Budget-Constrained Digital Twin Synchronization and Its Application on Fidelity-Aware Queries in Edge Computing

Yuchen Li, Weifa Liang, *Senior Member, IEEE*, Zichuan Xu, *Member, IEEE*, Wenzheng Xu, *Member, IEEE*, and Xiaohua Jia, *Fellow, IEEE*

***Abstract*—With the advance of mobile edge computing (MEC) and the Internet of Things (IoT), digital twin (DT) has become an emerging technology for provisioning IoT services between the real world and the cyber world. In this paper, we consider the state updating of DTs in an MEC network through synchronizing DTs with their physical objects. We make use of an energy-constrained UAV for data collection in a sensor network, as an illustrative example for the DT state updating of each object (sensor), and then use the DT data of objects (sensors) later for fidelity-aware query services. To this end, we first formulate a novel DT state staleness minimization, under a given update budget per update round. We then propose an optimal algorithm for a special case of the problem where the budget per update round is exactly K objects synchronizing with their DTs. We then devise an algorithm for the DT state staleness minimization problem by reducing to the award collection maximization problem, assuming that the volume of the update data generated by each object per update round is given. Otherwise, we adopt a deep learning method to predict the volume of the update data. To demonstrate the importance of the DT state staleness in practical applications, we consider fidelity-aware query services in the MEC network, and we develop a cost-effective evaluation plan for each query. We finally evaluate the performance of the proposed algorithms through simulations. Simulation results demonstrate that the proposed algorithms are promising.**

***Index Terms*—Digital twins, DT state staleness, mobile edge computing, DT synchronizations with physical objects, cost modeling, algorithm design and analysis, prediction mechanism.**

Received 18 September 2023; revised 21 August 2024; accepted 3 September 2024. Date of publication 6 September 2024; date of current version 4 December 2024. The work of Weifa Liang was supported in part by Hong Kong Research Grants Council (HK RGC) under CityUHK under Grant 7005845, Grant 8730094, Grant 9043510, and Grant 9380137. The work of Zichuan Xu was supported in part by the National Natural Science Foundation of China under Grant 62172068, and in part by the Shandong Provincial Natural Science Foundation under Grant ZR2023LZH013. The work of Wenzheng Xu was supported in part by NSFC under Grant 62272328, and in part by Sichuan Science and Technology Program under Grant 24NSFJQ0152. Recommended for acceptance by H. Liu. *(Corresponding author: Weifa Liang.)*

Yuchen Li is with the School of Computing, Australian National University, Canberra, ACT 2601, Australia (e-mail: u6013787@anu.edu.au).

Weifa Liang and Xiaohua Jia are with the Department of Computer Science, City University of Hong Kong, Kowloon Tong, Hong Kong (e-mail: weifa.liang@cityu.edu.hk; csjia@cityu.edu.hk).

Zichuan Xu is with the Key Laboratory for Ubiquitous Network and Service Software of Liaoning Province, School of Software, Dalian University of Technology, Dalian 116621, China (e-mail: z.xu@dlut.edu.cn).

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

Digital Object Identifier 10.1109/TMC.2024.3455357

I. Introduction

In THE past decade, digital twin (DT) has gained significant impetus as a breakthrough technological development that has the potential to transform the landscape of manufacturing today and tomorrow [5]. The DT of a physical object refers to a virtual representation of the object in a virtual environment, allowing for real-time monitoring and analysis of its behavior by synchronizing the DT with the object. The DT market is forecast to reach $15.66 billion by 2023 at an annual growth rate of 37.87% according to a market research in 2017 [4], and more organizations and institutions have gained a lot of interest in DTs. The wide adoption of DTs will further be fuelled up by strong service demands from users in the edge of networks, such as Internet of Things (IoT), autonomous driving, healthcare, smart city, AR/VR, and so on. However, the DT study and its applications is still in the early stage, accelerating the usage of DT technologies will bring unprecedented benefits in service provisioning, intelligent resource allocation for 6G wireless networks through real-time digital representation of physical objects [6], [9], [28], [32], [36].

On the other hand, the maturity of mobile edge computing (MEC) network brings computing and storage resources to the edge of core network in the proximity of end users, enabling digital twins to provide services with significant latency reduction. This effectively harnesses the real-time processing characteristic. Therefore, there is a growing interest in DT-enabled service provisioning in an MEC network. The conventional services provided by MEC networks are not comparable with DT-assisted services in MEC including emulating the evolution of physical environments of objects, predicting the behaviours of objects, and helping human decision-making through optimization. DT-assisted service provisioning in MEC now becomes a new frontier. Empowered by the DT technology, the network state information can be efficiently monitored in real time and resource allocation and optimization can be performed timely. Therefore, there urgently needs developing new schemes and algorithms for DT-enabled service provisioning in MEC networks.

As updating DT states incurs cost, the update (synchronization) budget at each update round usually is fixed. This implies that not all DT states of physical objects stored in cloudlets can be updated at each update round. It thus is crucial to determine

1536-1233 © 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: CITY UNIV OF HONG KONG. Downloaded on December 12,2024 at 01:07:45 UTC from IEEE Xplore. Restrictions apply.

the DT states of which physical objects to be updated at each update round in order to minimize the average DT state staleness of all physical objects in the system.

In this paper, we consider a fundamental DT state synchronization problem through an application example of deploying a UAV for remote monitoring in a renewable sensor network, where each sensor is powered by a solar panel and its data generation is determined by its energy. We assume that there is a DT of each sensor deployed in a cloudlet. When the data collected by the sensor is uploaded at one update round, which indicates the DT state of the sensor will be updated at that round. We further assume that the update budget at each update round is to dispatch an energy-limited UAV for data collection from sensors, with the aim to minimize the average DT state staleness of all objects for a finite time horizon, where the finite time horizon can be divided into equal time slots. For the sake of convenience, we assume that each UAV tour can be finished within one time slot. However, the proposed algorithm is still applicable if a UAV tour takes multiple time slots. In our rest discussion, we abuse sensors and objects interchangeably unless otherwise a confusion arises.

To minimize the average DT state staleness of all objects for the given time horizon, there is a non-trivial tradeoff between the DT state freshness of objects and the DT state update cost of objects per update round. The freshness of DT states can be improved by increasing the update budget per update round, e.g., dispatching multiple instead of a single UAV to collect data from sensors. Since the energy capacity on each UAV is limited, a UAV can only collect data from some but not all sensors per tour, this implies that different sensors may synchronize with their DTs at different time points, i.e., the DT state staleness of different sensors are different. Minimizing the average DT state staleness of sensors for a given time horizon subject to the update budget per update round is challenging, as the synchronization cost of the DT of an object (a sensor) is determined not only by the volume of data generated by the object but also by the distances of the object from the rest of other objects whose data to be collected in that UAV tour. In this paper, we will answer the following questions: how long should a UAV hover above a sensor for its data collection? what is the amount of data generated by a sensor since its last data collection? at which time slots should the sensory data of a sensor be collected or synchronized with its DT? how to update the DT states of sensors such that the average staleness of DT states of all sensors is minimized under the given update budget? and what is the relationship between the staleness of DT states and the fidelity of a query result built upon the DT data?

The novelty of this paper lies in studying budget-constrained synchronizations between DTs and their objects under a limited update budget, through a real-world example – a UAV-enabled data collection for a remote sensor network. We explore non-trivial interactions (synchronizations) between physical objects and their DTs in an MEC network to improve the average freshness of DT states of all objects. To the best of our knowledge, this is the first budget-constrained state synchronization framework between DTs and their corresponding objects, and the synchronization algorithms based on the proposed framework has also been proposed through an application example - the use of a UAV for data collection of sensors.

The main contributions of this paper are presented as follows.

- We consider the state synchronization issue between DTs and their physical objects in edge computing environments, and formulate a novel DT state staleness minimization problem over a finite time horizon, under a fixed budget per update round.
- We propose an optimal algorithm for a special DT state staleness minimization problem under a budget assumption of exactly K objects synchronizing with their DTs per update round.
- Motivated by the solution of this special problem, we devise an efficient algorithm for the DT state staleness minimization problem, through an illustrative example - an energy-constrained UAV for data collection, and we reduce the problem to the award collection maximization problem, assuming that the volume of the update data of each object since its last synchronization is given. Otherwise, we develop a deep learning mechanism to predict the volume of the update data of each object through analyzing the historical traces of update data based on the DT data of the object.
- To demonstrate the importance of reducing the staleness of DT states, we consider fidelity-aware query services in the DT-empowered MEC network, and develop a cost-effective evaluation algorithm for such a query evaluation.
- We evaluate the performance of the proposed algorithms through simulations. Simulation results demonstrate that the proposed algorithms are promising. Particularly, the average DT state staleness by the proposed algorithms are 44.92% less than the baselines, and the query evaluation cost is 19.42% less than the cost delivered by the baseline algorithm.

The rest of the paper is organized as follows. Section II summarizes the related work on DTs in an MEC network. Section III introduces notions, notations, and problem definitions. Section IV provides an optimal solution to a special DT state staleness minimization problem, and Section V devises an efficient algorithm for the problem. Built upon the DT technology, Section VI deals with fidelity-aware query services on DT information, by developing a cost-effective evaluation plan for each service query. Section VII evaluates the performance of the proposed algorithms, and Section VIII concludes the paper.

II. Related Work

MEC has emerged as a promising paradigm for delay-sensitive IoT applications [16], [17], [30]. Empowered by the DT technology, MEC platforms can provide enhanced services for various delay-aware IoT and AI applications [6], [9], [24], [26],

Authorized licensed use limited to: CITY UNIV OF HONG KONG. Downloaded on December 12,2024 at 01:07:45 UTC from IEEE Xplore. Restrictions apply.

[28]. For example, Dong et al. [6] focused on a deep learning-based model training by mapping an MEC network to its DT network, and proposed an efficient algorithm for the DT network training, with the aim to minimize the training energy consumption. Fan et al. [7] proposed a DT-empowered edge AI framework to control vehicles with different automation levels, by adopting deep reinforcement learning method. Lin et al. [24] devised an incentive-based congestion control scheme to meet dynamic service demands of DTs in an MEC network, by utilizing the Lyapunov optimization technology. Lu et al. [26] developed a federated learning algorithm based on the blockchain technique to enhance data privacy and security in DT-assisted MEC networks. Li et al. [12], [13] considered delay-aware IoT services in a DT-enabled MEC network and developed approximation and online algorithms, by exploring fine trades-off between the accuracy of DT services and service response delays. Zhang et al. [42] focused on mobility-aware delay-sensitive service provisioning in DT-empowered MEC networks by formulating static and dynamic DT replica placement problems, and they proposed efficient solutions to the problems. Zhang et al. [41] studied the DT migration problem in an MEC network while taking the mobility of objects into consideration, and developed a DRL algorithm for the problem to minimize the total cost of various resources. Zhang et al. [39] studied the DT-assisted Federated Learning (FL) in MEC platforms. They proposed a multi-FL service framework where the utility of FL services is maximized by utilizing DT models to optimize the scheduling of mobile devices and allocations of communication resources. Zhang et al. [44] studied both DT placements and migrations in MEC networks with the mobility assumption of both objects and users, and devised efficient algorithms to minimize the cost of DT update and user services.

