_0} - \sum_{t=0}^{|\mathbb{T}|-1} \mathcal{W}_t^S \\ &\quad - \frac{1}{\beta} \cdot \sum_{t=0}^{|\mathbb{T}|-1} (\sum_{q \in Q_t} W_{q,v_0} - \mathcal{W}_t^S), \text{ by Lemma 3} \\ &\ge (1 - \frac{1}{\beta})(\sum_{t=0}^{|\mathbb{T}|-1} \sum_{q \in Q_t} W_{q,v_0} - \sum_{t=0}^{|\mathbb{T}|-1} \mathcal{W}_t^S) \\ &\ge \frac{1}{\lambda}(1 - \frac{1}{\beta}) U^*, \text{ by Lemma 4.} \end{aligned} \quad (29)$$

We then analyze the time complexity of `Algorithm 2`, which is dominated by invoking `Algorithm 1` at each time slot. Note that finding a shortest path in G between each pair of nodes in $V \cup \{v_0\}$ takes $O(|V|^3)$ time initially. The time complexity of `Algorithm 2` thus is $O(|M|^2 \cdot |V|^2 \cdot |Q_t|_{max} \cdot |\mathbb{T}| + |V|^3)$, by Theorem 2. ■



(a) Accumulative utility (b) Running time

Fig. 4. Performance of different algorithms for the static utility maximization problem.



(a) Accumulative utility (b) Running time

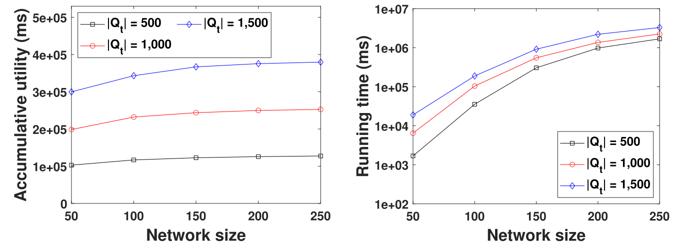
Fig. 5. Impact of the number of objects on the performance of `Alg.1`.

VII. Performance Evaluation

In this section, we evaluated the performance of the proposed algorithms for both static and dynamic utility maximization problems. We also studied impacts of important parameters on the performance of the proposed algorithms.

A. Experimental Settings

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 [6]. The finite time horizon consists of 20 time slots [5], and the length of a time slot is set as 50 ms. There are 200 objects. We s imulate the mobility of objects by the BonnMotion tool, based on the Random Walk model for the given finite time horizon [1]. The computing capacity on each cloudlet is set from $4,000$ MHz to $8,000$ MHz randomly [37]. 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 in the beginning of each time slot, with each query requesting the digital twin data of an object that is randomly chosen. 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, and the query result size per query is set within $[0.5,2]$ MB. The instantiation delay of a digital twin ranges from 20 ms to 40 ms [24]. The transmission delay for transmitting a unit of data (one MB) along a link between a pair of APs is drawn in $[0.2,1]$ ms [38]. The transmission delay for transmitting a unit of data (one MB) between the remote cloud and an AP via the gateway of the MEC network is set within $[2,10]$ ms [31]. The update delay of a digital twin is set within $[1,5]$ ms. The value of parameter β is set at 4.



(a) Accumulative utility (b) Running time

Fig. 6. Impact of the number of queries on the performance of `Alg.1`.

We evaluated algorithm `Algorithm 1`, referred to as `Alg.1`, for the static utility maximization problem, against the following four benchmarks:

(1) `Heu.1`: It iteratively identifies 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`: It considers the cloudlets one by one randomly. For the first cloudlet, it identifies 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`: It adopts the Linear Programming (LP) solution to the problem, i.e., the linear relaxation of its ILP solution (2), serving as an upper bound on its optimal solution.