There are studies focusing on optimizing service delays or data freshness in a DT-enabled MEC network [3], [14], [15], [18], [23], [27], [32], [34]. For example, Corneo et al. [3] studied the problem of timely dissemination of sensor updates to clouds that provide DT service for users with the aim to minimize the age of information (AoI). They devised a novel push-and-pull method of sensor information updating to strive for a non-trivial tradeoff between the sensory data freshness and the tolerable query delay. Sun et al. [32] utilized DTs to minimize the offloading latency by leveraging the Lyapunov optimization technique. Lu et al. [27] devised a deep learning-based algorithm to cope with the mobility of mobile devices and migration of DTs to minimize the average service delay. The study of this paper is closely related to the AoI that is to minimize the average duration of information from their generation to their usage in diverse applications [2], [25], [35], [37], [40]. Since the AoI is very sensitive to the transmission and processing delays of information since the generation of the information, most existing studies on AoI are conducted in MEC platforms, due to the MEC networks closer to the sources (sensory devices) of information generation. For example, Liu et al. [25] studied the AoT problem by developing an approximation algorithm for finding multiple routing paths between a pair of source and destination nodes in a network and showed that the problem is equivalent to minimizing the maximum-delay routing path among the routing paths. Bastopcu and Ulukus [2] considered an information update system where a receiver requests updates from an information provider to minimize the AoI, and the updates are generated at the information provider by completing a set of collecting data and performing computation tasks, subject to a minimum quality (maximum allowed distortion) constraint on the updates. They developed an efficient policy to achieve their optimization objective. Wang et al. [35] devised an offline scheduling algorithm and an online learning algorithm for minimizing the average Age of Critical Information (AoCI) of mobile clients. Xu et al. [37] dealt with the AoI minimization for IoT Big Data applications, and devised an online learning method for dynamic request admissions. Zhang et al. [40] proposed an efficient algorithm to minimize the average service delay while meeting content freshness requirements, by striving for non-trivial trades-off between the AoI and the service delay for timely content refreshing. Zhang et al. [43] investigated AoI-aware inference service provisioning in an MEC network by admitting Digital Twin Network (DTN) slicing requests for sharing limited communication and computing resources in MEC and proposed a framework for DT and service model placements with guaranteed service accuracy. They also devised efficient heuristics for DTN slicing admissions while meeting AoI and accuracy requirements of users. Vaezi et al. [33] investigated minimizing data request-response delays of digital twin (DT) applications, subject to a maximum data age target constraints at DTs and an application server. Li et al. [11], [19] explored dynamic placements of digital twins in MEC environments to enhance user service satisfaction by proposing algorithms that consider the mobility of objects by reducing the AoI of query services. Liang et al. [22] addresses maximizing the state freshness of digital twins and their inference service models in MEC networks and the updated data are then used for training the service models. Li et al. [18] considered how to maximize the fidelity gain of service models while minimizing the total cost of various resource consumed of DT synchronizations with their objects. Efficient algorithms are then devised to jointly optimize the placements of digital twins and service models. Although there are some similarities between the average AoI of data generated by physical objects and the average DT state staleness of physical objects, the DT technology is more powerful and complicated, which not only minimizes the average AoI of the DT states of physical objects but also is able to utilize the DT states of physical objects to activate or predict the behaviours of the physical objects in future.

Most of the aforementioned studies on DT-assisted IoT services in MEC networks focused on allocating limited MEC resources to optimize the AoI of services. We here instead consider the update budget on synchronizations between DTs and their physical objects. To the best of our knowledge, we are the very first to identify a fundamental problem - the budget-constrained DT state synchronization problem in MEC networks, by proposing a novel framework and algorithms. We also propose effective evolution plans for fidelity-aware query services through improving the freshness of DT states, using a real-world example of a UAV-enabled data collection in a renewable sensor network.

Authorized licensed use limited to: CITY UNIV OF HONG KONG. Downloaded on December 12,2024 at 01:07:45 UTC from IEEE Xplore. Restrictions apply.

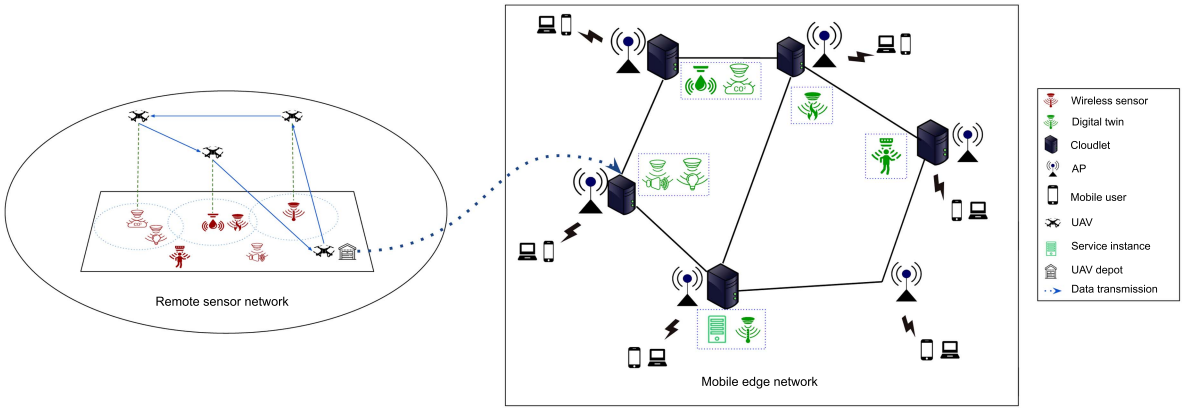


Fig. 1. An illustrative example of an MEC network and DT placements in cloudlets.

III. Preliminaries

In this section we first introduce the system model, notions and notations. We then define problems and show NP-hardness of the defined problems.

A. System Model

Given an MEC network $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, $\mathcal{N}$ is the set of Access Points (APs) and $\mathcal{E}$ is the set of links between APs. With each AP, there is a co-located cloudlet that is connected with the AP through a high speed optical fiber cable, and the communication delay between the AP and its co-located cloudlet can be ignored, and the AP and its co-located cloudlet can be used interchangeably if no confusion arises. Users can make use of query services provided by the MEC service provider through issuing fidelity-aware queries on the DT data. The DTs of different physical objects are deployed in cloudlets, and the DT states of physical objects can be synchronized (updated) with their physical objects at different update rounds.

In this paper, we study a general setting of DT state synchronizations with their physical objects through a real application scenario, where there is a renewable sensor network deployed in a remote region. Let V be the set of sensors in the sensor network. Each sensor $v_i \in V$ has a digital twin DT_i in a cloudlet in an MEC network $\mathcal{G}$, which is a virtual mirror or modeling of the sensor in cyber space, which contains all sensory data collected from sensor v_i in the past, including the volume of data collected at previous update time slots. We term all collected data related to the DT of a sensor to its last update as the *DT state* of the sensor. We further assume that DT_i of sensor v_i stores all information related to sensor v_i so far. Also, the DT_i can emulate or predict the behaviours of sensor v_i such as its data generation since last data collection. To ensure that DT_i can simulate the behaviour of v_i accurately, the state of DT_i needs to be synchronized with v_i quite often, and any sensory data update of v_i must be sent to DT_i accordingly.

An illustrative example of the proposed network is given in Fig. 1, where the left side figure is a wireless sensor network consisting of seven sensors, the right side figure is the MEC network, and the DTs of sensors are placed in cloudlets with the green color. Also, the UAV will upload its collected data per tour at depot to the MEC network, and DTs of the sensors uploading their update data will be updated accordingly.

B. Staleness of DT States

Let j_1 be the last time slot that DT_i synchronized with object $v_i \in V$, and let j_2 be its next synchronization time slot with $j_2 > j_1$. The update data generated by v_i at time slot t' ($j_1 \leq t' \leq j_2$) will be synchronized with DT_i at time slot j_2. Therefore, the state staleness length of DT_i at time slot t is $t - t'$ with $t' \leq t \leq j_2$, similar to the definition of the AoI [37]. The accumulative staleness $d_{v_i}(t)$ of the state of DT_i at time slot t then is

$$d_{v_i}(t) = \sum_{t'=j_1}^{t} (t - t') = \frac{(t - j_1)(t - j_1 + 1)}{2} \tag{1}$$

Let $D(t)$ be the total staleness of DT states of all objects at time slot t, then

$$D(t) = \sum_{v_i \in V} d_{v_i}(t), \tag{2}$$

where $d_{v_i}(t)$ is the accumulative number of time slots elapsed from the last synchronization of DT_i with object v_i to time slot t.

Let $l_{v_i}(t)$ be the number of time slots elapsed of DT_i of object v_i from its last state synchronization to time slot t, (2) can be rewritten as follows.

$$D(t) = \sum_{v_i \in V} d_{v_i}(t) = \sum_{v_i \in V} \frac{l_{v_i}(t) \cdot (l_{v_i}(t) + 1)}{2} \tag{3}$$

Let $F(t)$ be the accumulative staleness of DT states of all objects from time slot one to time slot t, then

$$F(t) = \sum_{t'=1}^{t} D(t') = \sum_{t'=1}^{t} \sum_{i=1}^{|V|} \frac{l_{v_i}(t') \cdot (l_{v_i}(t') + 1)}{2} \tag{4}$$

It can be seen that function $F(t)$ is a strictly monotonic increasing function due to the fact that $F(t+1) - F(t) = D(t+1) \geq 0$.

Authorized licensed use limited to: CITY UNIV OF HONG KONG. Downloaded on December 12,2024 at 01:07:45 UTC from IEEE Xplore. Restrictions apply.

For a given monitoring period T, to minimize the average staleness of DT states of all objects in V at each time slot, the optimization objective is to

$$minimize \quad \frac{F(T)}{T \cdot |V|} \tag{5}$$

Definition 1: Given an MEC network $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, a set V of objects, a digital twin DT_i of each object v_i that is deployed in a cloudlet of $\mathcal{G}$, a given budget per update round (a UAV with limited energy capacity is used to collect data from some sensors), assume the update data of each object $v_i \in V$ since its last uploading is stored at itself until to be uploaded in some future time slot, and the update data of each object will be uploaded to its DT and the DT state of the object is updated accordingly. Given a finite time horizon T, *the DT state staleness minimization problem* in $\mathcal{G}$ is to minimize the average DT state staleness $\frac{F(T)}{|V|T}$ of all objects in V that is defined in (5) for time horizon T, subject to the update budget per time slot.