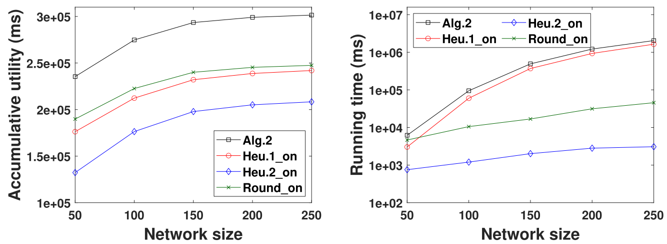
(4) `Round`: It is a variant of the rounding algorithm in [30] that places digital twins in edge servers to minimize the maximum response delay of requests, subject to the maximum AoI constraints, at the expense of constraint violations. Following the spirit of [30], `Round` first obtains a fractional solution by linear relaxation of an ILP solution (2), and then performs random rounding to obtain an integral solution. This solution however is obtained at the expense of computing capacity violations. Therefore, for each cloudlet with capacity violation, algorithm `Round` removes digital twins from the cloudlet one by one randomly until no capacity violation occurs. A feasible solution for the problem thus is obtained.

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

It is mentioned that the value of each figure is the average of results over 30 different topological structures of MEC networks with the same size. The actual running time of each algorithm is based 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.

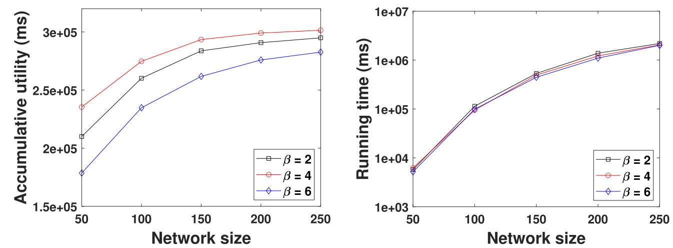
B. Algorithm Performance for the Static Utility Maximization Problem

We first studied the performance of algorithm `Alg.1` for the static utility maximization problem against our benchmark algorithms `Heu.1`, `Heu.2`, `LP` and `Round`, via varying the



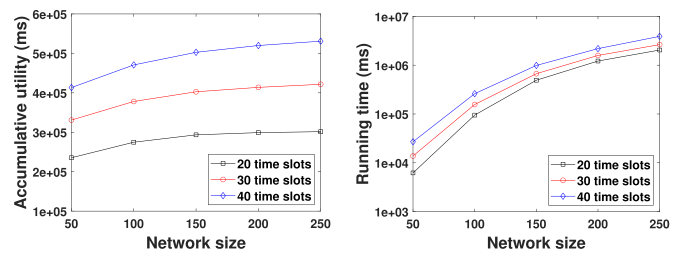
(a) Accumulative utility (b) Running time

Fig. 7. Performance of different algorithms for the dynamic utility maximization problem.



(a) Accumulative utility (b) Running time

Fig. 8. Impact of the parameter β on the performance of Alg.2.



(a) Accumulative utility (b) Running time

Fig. 9. Impact of the number of time slots on the performance of Alg.2.

network size from 50 to 250. Fig. 4 illustrates the accumulative utility and running time of different algorithms. We observe from Fig. 4(a) that when the network size reaches 250, Alg.1 outperforms Heu.1, Heu.2 and Round by 12.5%, 18.2% and 10.7%, respectively, while the performance of Alg.1 is 93.9% of that of LP. The reason is that 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. 4(b), Heu.2 consumes 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 Alg.1. Fig. 5 depicts the performance curves of Alg.1 with 100, 200 and 300 objects, respectively. From Fig. 5(a), the accumulative utility by Alg.1 with 100 objects is 43.8% 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. 5(b) shows that Alg.1 takes the most running time when there are 300 objects, because Alg.1 examines every potential deployment of digital twins at each iteration until no more digital twin can be deployed in any cloudlets.

We finally studied the impact of the number of queries on the performance of Alg.1 when there are 500, $1,000$ and $1,500$ queries issued in the beginning of each time slot. As shown in Fig. 6, we observe from Fig. 6(a) that the accumulative utility by Alg.1 with $|Q_t| = 500$ is 34.3% of that by itself with $|Q_t| = 1,500$ when the network size is fixed at 250. This is because more utility gain can be earned when more queries are admitted. Moreover, Alg.1 with $|Q_t| = 1,500$ takes the most running time which can be seen from Fig. 6(b).

C. Algorithm Performance for the Dynamic Utility Maximization Problem

In the following, we first evaluated the performance of algorithm Alg.2 for the dynamic utility maximization problem against benchmarks Heu.1_on, Heu.2_on and Round_on, by varying network size from 50 to 250, respectively. Fig. 7 plots the accumulative utility and running time curves of different algorithms. Fig. 7(a) demonstrates that Alg.2 outperforms Heu.1_on, Heu.2_on and Round_on by 24.5%, 44.2% and 21.8% respectively when the network size is 250. The rationale behind is that Alg.2 employs an efficient control policy of digital twin replacements to ensure the freshness of query results without the knowledge of future query arrivals.