C. The Fidelity-Aware Query Evaluation Problem

Given an MEC network $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ and a set V of objects, assume that each object $v_i \in V$ has a DT_i placed in a cloudlet of $\mathcal{G}$. Built upon the DT data, we deal with fidelity-aware query services provided by the service provider of $\mathcal{G}$. Since different objects have different DT states, the freshness on the accumulative update data at their DTs are different too. Queries on these DT data at different time slots will result in differential fidelity of the query results. That is, some query results may have high fidelity, while others may have low fidelity due to lack of synchronizations between DTs and their objects for a while. We here make use of the DT state information of objects to help fidelity-aware query services in the MEC network.

Specifically, assuming that there is a query on a subset of objects $V' \subseteq V$ with specific requirements, such as the data staleness on most DTs of objects in V' is no greater than δ time slots. For example, a user u issues a query $Q(t)$ to an AP co-located with cloudlet n_u at time slot t, which is expressed by a tuple $Q(t) = (V', \delta, \vartheta)$, where the update data from object $v_i \in V'$ at DT_i is *fresh* if its last synchronization with object v_i is no greater than δ time slots; otherwise, the data stored at DT_i is not *fresh* by the requirement of query $Q(t)$. The set V' of objects specified by query $Q(t)$ can then be partitioned into two disjoint subsets $V'_{>\delta} = \{v_i \mid v_i \in V', l_{v_i}(t) > \delta\}$ and $V'_{\leq\delta} = \{v_i \mid v_i \in V', l_{v_i}(t) \leq \delta\}$, where $l_{v_i}(t)$ is the staleness length of DT_i from its last synchronization with object v_i to time slot t. The value $|V'_{\leq\delta}|/|V'|$ is referred to as *the ratio of data freshness of the query result*, or *the fidelity* of a query result. An evaluation plan for query $Q(t)$ is *feasible* if the ratio of data freshness of the query result is no less than its pre-defined threshold ϑ, i.e., $|V'_{\leq\delta}|/|V'| \geq \vartheta$.

As the DTs of different objects in V' are placed at different cloudlets, the update data of these objects stored at their DTs will be aggregated for various query services. Assume that the DT data of different DTs in the same cloudlet will be aggregated prior to sending their data to the *source cloudlet* $n_u \in \mathcal{N}$ of query $Q(t)$, we term this aggregated data of these DTs as a single 'virtual' DT in the cloudlet, i.e., if a cloudlet contains multiple DTs of objects in V' for a query, it only sends the aggregation result of the DT data to the source cloudlet once, while the *source cloudlet* of a query is the cloudlet (the co-located AP) from which the query is uploaded to the MEC network.

To find a cost-effective query evaluation plan for a query $Q(t)$ issued by a user u under the coverage of AP (or source cloudlet) $n_u \in \mathcal{N}$ at time slot t, it needs to find an evaluation tree T_u in $\mathcal{G}(\mathcal{N}, \mathcal{E})$ rooted at cloudlet n_u and spanning cloudlets containing the DTs of objects in V' such that the evaluation cost $C(Q(t))$ is minimized, and $C(Q(t))$ is defined as follows

$$C(Q(t)) = \sum_{(x,y)\in E(T_u)} \rho(x,y) D^{(T_u)}(x,y) + \sum_{v \in V(T_u)} agg^{(T_u)}(v), \tag{6}$$

where $\rho(x, y)$ is the transmission cost per unit data along link (x, y) in $\mathcal{G}$ between cloudlets x and y, $D^{(T_u)}(x, y)$ is the volume of data transferred on link (x, y) in the evaluation tree T_u, $agg^{(T_u)}(v)$ is the processing cost of aggregating data from the child nodes of node v in tree T_u, and $V(T_u)$ is the set of nodes and $E(T_u)$ is the set of edges in T_u, respectively. The cloudlets in tree T_u participating in query evaluation of $Q(t)$ consume their computing resources. Assume that the tree node of cloudlet v in T_u contains l children: $u_1, u_2, \ldots, u_l$. The processing cost $agg^{(T_u)}(v)$ at cloudlet v in tree T_u is defined as follows.

$$agg^{(T_u)}(v) = \xi \cdot g(D^{(T_u)}(u_1, v), D^{(T_u)}(u_2, v), \ldots, D^{(T_u)}(u_l, v)), \tag{7}$$

where ξ is the processing cost per unit data at a cloudlet, $D^{(T_u)}(u_{l'}, v)$ is the volume of the aggregate data from subtree $T_{u_{l'}}$ of T_u rooted at cloudlet $u_{l'}$ with $1 \leq l' \leq l$. If cloudlet v is a tree leaf, $agg^{(T_u)}(v) = \sum_{v_i \in V'} \{D_{v_i} \mid DT_i \text{ ofobject } v_i \text{isplacedatclouldletv}\}$; otherwise $agg^{(T_u)}(v)$ is defined in (7), and $g(\cdot, \cdot, \ldots, \cdot)$ is an aggregate function that is determined by query $Q(t)$, and the volume of the aggregate result at a node usually is no more than the accumulative volume of data from its children in T_u.

Definition 2: Given the DTs of objects in V deployed in cloudlets of an MEC network $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, a user u issues a fidelity-aware query $Q(t) = (V', \delta, \vartheta)$ from an AP $n_u \in \mathcal{N}$ with $V' \subseteq V$ at time slot t with $1 \leq t \leq T$, *the fidelity-aware query evaluation problem* of $Q(t)$ in $\mathcal{G}$ is to determine whether there is a feasible evaluation plan for $Q(t)$ and find a feasible evaluation plan for such that the evaluation cost $C(Q(t))$ in (6) is minimized if such a plan does exist.

D. A Real Application Scenario

In this paper, we make use of a real-world application example to illustrate the DT state staleness concept, the DT state synchronization framework and algorithms, and the use of DT data for fidelity-aware query services in an MEC network $\mathcal{G}$.

We assume that there is a remote sensor network for monitoring purpose. Let V be the set of sensors, and there is a DT_i

Authorized licensed use limited to: CITY UNIV OF HONG KONG. Downloaded on December 12,2024 at 01:07:45 UTC from IEEE Xplore. Restrictions apply.

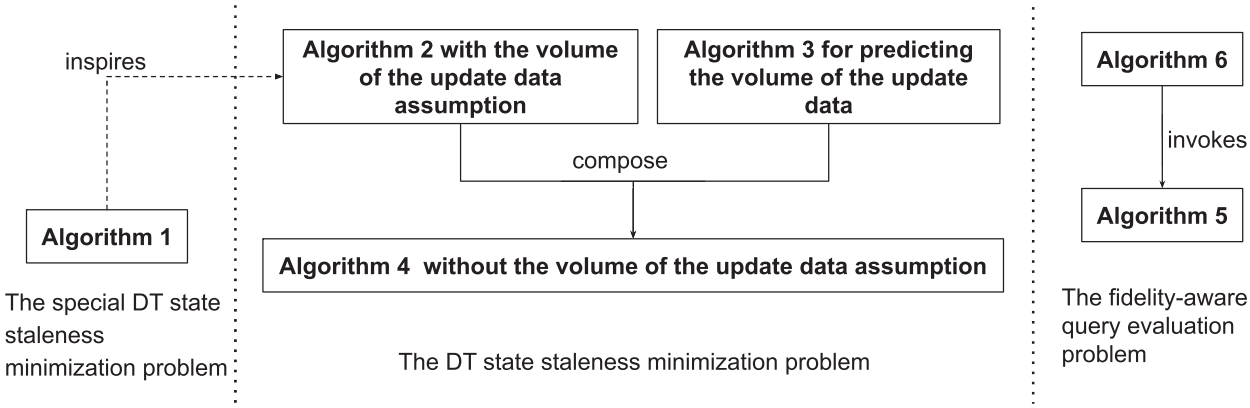


Fig. 2. The relationships among the six proposed algorithms in this paper.

deployed in a cloudlet of $\mathcal{G}$ for each sensor $v_i \in V$. In the rest of discussions, each sensor is an object, and sensors and objects can be used interchangeably. The update budget per update round is the energy capacity of an energy-limited UAV for data collection. This implies that not every sensor's sensory data can be collected by the UAV at each update round (or each time slot) due to its limited energy capacity $\mathcal{E}_{UAV}$. The UAV consumes energy on hovering for collecting sensory data from sensors and traveling above different sensors. Its total energy consumption per update round is no greater than $\mathcal{E}_{UAV}$. For the sake of convenience, we assume that the UAV travels at a constant speed ν, and let η_1 and η_2 be its energy consumption costs on hovering and traveling per unit time, respectively.

Let $\mathcal{T}_h$ and $\mathcal{T}_t$ be the amounts of time that the UAV spends on hovering and traveling at a tour, respectively, then we must have

$$\eta_1 * \mathcal{T}_h + \eta_2 * \mathcal{T}_t \leq \mathcal{E}_{UAV}. \tag{8}$$

A sensor whose data is collected at an update round will be uploaded to its DT in $\mathcal{G}$ in the end of that update round, and its DT state will also be updated at that update round accordingly.

E. NP-Hardness of the Defined Problems

Theorem 1: The DT state staleness minimization problem in an MEC network $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ is NP-hard.

Proof: We consider a special case of the problem when $T = 1$, in which case the problem is to synchronize as many objects as possible with their DTs, by collecting their sensory data in one tour of the UAV (the update budget). Since the energy capacity of the UAV is limited, the amount of energy consumed by the UAV at a sensor is determined by the volume of data generated by the sensor since its last collection (or the synchronization with its DT in the last round). To this end, it is to find a closed tour for the UAV such that as many sensors as possible are included in the tour, subject to the energy capacity on the UAV, assuming that the UAV is in a depot s initially and will return to the depot after the tour. It can be seen that the travelling salesman problem (TSP) can be reduced to this special problem. It is well known that the TSP is a classic NP-hard problem, the DT state staleness minimization problem is NP-hard too. □

Theorem 2: The fidelity-aware query evaluation problem in an MEC network $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ is NP-hard.

Proof: We consider a special case of the fidelity-aware query evaluation problem where a query $Q(t) = (V', \delta, \vartheta)$ has the data volume of the DT of every sensor in V' of 1, and the DTs of different sensors are deployed at different cloudlets. We then construct a multicast tree for the query rooted at the source cloudlet of the query such that its evaluation cost is minimized. This is equivalent to constructing a Steiner tree in an MEC network $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ spanning the source cloudlet and all other cloudlets that contain the DTs of sensors in V'. Since the Steiner tree problem is a well known NP-hard problem, the fidelity-aware query evaluation problem is NP-hard, too. □

In the rest of this paper, we develop six algorithms to tackle the defined problems. The relationships among the six algorithms are illustrated by Fig. 2, where Algorithm 1 is for a special case of the DT state staleness minimization problem. Inspired by the solution of Algorithm 1, Algorithm 2 is developed for the DT state staleness minimization problem, assuming that the amount of sensory data of each sensor at each time slot is given. However, the volume of sensory data of each sensor is not known, Algorithm 3 is designed to predict the amount of data generated by each sensor at each time slot, and Algorithm 4 is devised for the DT state staleness minimization problem with the assumption that the volume of sensory data generated by each sensor is given. In addition to dealing with DT state freshness, one potential application of DT state freshness - the fidelity-aware query evaluation problem is considered, for which Algorithm 5 finds a cost-effective evaluation plan for a query. To examine whether the query evaluation plan meets the fidelity requirement of its user, Algorithm 6 examines the feasibility of the evaluation plan by invoking Algorithm 5.

For the sake of convenience, we list all symbols adopted in this paper in Table I.

IV. Optimal Algorithm for a Special DT State Staleness Minimization Problem

Since the synchronization budget (a UAV tour for data collection) at each time slot is given, this implies that not all DT

Authorized licensed use limited to: CITY UNIV OF HONG KONG. Downloaded on December 12,2024 at 01:07:45 UTC from IEEE Xplore. Restrictions apply.

TABLE I
TABLE OF SYMBOLS

Notations	Descriptions
$\mathcal{G} = (\mathcal{N}, \mathcal{E})$	an MEC network with a set $\mathcal{N}$ of APs and cloudlets, and a set of $\mathcal{E}$ links
V and v_i	the set of sensors in the sensor network and a sensor $v_i \in V$
DT_i	the digital twin of sensor v_i
t and T	the time slot index of monitoring period and the maximum time slot
$d_{v_i}(t)$	the accumulative staleness of the state of DT_i at time slot t
$D(t)$	the total staleness of DT states of all objects at time slot t
$l_{v_i}(t)$	the number of time slots elapsed of DT_i of object v_i from its last state synchronization to time slot t
$F(t)$	the accumulative staleness of DT states of all objects from time slot one to time slot t
u and n_u	a user and the AP he issues a query to
$Q(t) = (V', \delta, \vartheta)$	$Q(t)$ is a query issued at time slot t; V' is the subset of queried objects; δ is the freshness threshold; ϑ the threshold of fidelity
$V'_{\leq\delta}$	the subset of fresh queried objects
$\lvert V'_{\leq\delta}\rvert/\lvert V'\rvert$	the fidelity of a query result
T_u	an evaluation tree rooted at cloudlet n_u
$C(Q(t))$	the evaluation cost of query $Q(t)$ issued at time slot t.
$\rho(x, y)$	transmission cost per unit data along link between cloudlets x and y
$D^{(T_u)}(x, y)$	the volume of data transferred on link (x, y) in the evaluation tree T_u
ξ	the processing cost per unit data at a cloudlet
$agg^{(T_u)}(v)$	the processing cost of aggregating data from the child nodes of node v in tree T_u
$V(T_u)$ and $E(T_u)$	the set of nodes and the set of edges in T_u
$u_1, u_2, \ldots, u_l$	the l children of a tree node in T_u
$g(\cdot, \cdot, \ldots, \cdot)$	an aggregate function
$\mathcal{E}_{UAV}$ and ν	the energy capacity and the flying speed of the UAV.
η_1 and η_2	the hovering energy consumption and traveling energy consumption per unit time of the UAV.

states of sensors can be synchronized at each time slot. Then, it is challenging to identify which sensors to be included in the UAV tour to minimize the average DT staleness of all sensors. We address the challenge by considering a special case of the problem where exactly K sensors can synchronize with their DTs at each time slot t. That is, each UAV tour can choose K sensors for their data collection at each time slot t. Without loss of generality, let $v_{i_1}, v_{i_2}, \ldots, v_{i_{|V|}}$ be the sequence of sorted sensors in non-increasing order of their state staleness length at time slot t, with $l_{v_{i_1}}(t) \geq l_{v_{i_2}}(t) \geq \cdots \geq l_{v_{i_{|V|}}}(t)$. To minimize the objective function $\frac{F(T)}{T \cdot |V|}$ defined in formula (5), we choose to update the DT states of the first K sorted sensors per time slot t. We refer to this algorithm as Algorithm 1.

We claim that this will give an optimal solution to the special DT state staleness minimization problem.

Theorem 3: Assuming that the DTs of sensors are stored in cloudlets in an MEC network $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, the update budget is that the DT states of the first K sorted sensors are synchronized with their sensors at each time slot t with $1 \leq t \leq T$. Let $S(t)$ be the top-K sensors with the longest staleness length at time slot t delivered by the proposed algorithm, the solution $\sum_{t=1}^{T} S(t)$ delivered by Algorithm 1 for the special DT state staleness minimization problem is optimal, where $K \leq |V|$.

Algorithm 1: An Optimal Algorithm for the Special DT State Staleness Minimization Problem.

Input: A set V of sensors with each sensor $v_i \in V$ having a DT_i in $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, a UAV with energy capacity located at depot s, and K DTs of sensors can be synchronized at each time slot t with $1 \leq t \leq T$.

Output: A schedule $S(t)$ of DT state synchronizations between sensors and DTs at each time slot t such that $F(t)$ is minimized.

1: $S(t) \leftarrow \emptyset$; /* the solution at time slot t */
2: Calculate the staleness length $l_{v_i}(t)$ of DT_i of each sensor $v_i \in V$;
3: Sort sensors in non-increasing order of their staleness length;
4: **return** Set $S(t)$ contains the first K sorted sensors at time slot t.

Proof: We first show that the solution $S(t)$ delivered by the proposed algorithm at each time slot t is optimal by induction, where $K \leq |V|$. It can be seen that at time slot $t = 1$, the claim is true. Assume that the claim is true for the first $t - 1$ time slots, we now show that the claim is also true at time slot t by

Authorized licensed use limited to: CITY UNIV OF HONG KONG. Downloaded on December 12,2024 at 01:07:45 UTC from IEEE Xplore. Restrictions apply.

contradiction. Recall that the sorted sensor sequence is $v_1, v_2, \ldots, v_{|V|}$ in non-increasing order of their DT state staleness length. Assume that the DT state updates of sensors in the optimal solution $S'(t)$ at time slot t does not contain all the first K sorted sensors, let v_p be one of such a sensor with $p > K$, and let v_k be the first one among the K sorted sensors that is not in the optimal solution with $1 \leq k \leq K$. Following the definition of function l, we have $l_p(t) \leq l_k(t)$, another solution is obtained by replacing sensor v_p in the optimal solution with sensor v_k. The value difference between the new solution $S'(t) \cup \{v_k\} \setminus \{v_p\}$ and the optimal solution $S'(t)$ is $\frac{l_p(t)\cdot(l_p(t)+1)}{2} - \frac{l_k(t)\cdot(l_k(t)+1)}{2} \leq 0$, as $l_k(t) \geq l_p(t)$ by the assumption. Thus, we can replace each sorted sensor in the optimal solution whose index is not in $\{1, 2, \ldots, K\}$ by another sorted sensor whose index is in that range until all K sensors in the final solution are the first K sorted sensors. It can be seen that the final solution has the minimum value. This contradicts that solution $S'(t)$ is optimal.

Let $S(t)$ be the solution delivered by Algorithm 1 at time slot t. We then show that the solution $\cup_{t=1}^{T} S(t)$ is optimal by contradiction. Assume that the optimal solution to this special problem is $\cup_{t=1}^{T} S'(t)$, let $l(S'(t))$ be the sum of the DT staleness lengths of objects in the optimal solution at time slot t. We have shown that $l(S(t)) \leq l(S'(t))$. Thus, $\sum_{t=1}^{T} l(S(t)) \leq \sum_{t=1}^{T} l(S'(t))$. We claim that $\sum_{t=1}^{T} l(S(t)) = \sum_{t=1}^{T} l(S'(t))$. Otherwise, this contradicts the assumption that $\cup_{t=1}^{T} S'(t)$ is the optimal solution. The theorem then follows. □

V. Algorithm for the DT State Staleness Minimization Problem

In this section, we deal with the DT state staleness minimization problem. We first develop an efficient approximation algorithm for a sub-problem of the problem. That is, given the update budget per update round (per time slot), we find a closed tour for the UAV such that the optimization objective $F(t)$ defined in (4) is minimized if the volume of data generated by each sensor since its last data collection is given; otherwise we develop a deep learning mechanism to predict the volume of data generated by the sensor based on its accumulative DT data in its DT. We then devise an efficient algorithm for the DT state staleness minimization problem.

A. Approximation Algorithm With the Volume of Update Data of Each Object Being Given

We consider a subproblem to minimize $F(t)$, defined in (4), at time slot t through finding a close tour for the UAV such that as many sensors with longer DT state staleness length as possible are included in the tour, i.e., the DT states of the chosen sensors will be updated at time slot t. The basic idea is to reduce this subproblem to the award collection maximization problem in an auxiliary graph $G(t)$. A closed tour in $G(t)$ subject to the tour length is found, which derives a feasible solution for the subproblem. Before we proceed, we define *the orienteering problem* and *the award collection maximization problem* as follows.

Given an undirected graph $G = (V, E; \pi, c)$ with $\pi : V \mapsto \mathbb{Z}^+$ and $c : E \mapsto \mathbb{Z}^+$, a source node $s \in V$, a destination $t \in V$, and a non-negative integer $L > 0$, *the orienteering problem* in G is to find a path $P_{s,t}$ from node s to node t such that the total reward $\sum_{v \in P_{s,t}} \pi(v)$ collected from the nodes in $P_{s,t}$ is maximized, while the length $\sum_{e \in P_{s,t}} c(e)$ of path $P_{s,t}$ is no greater than L, i.e., $\sum_{e \in P_{s,t}} c(e) \leq L$. The mentioned orienteering problem is NP-hard, and there is a 3-approximation algorithm for it if the edge weights in G abide by the triangle inequality, due to Bansal et al. [1].

Given the metric graph $G = (V, E; \pi, c)$, a non-negative value $\mathcal{E}_{UAV}$, and a specific node $s \in V$, *the award collection maximization problem* in G is to find a closed tour including node s such that the total award collected from the nodes in the closed tour is maximized, subject to that the tour length is upper bounded by $\mathcal{E}_{UAV}$.