We then studied the impact of parameter β on the performance of Alg.2. Fig. 8 plots the performance curves of Alg.2 when $\beta = 2$, 4, and 6, respectively. It can be seen from Fig. 8(a) that when the network size is 50, the performance of Alg.2 with $\beta = 4$ is 12.1% and 31.8% higher than that by itself with $\beta = 2$ and $\beta = 6$, respectively. This is because a larger β implies that it intends to tolerate less instantiation delays and avoid digital twin replacements. It can also be seen from Fig. 8(b) that the impact of the value of β on the running time of Alg.2 is negligible.

We finally investigated the impact of the number $|\mathbb{T}|$ of time slots on the performance of Alg.2. Fig. 9 plots the performance of Alg.2 when there are 20, 30 and 40 time slots, respectively. Fig. 9(a) shows the accumulative utility by Alg.2 with 20 time slots is 56.8% of that by itself with 40 time slots when the network size is 250, while Fig. 9(b) illustrates that Alg.2 with 40 time slots takes the most running time.

VIII. Conclusion

In this paper, we studied digital twin-assisted user satisfaction enhancement in a dynamic MEC environment with highly mobile users and objects. We formulated the static utility maximization problem, and the dynamic utility maximization problem, respectively. We first proposed an ILP solution and then a constant approximation algorithm for the first problem. We then devised an online algorithm with a provable competitive ratio for the second problem. We finally evaluated the performance of the proposed algorithms via simulations. Simulation results demonstrated that the proposed algorithms are promising, improving the algorithm performance by at least 10.7% compared to the baseline algorithms.

ACKNOWLEDGMENT

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

REFERENCES

[1] N. Aschenbruck, R. Ernst, E. Gerhards-Padilla, and M. Schwamborn, "BonnMotion: A mobility scenario generation and analysis tool," in *Proc. 3rd Int. ICST Conf. Simulation Tools Techn.*, 2010, pp. 1–10.

[2] H.-H. Chang, H. Chen, J. Zhang, and L. Liu, "Decentralized deep reinforcement learning meets mobility load balancing," *IEEE/ACM Trans. Netw.*, vol. 31, no. 2, pp. 473–484, Apr. 2023.

[3] 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," in *Proc. IEEE INFOCOM*, May 2022, pp. 770–779.

[4] Q. Chen, S. Guo, W. Xu, Z. Cai, L. Cheng, and H. Gao, "AoI minimization charging at wireless-powered network edge," in *Proc. ICDCS*, Jul. 2022, pp. 713–723.

[5] B. Gao, Z. Zhou, F. Liu, and F. Xu, "Winning at the starting line: Joint network selection and service placement for mobile edge computing," in *Proc. IEEE INFOCOM*, Apr. 2019, pp. 1459–1467.

[6] (2019). *GT-ITM*. [Online]. Available: http://www.cc.gatech.edu/projects/gtitm/

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

[8] Z. Hu et al., "An efficient online computation offloading approach for large-scale mobile edge computing via deep reinforcement learning," *IEEE Trans. Services Comput.*, vol. 15, no. 2, pp. 669–683, Mar. 2022.

[9] M. E. Ildiz, O. T. Yavascan, E. Uysal, and O. T. Kartal, "Query age of information: Optimizing AoI at the right time," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jun. 2022, pp. 144–149.

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

[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. MASS*, Oct. 2022, pp. 311–317.

[13] J. Li et al., "Digital twin-assisted, SFC-enabled service provisioning in mobile edge computing," *IEEE Trans. Mobile Comput.*, early access, Dec. 7, 2022, doi: 10.1109/TMC.2022.3227248.

[14] J. Li et al., "Wait for fresh data? Digital twin empowered IoT services in edge computing," in *Proc. MASS*, 2023, pp. 1–9.

[15] J. Li, W. Liang, M. Chen, and Z. Xu, "Mobility-aware dynamic service placement in D2D-assisted MEC environments," in *Proc. WCNC*, Mar. 2021, pp. 1–6.