Having the above preparations, we now construct an auxiliary graph $G(t) = (V \cup \{s\}, E(t); c(\cdot,\cdot), \pi(\cdot))$ for the optimization subproblem with the optimization objective defined in (4) as follows.

The edge cost function $c : E(t) \mapsto \mathbb{R}^{\geq 0}$ and the node award function $\pi : V \cup \{s\} \mapsto \mathbb{Z}^{\geq 0}$ at each time slot t, node s is the depot of the UAV. There is an edge in $E(t)$ between any pair of nodes in $V \cup \{s\}$. For each edge $(u, v) \in E(t)$, its weight $c(u, v)$ is defined as follows.

$$c(u,v) = \eta_1 \cdot \frac{D_u(t) + D_v(t)}{2B} + \eta_2 \cdot \frac{dist(u,v)}{\nu}, \quad (9)$$

where $D_u(t)$ and $D_v(t)$ are the volumes of data generated by sensors u and v since their last data collection, respectively, and $D_s(t) = 0$ of depot s. B is the radio bandwidth of the UAV, ν is the traveling speed of the UAV, $dist(u, v)$ is the euclidean distance between two sensors u and v, and η_1 and η_2 are the amounts of energy consumed of the UAV on hovering and travelling per time unit. Note that cost $c(u, v)$ is the sum of energy consumption costs of the UAV at hovering on sensors u and v and travelling between them. Since edge (u, v) is contained in the closed tour, let (u', u) and (v', v) be the edges incident to nodes u and v in the closed tour respectively, then the energy consumption of the UAV for collecting data from sensor u is distributed to both edges (u', u) and (u, v) with each a half the amount of energy, and the energy consumption of the UAV for collecting data from sensor v is distributed to both edges (v', v) and (u, v) with each a half as well.

We aim to find a closed tour in $G(t)$ that includes depot s and as many sensors as possible, as each sensor in the tour will synchronize with its DT after the UAV finished the closed tour at time slot t. Also, we expect the first K sorted sensors to be included in the closed tour to minimize the average DT state staleness of all sensors in the next update round. Thus, a sensor with a longer DT state staleness length should be assigned a larger award, i.e., each sensor node v_i in $G(t)$ is assigned the amount of award $\pi(v_i)$, which is the accumulative staleness of DT_i since its last data collection until time slot t that is defined as follows.

$$\pi(v_i) = \frac{l_{v_i}(t) \cdot (l_{v_i}(t) + 1)}{2}. \quad (10)$$

Authorized licensed use limited to: CITY UNIV OF HONG KONG. Downloaded on December 12,2024 at 01:07:45 UTC from IEEE Xplore. Restrictions apply.

Algorithm 2: An Approximation Algorithm for Minimizing $F(t)$ With the Given Data Volume Assumption at Time Slot t.

Input: A set V of sensors with each sensor $v_i \in V$ having a DT_i in an MEC network $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, and a UAV with energy capacity $\mathcal{E}_{UAV}$ located at depot s, the volume $D_{v_i}(t)$ of data generated by each sensor $v_i \in V$ from its last data collection to time slot t is given with $1 \leq t \leq T$.

Output: A schedule of DT state synchronizations between sensors and their DTs at time slot t such that $F(t)$ is minimized with $1 \leq t \leq T$.

1: $S(t) \leftarrow \emptyset$; /* the solution at time slot t */
2: Calculate the staleness length of DT_i of each sensor $v_i \in V$;
3: Construct an auxiliary graph $G(t) = (V \cup \{s\}, E'(t); c(\cdot,\cdot), \pi(\cdot))$, where the weight $c(u, v)$ of edge (u, v) is the energy consumption of the UAV for hovering at sensors u and v and travelling between them that is defined in (9), and $\pi(v_i)$ $(= \frac{l_{v_i}(t)(l_{v_i}(t)+1)}{2})$ is the award of node $v_i \in V$ that is the accumulative staleness length of DT_i from its last synchronization to time slot t;
4: Find a closed tour $\mathcal{C}(t)$ in graph $G(t)$ including depot s with maximizing the total award collected from sensors in the tour, by applying the approximation algorithm for the award collection maximization problem due to Liang et al. [21];
5: Collect data from sensors in $\mathcal{C}(t)$ by the UAV, and synchronize the DT state of each sensor in $\mathcal{C}(t)$ after the UAV uploaded all collected data at depot s;
6: **return** The synchronization schedule $S(t) \leftarrow \mathcal{C}(t)$ at time slot t.

It can be seen that $\pi(v_i) = d_{v_i}(t)$, where $d_{v_i}(t)$ is the accumulative number of time slots elapsed of sensor v_i from its last data collection time slot to time slot t that is defined in (1), and $l_{v_i}(t)$ is the number of time slots elapsed since its last data collection until time slot t. It can be shown that auxiliary graph $G(t)$ is a metric graph and its edge weights meet the triangle inequality.

The optimization subproblem with optimization objective $F(t)$, defined in (4), at time slot t now is reduced to *the award collection maximization problem* in $G(t)$, while the latter is to find a closed tour $\mathcal{C}(t)$ in $G(t)$ for the UAV including depot s so that the total award collected by the UAV from sensors in $\mathcal{C}(t)$ is maximized, subject to that the total energy consumption of the UAV in the tour is no greater than its energy capacity $\mathcal{E}_{UAV}$. It is well known that the award collection maximization problem is NP-hard, and there is a constant approximation algorithm for it due to Liang et al. [21].

The proposed approximation algorithm for the optimization subporblem - minimizing $F(t)$ defined in (4), is presented in Algorithm 2.

Algorithm 3: Prediction Algorithm of the Volume of Data Generated by Each Sensor $v_i \in V$ at Time Slot t.

Input: Historical data traces of sensor $v_i \in V$ before time slot t' and its corresponding trained LSTM stored at its DT_i in $\mathcal{G} = (\mathcal{N}, \mathcal{E})$.

Output: Predict the volume $\hat{D}_{v_i}(t)$ of data generated by sensor v_i from its last data collection to time slot t with $1 \leq t \leq T$.

1: Let $\zeta_i(1), \zeta_i(2), \ldots, \zeta_i(t')$ be the sequence of the volume of generated data of sensor v_i at each time slot before time slot t;
2: Feed $\zeta_i(1), \zeta_i(2), \ldots, \zeta_i(t')$ into the LSTM of sensor v_i sequentially, and the predicted volume $\hat{\zeta}_i(t'+1)$ of data generated by sensor v_i is then obtained;
3: $\hat{D}_{v_i}(t) \leftarrow \hat{\zeta}_i(t'+1)$;
4: **for** $k \leftarrow t'+1$ to $t-1$ **do**
5: Obtain $\hat{\zeta}_i(k+1)$ by feeding $\hat{\zeta}_i(k)$ into the LSTM of sensor v_i;
6: $\hat{D}_{v_i}(t) \leftarrow \hat{D}_{v_i}(t) + \hat{\zeta}_i(k+1)$;
7: **return** $\hat{D}_{v_i}(t)$ for each sensor $v_i \in V$ at time slot t.

B. Predicting the Volume of Update Data of Each Object

In the construction of auxiliary graph $G(t)$, we assumed that the volume $D_{v_i}(t)$ of data generated by each sensor $v_i \in V$ is given. which in fact is unknown. We here adopt the Long Short-Term Memory method (LSTM) [8] on DT_i of each sensor $v_i \in V$ to predict its update data volume $D_{v_i}(t)$ by time slot t, through analyzing its historical traces of data generated at previous time slots. We observe that the volume of data generated by sensor v_i has strong temporal patterns at different time slots. We thus make use of a deep neural network (DNN) for time series data predictions [8].

Let $\zeta_i(t)$ be the volume of data generated by sensor v_i at time slot t. We use the DNN of each object $v_i \in V$ to work as a non-linear function $f(\zeta_i(1), \zeta_i(2), \ldots, \zeta_i(t))$, and the output of function $f(\cdot,\cdot,\ldots,\cdot)$ is the predicted volume $\hat{\zeta}_i(t+1)$ of data generated by sensor v_i at time slot $t+1$.

For each sensor $v_i \in V$, we first use its historical data volume at each time slot to train the learn-able parameters of the DNN of object v_i. We then use the DNN to predict the volume of data generated by the sensor at each later time slot. We finally obtain the predicted volume $\hat{D}_{v_i}(t)$ of sensor v_i from its last synchronization to time slot t, by adding up the volumes of data generated at these time slots.

Let $\zeta_i(t)$ be the volume of data generated by sensor v_i at time slot t. Assume that we aim to predict $D_{v_i}(t_2)$ and let t_1 be the last time slot sensor v_i was synchronized. We feed $\zeta_i(1), \zeta_i(2), \ldots, \zeta_i(t_1)$ into the LSTM of sensor (object) v_i sequentially. The LSTM then outputs the predicted volume $\hat{\zeta}_i(t_1+1)$ of the data generated by sensor v_i at time slot t_1+1. We then feed $\hat{\zeta}_i(t_1+1)$ to the LSTM to obtain $\hat{\zeta}_i(t_1+2)$, and so on. This procedure continues until we obtain $\hat{\zeta}_i(t_2)$. $\hat{D}_{v_i}(t_2)$ then can be calculated, i.e., $\hat{D}_{v_i}(t_2) = \sum_{t'=t_1+1}^{t_2} \hat{\zeta}_i(t')$. The detailed algorithm is given in Algorithm 3.

Authorized licensed use limited to: CITY UNIV OF HONG KONG. Downloaded on December 12,2024 at 01:07:45 UTC from IEEE Xplore. Restrictions apply.

C. Algorithm for the DT State Staleness Minimization Problem

Inspired by the optimal solution to the special case of the problem, the solution may contain top-K sorted sensors with the largest DT staleness due to the fact that their DTs have not been synchronized with them for a while. Meanwhile, the value of K should be as large as possible, which implies that more and more DT states of sensors can be updated per update round. However, the K chosen sensors may be far away from each other in their locations, implying that a closed tour including all them may result in the total energy consumption of the UAV larger than its energy capacity. In other words, the top K sensors may not form a feasible solution to the problem. On the other hand, if a sensor is included in the UAV tour, the volume of data generated by the sensor since its last data collection (synchronization) is unknown. The volume of the data however is critical to determine the next UAV tour, as it will determine the hovering duration of the UAV on the sensor and the amount of energy consumed on its data collection.

In the following, we devise an efficient algorithm for the problem, by incorporating the above two mentioned issues. That is, the proposed algorithm should assign higher priorities to sensors with larger staleness on their DT states, i.e., the top K sensors should be assigned with higher priorities to encourage their inclusions in the UAV tour with high probability. Meanwhile, the proposed algorithm should have an effective prediction mechanism to accurately predict the volume of data generated by each sensor since its last data collection, and this prediction result will be used for finding the next tour of the UAV.