[16] J. Li et al., "Budget-aware user satisfaction maximization on service provisioning in mobile edge computing," *IEEE Trans. Mobile Comput.*, early access, Sep. 9, 2022, doi: 10.1109/TMC.2022.3205427.

[17] J. Li, W. Liang, W. Xu, Z. Xu, Y. Li, and X. Jia, "Service home identification of multiple-source IoT applications in edge computing," *IEEE Trans. Services Comput.*, vol. 16, no. 2, pp. 1417–1430, Mar. 2023.

[18] J. Li, W. Liang, Z. Xu, X. Jia, and W. Zhou, "Service provisioning for multi-source IoT applications in mobile edge computing," *ACM Trans. Sensor Netw.*, vol. 18, no. 2, pp. 17:1–17:25, 2021.

[19] J. Li et al., "Maximizing user service satisfaction for delay-sensitive IoT applications in edge computing," *IEEE Trans. Parallel Distrib. Syst.*, vol. 33, no. 5, pp. 1199–1212, May 2022.

[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 INFOCOM*, May 2023, pp. 1–10.

[21] 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.

[22] 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 Things J.*, vol. 9, no. 2, pp. 1427–1444, Jan. 2022.

[23] 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.

[24] J. Martins et al., "ClickOS and the art of network function virtualization," in *Proc. USENIX NSDI*, 2014, pp. 459–473.

[25] E. F. Maleki, L. Mashayekhy, and S. M. Nabavinejad, "Mobility-aware computation offloading in edge computing using machine learning," *IEEE Trans. Mobile Comput.*, vol. 22, no. 1, pp. 328–340, Jan. 2023.

[26] D. Pisinger, "Algorithms for knapsack problems," Citeseer, Princeton, NJ, USA, Tech. Rep. 95/1, 1995.

[27] 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 Trans. Mobile Comput.*, vol. 20, no. 12, pp. 3367–3382, Dec. 2021.

[28] 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.

[29] H. Tout, A. Mourad, N. Kara, and C. Talhi, "Multi-persona mobility: Joint cost-effective and resource-aware mobile-edge computation offloading," *IEEE/ACM Trans. Netw.*, vol. 29, no. 3, pp. 1408–1421, Jun. 2021.

[30] 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.

[31] 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 Trans. Mobile Comput.*, vol. 20, no. 3, pp. 939–951, Mar. 2021.

[32] S. Wang, R. Urgaonkar, M. Zafer, T. He, K. Chan, and K. K. Leung, "Dynamic service migration in mobile edge computing based on Markov decision process," *IEEE/ACM Trans. Netw.*, vol. 27, no. 3, pp. 1272–1288, Jun. 2019.

[33] L. Wang, L. Jiao, J. Li, J. Gedeon, and M. Mühlhäuser, "MOERA: Mobility-agnostic online resource allocation for edge computing," *IEEE Trans. Mobile Comput.*, vol. 18, no. 8, pp. 1843–1856, Aug. 2019.

[34] X. Wang, Z. Ning, S. Guo, M. Wen, and H. V. Poor, "Minimizing the age-of-critical-information: An imitation learning-based scheduling approach under partial observations," *IEEE Trans. Mobile Comput.*, vol. 21, no. 9, pp. 3225–3238, Sep. 2022.

[35] Z. Wang et al., "Mobility digital twin: Concept, architecture, case study, and future challenges," *IEEE Internet Things J.*, vol. 9, no. 18, pp. 17452–17467, Sep. 2022.

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

[37] Z. Xu et al., "Schedule or wait: Age-minimization for IoT big data processing in MEC via online learning," in *Proc. IEEE INFOCOM*, May 2022, pp. 1809–1818.

[38] Z. Xu et al., "Energy-aware collaborative service caching in a 5G-enabled MEC with uncertain payoffs," *IEEE Trans. Commun.*, vol. 70, no. 2, pp. 1058–1071, Feb. 2022.

[39] Z. Xu, W. Gong, Q. Xia, W. Liang, O. F. Rana, and G. Wu, "NFV-enabled IoT service provisioning in mobile edge clouds," *IEEE Trans. Mobile Comput.*, vol. 20, no. 5, pp. 1892–1906, May 2021.