As the volume of update data of each object (sensor) in a UAV tour is the predicted one, this predicted value may be larger or smaller than the actual one. If the actual data volume of a sensor is larger than its predicted one, the UAV only collects the predicted amount of data, leaving the rest data uncollected; otherwise (the actual volume is less than the predicted one), the UAV stays above the sensor much longer than needed for collecting data from that sensor, resulting in wasting the UAV energy. To reduce unnecessary energy consumption of the UAV and maximize data collection, we develop an adaptive data collection strategy for tour $\mathcal{C}(t)$ of the UAV at time slot t s follows. Let $v_0, v_1, v_2, \ldots, v_{m-1}$ be sensors in tour $\mathcal{C}(t)$ in their visiting order with $v_0 = s$. The *planed energy budget* $\hat{\mathcal{E}}(v_i, t)$ of the UAV when visiting sensor v_i in $\mathcal{C}(t)$ is defined as $\hat{\mathcal{E}}(v_i, t) = \sum_{j=0}^{i}(\eta_1 \cdot \hat{D}_{v_j}(t)/B + \eta_2 \cdot \frac{dist(v_j, v_{(j+1) \bmod m})}{sp})$ with $0 \leq i \leq m-1$ and $1 \leq m \leq |V|$, where $\hat{D}_{v_i}(t)$ is the predicted volume of data generated by sensor $v_i \in V$ since its last data collection.

Assuming that the UAV is currently visiting sensor v_i for its data collection. If the UAV finishes the data collection before its allocated time duration at v_i, it moves to the next sensor $v_{(i+1) \bmod m}$ for data collection; otherwise (the predicted volume of data is no larger than the actual volume of data of sensor v_i), if its actual energy consumption $\mathcal{E}(v_i, t)$ so far is less than its planned energy budget $\hat{\mathcal{E}}(v_i, t)$, it continues data collection from sensor v_i until reaching its planned energy budget $\hat{\mathcal{E}}(v_i, t)$ or no more data can be collected from sensor v_i. Its total energy consumption is upper bounded by $\mathcal{E}_{UAV}$ when it finishes the tour, and the accumulative volume of uncollected data from the sensors in the tour can be significantly reduced, which is demonstrated in the later empirical analysis.

Algorithm 4: Algorithm for the DT State Staleness Minimization Problem.

Input: A set V of sensors with each $v_i \in V$ having a DT_i in $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, and a UAV with energy capacity $\mathcal{E}_{UAV}$ located at depot s.
Output: A schedule of DT state synchronizations between sensors and their DTs at each time slot t with $1 \leq t \leq T$.

Assume that $D_{v_i}(1)$ of each sensor $v_i \in V$ is given;
$t \leftarrow 1$; $S \leftarrow \emptyset$;
while $t \neq T + 1$ **do**
  Calculate the staleness length $l_{v_i}(t)$ of DT_i of each sensor $v_i \in V$;
  Construct an auxiliary graph $G(t) = (V \cup \{s\}, E'(t); c(\cdot,\cdot), \pi(\cdot))$;
  Find a closed tour $\mathcal{C}(t)$ in $G(t)$ by the approximation algorithm, Algorithm 2. Let $v_0, v_1, \ldots, v_{m-1}$ be sensors in $\mathcal{C}(t)$ with $v_0 = s$;
  Calculate the planned energy consumption budget $\hat{\mathcal{E}}(v_i)$ of each sensor $v_i \in \mathcal{C}(t)$;
  for each $v_i \in \mathcal{C}(t)$ **do**
    Collect data from sensor v_i in tour $\mathcal{C}(t)$ by the UAV;
    if no more data can be collected from v_i within its allocated time duration **then**
      The UAV moves to sensor $v_{(i+1) \ mod \ m}$ for data collection;
    else
      The actual volume of data of v_i is larger than the predicted one, the UAV continues collecting data from v_i until reaching its planned energy budget $\hat{\mathcal{E}}(v_i, t)$ or no more data can be collected from v_i;
    Reset the data collection duration at sensor $v_i \in \mathcal{C}(t)$ by the above adjustment;
  Let $\mathcal{C}'(t)$ be the updated closed tour for the UAV with the adjusted hovering time at each sensor in the tour;
  The UAV uploads its collected data to the MEC network $\mathcal{G}$ at depot s, and the DT states of sensors in $\mathcal{C}'(t)$ are synchronized with their sensors;
  $S \leftarrow S \cup \{\mathcal{C}'(t)\}$;
  Predict the volume $\hat{D}_{v_i}(t+1)$ of data generated by each sensor $v_i \in V$ on the LSTM of v_i from its last data collection to time slot $t+1$, by applying Algorithm 3;
  $t \leftarrow t + 1$;
return The synchronization schedule S from time slot 1 to time slot T.

We now propose an efficient algorithm, Algorithm 4, for the DT state staleness minimization problem as follows.

D. Algorithm Analysis

Lemma 1: Given the volume $D_{v_i}(t)$ of update data of each sensor v_i with staleness length $l_{v_i}(t)$ of DT_i since its last synchronization, a UAV with energy capacity $\mathcal{E}_{UAV}$, the maximum

Authorized licensed use limited to: CITY UNIV OF HONG KONG. Downloaded on December 12,2024 at 01:07:45 UTC from IEEE Xplore. Restrictions apply.

award collection tour in the auxiliary graph $G(t)$ of the UAV including depot s corresponds to an optimal solution to the optimization subproblem with minimizing $F(t)$ in (4) at time slot t with $1 \leq t \leq T$.

Proof: If there is a maximum award collection tour in $G(t)$ for the UAV, the DT states of sensors in the tour will be synchronized with their sensors in the end of time slot t, i.e., the total award collected from sensors in the tour is the maximum one, which is also the maximum reduction on the value of $F(t)$ defined in (4) at time slot t with $1 \leq t \leq T$.

Now, assume that there is an optimal solution to the optimization subproblem at time slot t. Then, the visiting order of sensors by the UAV in this tour is given. A closed tour in $G(t)$ can then be derived from the visiting order of sensors, and the total award collected from the sensors in the closed tour should not be greater than the maximum one in the optimal solution to the award collection maximization problem in $G(t)$. It can be shown that the total award collected from sensors in the closed tour is the maximum one; otherwise, the value reduction of $F(t)$ equals the total award, which is not the maximum one. This contradicts that the solution is optimal, implying the maximum reduction of the value of $F(t)$ defined in (4) after the UAV tour.□

Theorem 4: There is a $(1+\frac{2\beta}{3})$-approximation algorithm Algorithm 2, for the special optimization subproblem with optimization objective $F(t)$ defined in (4), under the assumption that the volume of data generated by each sensor since its last data collection is given, where $OPT(t)$ is the optimal value of function $F(t)$, $A^*(t)$ is the maximum value of the accumulative award collected by a UAV along an optimal closed tour, and $\beta = \max\{A^*(t)/OPT(t) \mid 1 \leq t \leq T\}$ with $\beta > 1$.

Proof: Following Lemma 1, if there is an optimal solution to the award collection maximization problem in $G(t)$, there is an optimal solution to the special optimization sub-problem at time slot t.

Consider the construction of the auxiliary graph $G(t)$. Assume that the closed tour $\mathcal{C}(t)$ delivered by Algorithm 2 consists of sensors $v_{i_0}, v_{i_1}, \ldots, v_{i_{m-1}}$ with $m \leq |V|$, and one of the nodes in $\mathcal{C}(t)$ is the depot s and $D_s(t) = 0$. The total amount of energy consumed of the UAV in $\mathcal{C}(t)$ is $\sum_{j=0}^{m-1} c(v_{i_j}, v_{i_{(j+1) \bmod m}})$ $(= \sum_{j=0}^{m-1}(\eta_1 \cdot \frac{D_{v_{i_j}}(t)}{B} + \eta_2 \cdot \frac{dist(v_{i_j}, v_{i_{(j+1) \bmod m}})}{sp}))$, where $D_{v_{i_j}}(t)$ is the volume of update data of sensor v_{i_j} for collection at time slot t and B is the bandwidth of the UAV for data collection at each sensor. As the total energy consumption on tour $\mathcal{C}(t)$ is upper bounded by $\mathcal{E}_{UAV}$, the energy capacity of the UAV will not be violated, and the solution obtained is feasible.

The rest is to analyze the reduction on the accumulative DT state staleness length $F(t)$ of all sensors at time slot t. Let $OPT(t)$ be the optimal value of function $F(t)$ and let $OPT(t) = F(t) - A^*(t)$. It can be seen that $A^*(t)$ is the maximum (optimal) value of the accumulative award collected by the UAV along an optimal closed tour in $G(t)$. As the award collection maximization problem in $G(t)$ is NP-complete, there is a $\frac{1}{\gamma+1}$-approximation algorithm for it when there is a given $\frac{1}{\gamma}$-approximation algorithm for the orienteering problem [21], where γ is a constant with $\gamma \geq 1$. Considering the best approximation algorithm for the orienteering problem with an approximation ratio of $\frac{1}{2}$ so far [29], we have $\gamma = 2$. Thus, there is a $\frac{1}{3}$-approximation algorithm for the award collection maximization problem at time slot t. Following the approximation algorithm for the award collection maximization problem in $G(t)$, a $\frac{1}{3}$-approximate solution is delivered by approximation algorithm Algorithm 2. In other words, the value of the approximate solution is no less than $\frac{1}{3}A^*(t)$.

Let ρ be the ratio of the approximation algorithm for minimizing $F(t)$. We analyze ρ by distinguishing the following two cases.

Case 1: If $A^*(t) \leq OPT(t)$, we have

$$\rho \leq \frac{F(t) - \frac{1}{3}A^*(t)}{OPT(t)} = \frac{OPT(t) + A^*(t) - \frac{1}{3}A^*(t)}{OPT(t)}$$
$$= \frac{OPT(t) + \frac{2}{3}A^*(t)}{OPT(t)} = 1 + \frac{\frac{2}{3}A^*(t)}{OPT(t)} \leq 5/3.$$

Case 2: Otherwise, let $\beta = \max\{A^*(t)/OPT(t) \mid 1 \leq t \leq T\}$, clearly $\beta > 1$. We have

$$\rho \leq \frac{F(t) - \frac{1}{3}A^*(t)}{OPT(t)} = \frac{OPT(t) + A^*(t) - \frac{1}{3}A^*(t)}{OPT(t)}$$
$$= \frac{OPT(t) + \frac{2}{3}A^*(t)}{OPT(t)} = 1 + \frac{\frac{2}{3}A^*(t)}{OPT(t)}$$
$$\leq 1 + \frac{2\beta}{3}, \text{ as } \beta \geq A^*(t)/OPT(t) \text{ by the assumption.}$$

The theorem then follows. □

Theorem 5: There is an efficient algorithm Algorithm 4 for the DT state staleness minimization problem in an MEC network $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, and the solution delivered by the algorithm is feasible.

Proof: The feasibility of the solution obtained by Algorithm 4 can be shown exactly as we did for Theorem 4, i.e., the total energy consumption of the UAV per tour is upper bounded by $\mathcal{E}_{UAV}$ if we adopt the adaptive strategy of data collection for prediction error corrections.

Since the UAV tour at time slot t is based on the prediction of the volume of data generated by each sensor since its last data collection, such a prediction may not be 100% accurate and the prediction error is unavoidable, this implies that the actual volumes of data generated of some sensors may be larger or smaller than their predicted ones. In Algorithm 4, we adopt an adaptive strategy of data collection to mitigate the prediction errors, and show that the total energy consumption of the UAV per update round (per time slot) is no greater than its energy capacity $\mathcal{E}_{UAV}$. Thus, the solution delivered by Algorithm 4 is feasible, and the accumulative volume of uncollected data is significantly reduced in comparison with the case without adopting the adaptive strategy of data collection in later empirical evaluations. The solution thus is feasible. □

Authorized licensed use limited to: CITY UNIV OF HONG KONG. Downloaded on December 12,2024 at 01:07:45 UTC from IEEE Xplore. Restrictions apply.

VI. Fidelity-Aware Query Services on DT Data

To demonstrate the critical impact of the staleness of DT states on the fidelity of query results, in this section, we consider fidelity-aware query evaluation on the DT data in $\mathcal{G}$, by developing a cost-effective evaluation plan for fidelity-aware queries.

A. Evaluation Trees for Fidelity-Aware Queries

Recall that query $Q(t) = (V', \delta, \vartheta)$ is issued by user u from AP n_u at time slot t. Assume that the query result of $Q(t)$ will be returned to cloudlet n_u as well. As the DTs of sensors in V' are placed in different cloudlets, the collected data from these sensors stored at their DTs will be aggregated for the query evaluation.

To minimize the evaluation cost of query $Q(t)$, we will find a query evaluation tree T_u rooted at cloudlet n_u and spanning all cloudlets that contain the DTs of sensors in V' such that the evaluation cost $C(Q(t))$ is minimized. In the following we detail the construction of tree T_u. Let $\mathcal{N}'$ be the set of cloudlets that contain the DTs of sensors in V' of $Q(t)$. We further assume that cloudlets in $\mathcal{N}'$ are sorted in non-increasing order of the accumulative volume of DT data of sensors in V' in them. Let $n_{c_1}, n_{c_2}, \ldots, n_{c_{|\mathcal{N}'|}}$ be the sorted cloudlet sequence, where cloudlet n_{c_1} contains the largest accumulative volume of DT data of sensors in V', while cloudlet $n_{c_{|\mathcal{N}'|}}$ contains the least accumulative volume of DT data of sensors in V', respectively.

The query evaluation tree T_u for $Q(t)$ is constructed greedily. Within each iteration, a cloudlet in $\mathcal{N}'$ with the minimum evaluation cost gain (which will be defined later) will be added to T_u. This procedure continues until all cloudlets in $\mathcal{N}'$ are added to T_u. The construction of T_u is obtained through constructing a series of partial evaluation trees $T(u,k)$ that spans k cloudlets in $\mathcal{N}'$ with the root at cloudlet $n_u \in \mathcal{N}$.

When $k = 1$, the partial evaluation tree $T(u,1)$ is a shortest path P_1 in $\mathcal{G}(\mathcal{N}, \mathcal{E})$ between cloudlet n_{c_1} and cloudlet n_u, where each edge $(v_i, v_{i+1}) \in P_1$ with $v_1 = n_{c_1}$, $v_{|P_1|+1} = n_u$, and v_{i+1} is the parent of v_i in the partial evaluation tree. The accumulative volume of data on edge $(v_i, v_{i+1}) \in P_1$ is $agg^{(T(u,1))}(v_i) = D_{c_1}$, which is the accumulated volume of DT data of sensors in V' stored at cloudlet n_{c_1}, where $agg^{(T(u,1))}(v)$ is an aggregate function related to query $Q(t)$ at node v in tree $T(u,1)$. The communication cost of tree edge (v_i, v_{i+1}) is $\rho(v_i, v_{i+1}) \cdot agg^{(T(u,1))}(v_i)$, where $\rho(v_i, v_{i+1})$ is the communication cost of one unit data transfer on edge (v_i, v_{i+1}).

Assuming that tree $T(u,k-1)$ has been constructed with $k > 1$, we now construct $T(u,k)$ by adding a cloudlet $n_{c_i} \in \mathcal{N}'$ to it with the minimum evluation cost gain $\Delta(T(u,k-1), n_{c_i}, n_{c_j})$, where n_{c_j} is the tree node connecting cloudlet $n_{c_i} \in \mathcal{N}'$ through a shortest path $P_k(n_{c_i}, n_{c_j})$ in $\mathcal{G} \setminus T(u,k-1)$ that has not been contained in $T(u,k-1)$ yet.

Let $P_k^T(n_{c_j}, n_u)$ be the tree path in $T(u,k-1)$ between cloudlet n_{c_j} and cloudlet n_u. For the volume of data to be routed along a routing path from n_{c_i} to n_u, the routing path can be further divided into two segments $P_k(n_{c_i}, n_{c_j})$ from n_{c_i} to n_{c_j} and $P_k^T(n_{c_j}, n_u)$ in tree $T(u,k-1)$ from n_{c_j} to n_u, respectively. The data volume on each edge in $P_k(n_{c_i}, n_{c_j})$ is D_{c_i}, while the data volume on each edge in $P_k^T(n_{c_j}, n_u)$ is the aggregate result of the volume D_{c_i} with other existing ones in $T(u,k-1)$. The evaluation cost gain $\Delta(T(u,k-1), n_{c_i}, n_{c_j})$ of $T(u,k)$ by adding a shortest path from cloudlet n_{c_i} to a tree node n_{c_j} is defined as

$$
\begin{aligned}
&\Delta(T(u,k-1), n_{c_i}, n_{c_j}) \\
&= \sum_{(x,y)\in P_k^T(n_{c_i},n_u)}^{y=p(x)} (\rho(x,y)D^{(T(u,k))}(x,y) + agg^{(T(u,k))}(y)) \\
&\quad - \sum_{(x,y)\in P_{k-1}^T(n_{c_j},n_u)}^{y=p(x)} (\rho(x,y)D^{(T(u,k-1))}(x,y) \\
&\quad + agg^{(T(u,k-1))}(y)) \\
&= \sum_{(x,y)\in P_k(n_{c_i},n_{c_j})}^{y=p(x)} (\rho(x,y)D^{(T(u,k))}(x,y) + agg^{(T(u,k))}(y)) \\
&\quad + \sum_{(x,y)\in P_k^T(n_{c_j},n_u)}^{y=p(x)} (\rho(x,y)D^{(T(u,k))}(x,y) + agg^{(T(u,k))}(y)) \\
&\quad - \sum_{(x,y)\in P_{k-1}^T(n_{c_j},n_u)}^{y=p(x)} (\rho(x,y)D^{(T(u,k-1))}(x,y) \\
&\quad + agg^{(T(u,k-1))}(y)), \qquad (11)
\end{aligned}
$$

where $P_{k-1}^T(n_{c_j}, n_u)$ is the tree path in $T(u,k-1)$, $y = p(x)$ implies that node y is the parent of node x in the tree. Function $agg^{(T(u,k))}(y)$ at node $y \in P_k^T(n_{c_j}, n_u)$ in tree $T(u,k)$ now adds one more data volume D_{c_i} from cloudlet n_{c_i} for aggregation in addition to the original ones from its children in $T(u,k-1)$. The value $D^{(T(u,k))}(x,y)$ of edge (x,y) in $T(u,k)$ is the accumulative volume of data transferred from x to y, which is no less than its value in tree $T(u,k-1)$ and determined by aggregate function $g(\cdot)$.

The construction of query evaluation tree T_u for $Q(t)$ proceeds as follows. It contains only cloudlet n_u as its root initially. Cloudlet n_{c_1} is then added to T_u, i.e., tree $T(u,1)$ is built. Each other cloudlet $n_{c_i} \in \mathcal{N}' \setminus \{n_{c_1}\}$ is then added to T_u iteratively. In iteration k with $2 \le k \le |\mathcal{N}'|$, a cloudlet $n_{c_{i_0}} \in \mathcal{N}'$ with the minimum evaluation cost gain in (11) is added to the partial evaluation tree $T(u,k-1)$ until all cloudlets in $\mathcal{N}'$ are added. The detailed algorithm for the construction of query evaluation tree T_u for $Q(t)$ is presented in Algorithm 5.

B. Algorithm for Fidelity-Aware Query Evaluation

Having query evaluation tree T_u for query $Q(t)$ of user u issued at time slot t, the rest is to examine whether the evaluation plan T_u is feasible, by collecting the DT information of sensors in V' and forwarding the information to cloudlet n_u along tree paths of T_u. The proposed algorithm for the fidelity-aware query evaluation problem is given in Algorithm 6.

Authorized licensed use limited to: CITY UNIV OF HONG KONG. Downloaded on December 12,2024 at 01:07:45 UTC from IEEE Xplore. Restrictions apply.

Algorithm 5: The Construction of a Query Evaluation Tree T_u for $Q(t)$ Rooted at Cloudlet $n_u \in \mathcal{N}$ in MEC $\mathcal{G} = (\mathcal{N}, \mathcal{E})$.

Input: An MEC network $\mathcal{G} = (\mathcal{N}, \mathcal{C})$ and a query $Q(t) = (V', \delta, \vartheta)$, assuming cloudlets that contain the DTs of sensors in V' are sorted as $n_{c_i}, n_{c_2}, \ldots, n_{c_{|\mathcal{N}'|}}$ in non-increasing order of the data volume of DTs, where cloudlet n_{c_i} has a data volume D_{c_i}, with $D_{c_1} \geq D_{c_2} \geq \cdots \geq D_{c_{|\mathcal{N}'|}}$, $1 \leq c_i \leq |\mathcal{N}|$ and $1 \leq i \leq |\mathcal{N}|$.

Output: A query evaluation tree T_u rooted at n_u and spanning all cloudlets containing the DTs of sensors in V' such that the evaluation cost $C(Q(t))$ is minimized.