[40] R. D. Yates, Y. Sun, D. R. Brown, S. K. Kaul, E. Modiano, and S. Ulukus, "Age of information: An introduction and survey," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 5, pp. 1183–1210, May 2021.

[41] L. Zhang, C. Wu, Z. Li, C. Guo, M. Chen, and F. C. M. Lau, "Moving big data to the cloud: An online cost-minimizing approach," *IEEE J. Sel. Areas Commun.*, vol. 31, no. 12, pp. 2710–2721, Dec. 2013.

[42] D. Wu, H. Xu, Z. Jiang, W. Yu, X. Wei, and J. Lu, "EdgeLSTM: Towards deep and sequential edge computing for IoT applications," *IEEE/ACM Trans. Netw.*, vol. 29, no. 4, pp. 1895–1908, Aug. 2021.

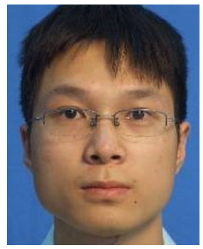
[43] P. Zou, O. Ozel, and S. Subramaniam, "Optimizing information freshness through computation–transmission tradeoff and queue management in edge computing," *IEEE/ACM Trans. Netw.*, vol. 29, no. 2, pp. 949–963, Apr. 2021.

Jing Li received the B.Sc. and Ph.D. degrees (Hons.) from The Australian National University in 2018 and 2022, respectively. He is currently a Post-Doctoral Fellow with The Hong Kong Polytechnic University. His research interests include edge computing, the Internet of Things, digital twin, network function virtualization, and combinatorial optimization.

Song Guo (Fellow, IEEE) is currently 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. His research interests include edge AI, machine learning, mobile computing, and distributed systems. He published many papers in top venues with wide impact in these areas. He is a fellow of the Canadian Academy of Engineering and the IEEE Computer Society. He was a member of the IEEE ComSoc Board of Governors. He has served for the IEEE Computer Society on the Fellow Evaluation Committee. He was a recipient of over a dozen best paper awards from IEEE/ACM conferences, journals, and technical committees. He has also served as the chair for organizing and technical committees of many international conferences. He is the Chair of the IEEE Communications Society (ComSoc) Space and Satellite Communications Technical Committee. He is the Editor-in-Chief of IEEE OPEN JOURNAL OF THE COMPUTER SOCIETY. He has been named on editorial board of a number of prestigious international journals, such as IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, IEEE TRANSACTIONS ON CLOUD COMPUTING, and IEEE TRANSACTIONS ON EMERGING TOPICS IN COMPUTING. He was recognized as a Highly Cited Researcher (Clarivate Web of Science). He was an IEEE ComSoc Distinguished Lecturer.

Weifa Liang (Senior Member, IEEE) received the B.Sc. degree in computer science from Wuhan University, China, in 1984, the M.E. degree in computer science from the University of Science and Technology of China in 1989, and the Ph.D. degree in computer science from The Australian National University in 1998. He is currently 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), the Internet of Things, digital twins, design and analysis of parallel and distributed algorithms, approximation algorithms, and graph theory. He also serves as an Editor for IEEE TRANSACTIONS ON COMMUNICATIONS.

Jianping Wang (Fellow, IEEE) received the B.S. and M.S. degrees in computer science from Nankai University, Tianjin, China, in 1996 and 1999, respectively, and the Ph.D. 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 B.S., master's, and Ph.D. degrees from the School of Computer Science and Technology, Harbin Institute of Technology, China. He was a Post-Doctoral Research Fellow with the Department of Computer Science, Georgia State University. He is currently an Associate Professor with the School of Computers, Guangdong University of Technology. His research interests include routing and scheduling algorithms in wireless networks and sensor networks.

Zichuan Xu (Member, IEEE) received the B.Sc. and M.E. degrees in computer science from the Dalian University of Technology, China, in 2008 and 2011, respectively, and the Ph.D. 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 a Ph.D. 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 B.Sc., M.E., and Ph.D. degrees in computer science from Sun Yat-sen University, Guangzhou, China, in 2008, 2010, and 2015, respectively. He was a Visitor with The Australian National University, Australia, and The Chinese University of Hong Kong, Hong Kong. He is currently an Associate Professor with Sichuan University, China. His research interests include wireless ad hoc and sensor networks, mobile computing, approximation algorithms, combinatorial optimization, online social networks, and graph theory.