$C(Q(t)) \leftarrow \infty$; /* the solution cost*/
$T_u \leftarrow \emptyset$; /* the query evaluation tree */
$\mathcal{N}' \leftarrow \{n_{c_1}, n_{c_2}, \ldots, n_{c_{|\mathcal{N}'|}}\}$;
Find a shortest path P_1 in $\mathcal{G}(\mathcal{N}, \mathcal{E})$ from n_{c_1} to n_u;
$T_u \leftarrow P_1$; $k \leftarrow 1$; $\mathcal{N}' \leftarrow \mathcal{N}' \setminus \{n_u\}$; $T(u, 1) \leftarrow P_1$;
while $\mathcal{N}' \neq \emptyset$ **do**
  $k \leftarrow k + 1$;
  Construct the partial evaluation tree $T(u, k)$ from tree $T(u, k-1)$;
  for each $n_{c_i} \in \mathcal{N}'$ **do**
    for *each* $n_{c_j} \in T_u$ **do**
      Find a shortest path $P_k(n_{c_i}, n_{c_j})$ in $\mathcal{G} \setminus T(u, k-1)$ between cloudlets n_{c_i} and n_{c_j};
      Compute the costs of segments $P_k(n_{c_i}, n_{c_j})$ and $P_k^T(n_{c_j}, n_u)$ of $P_k^T(n_{c_i}, n_u)$;
      Construct a potential partial evaluation tree $T(u, k) \leftarrow T(u, k-1) \cup P_k(n_{c_i}, n_{c_j})$, by connecting cloudlet n_{c_i} to tree $T(u, k-1)$ through the tree node n_{c_j} with a shortest path $P_k(n_{c_i}, n_{c_j})$;
      Calculate $\Delta(T(u, k-1), n_{c_i}, n_{c_j})$ by (11);
    $i_0, j_0 \leftarrow \arg\min_{n_{c_i} \in \mathcal{N}',\ n_{c_j} \in T(u,k-1)} \Delta(T(u, k-1), n_{c_i}, n_{c_j})$;
    $T(u, k) \leftarrow T(u, k-1) \cup P_k(n_{c_{i_0}}, n_{cj_0})$; /* adding cloudlet node $n_{c_{i_0}}$ to the multicast tree */
    $\mathcal{N}' \leftarrow \mathcal{N}' \setminus \{n_{c_{i_0}}\}$;
$T_u \leftarrow T(u, k)$;
$C(Q(t)) \leftarrow C(T_u)$;
return Solution T_u with evaluation cost $C(T_u)$.

C. Algorithm Analysis

Theorem 6: Given an MEC network $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, a set V of objects, the DTs of objects (sensors) are placed in cloudlets of $\mathcal{G}$, a user u who issues a fidelity-aware query $Q(t) = (V', \delta, \vartheta)$ under the coverage of AP $n_u \in \mathcal{N}$, there is a query evaluation algorithm, Algorithm 6, for the fidelity-aware query evaluation problem of $Q(t)$ in $\mathcal{G}$, which delivers a feasible evaluation plan if the plan does exist, and the algorithm takes $O(|\mathcal{E}|^2 \cdot |\mathcal{N}|^2 \cdot \log |\mathcal{N}|)$ time.

Proof: It can be seen that if the query evaluation plan is feasible, the solution delivered by Algorithm 6 is feasible.

Algorithm 6: Algorithm for Fidelity-Aware Query Evaluation.

Input: An MEC network $\mathcal{G}(\mathcal{N}, \mathcal{E})$ and a query $Q(t) = (V', \delta, \vartheta)$ issued by user u under the coverage of AP n_u at time slot t with $1 \leq t \leq T$, assuming that cloudlets containing the DTs of sensors in V' are sorted as $n_{c_i}, n_{c_2}, \ldots, n_{c_{|\mathcal{N}'|}}$ in non-increasing order of the accumulative volume of DT data in them, i.e., each cloudlet n_{c_i} has a data volume D_{c_i}, with $D_{c_1} \geq D_{c_2} \geq \cdots \geq D_{c_{|\mathcal{N}'|}}$, where $1 \leq c_i \leq |\mathcal{N}'|$ and $1 \leq i \leq |\mathcal{N}|$.

Output: A query evaluation tree T_u for $Q(t)$ rooted at n_u and spanning cloudlets in $\mathcal{N}'$ such that the evaluation cost $C(Q(t))$ is minimized.

Find a cost-effective query evaluation tree T_u for $Q(t)$, by invoking Algorithm 5;
Aggregate DT data by the aggregate function $agg(\cdot)$ for $Q(t)$ at each cloudlet in T_u from leaves to root n_u;
Identify the subset $V'_{\leq\delta}$ of set V' at cloudlet n_u;
if $|V'_{\leq\delta}|/|V'| \geq \vartheta$ **then**
  A cost-effective feasible evaluation plan T_u is obtained
else
  No solution can meet the query requirement; EXIT.

The rest is to analyze the time complexity of the evaluation algorithm. The dominant time complexity in Algorithm 6 is the construction of a query evaluation tree T_u for $Q(t)$, which consists of $|\mathcal{N}'|$ iterations (with $|\mathcal{N}'| \leq \mathcal{N}|$ and $\mathcal{N}'$ is the set of cloudlets containing the DTs of sensors in V'). Within each iteration, it finds a shortest path in $\mathcal{G}$ that takes $O(|\mathcal{N}| \cdot |\mathcal{E}| \cdot \log |\mathcal{N}|)$ time. Thus, the construction of tree T_u takes $O(|\mathcal{E}| \cdot |\mathcal{N}|^2 \cdot \log |\mathcal{N}|)$ time. The determination whether tree T_u is feasible for fidelity-aware query $Q(t) = (V', \delta, \vartheta)$ can be done within $O(|\mathcal{N}|)$ time, by collecting the DT state information of sensors in V' and aggregating the data at tree nodes. The query evaluation algorithm thus takes $O(|\mathcal{E}| \cdot |\mathcal{N}|^2 \cdot \log |\mathcal{N}|)$ time. □

VII. Performance Evaluation

In this section, we evaluated the proposed algorithms for the DT state staleness minimization problem in an MEC network through simulations, using a real dataset. We also investigated the impacts of important parameters on the performance of the proposed algorithms.

A. Experimental Settings

We considered 500 objects (sensors) randomly deployed in a circular area with a 500 meter radius and a depot s located at the center of the area. A UAV is deployed at depot s initially. For the special DT state staleness minimization problem, we assumed that the synchronization budget is $K = 100$ sensors per UAV tour. For the DT state staleness minimization problem, the UAV has energy capacity $\mathcal{E} = 3 \times 10^5\ joules$ at constant flying speed $\nu = 10\, m/s$. The energy consumption rates of the UAV on hovering and traveling are $\eta_1 = 150\ J/s$ and $\eta_2 = 100\ J/s$,

Authorized licensed use limited to: CITY UNIV OF HONG KONG. Downloaded on December 12,2024 at 01:07:45 UTC from IEEE Xplore. Restrictions apply.

respectively [20], and the data transmission rate from objects to the UAV is 1 Mbps [38]. The monitoring period consists of 1,000 equal time slots. In our simulations, we adopt a real-world dataset: the California traffic usage dataset collected from 17,544 sensors on freeways in San Francisco Bay area [10]. Each data point in the dataset is a time series of hourly recorded road occupy rates (between 0 and 1).

To evaluate the performance of the proposed algorithms, we used the road occupy rate data to simulate the volume of data generated by each sensor. We assumed that a sensor can generate at most $[5, 10]$ MB data per time slot [38]. For each sensor, we selected a time series in the California traffic usage dataset and map the road occupy rate (between 0 and 1) to the generated data volume (between 0 and 10 MB). The data collected by the UAV is sent to the DTs of objects (sensors), where the DTs of objects (sensors) are randomly deployed to cloudlets in an MEC network that consists of 20 APs, and each AP has a co-located cloudlet.

The MEC network is generated by GT-ITM [16]. Users at different locations issue fidelity-aware queries to DTs randomly, and each query requests 4-8 different DTs [16]. The fidelity threshold ratio ϑ is set at 0.8, the threshold δ is set at 1, and the aggregation function $g(\cdot)$ is randomly chosen from a family of aggregation functions, e.g., {sum, average, max, min} [16]. The transmission cost per MB along an edge in G between two APs is randomly drawn from \$0.1 to \$0.4, and the processing cost per MB data in a cloudlet is randomly drawn from \$0.04 to \$0.06 [16].

To evaluate the performance of Algorithm 1 for the special case of the problem by choosing top-K sensors, we proposed a baseline algorithm Random that randomly chooses K sensors to synchronize with their DTs at each time slot.

To evaluate the performance of Algorithm 4, we adopted two heuristics as follows. (a) Heuristic Heur proceeds as follows. At each time slot t, it first sorts sensors in descending order of their staleness. it then constructs a closed tour for the UAV greedily. Specifically, the tour contains only depot s initially. The sorted sensors are then added to the tour one by one. A sensor is added to the tour only if the energy consumption of tge tour so far does not exceed the energy capacity of the UAV; otherwise the next sorted sensor is examined. This procedure continues until all sorted sensors are considered. (b) Heuristic MRE-S in [31] proceeds greedily too. The UAV tour is constructed by adding sensors one by one, and the initial tour contains only the depot s . It then selects a sensor with the largest ratio of staleness to the incremental energy cost of adding the sensor. This procedure continues until no more sensors can be added, due to insufficient energy left by the UAV.

To evaluate the adaptive strategy of data collection for the DT state staleness minimization problem, a variant of Algorithm 4 without adopting the policy is also considered, referred to as NoPolicy, in which the UAV strictly follows the tour delivered by Algorithm 2, using the predicated data, i.e., line 10–13 in Algorithm 4 are ignored.

To study the performance of the proposed algorithms for the (special) DT state staleness minimization problem, we refer to Algorithms 2 and 4 as Actual and Pred with and without the prediction on the volume of the update data generated by each sensor since the last DT synchronization.

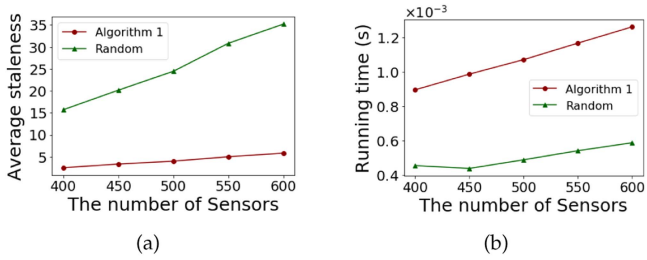


Fig. 3. Performance of different algorithms for the special DT state staleness minimization problem by varying the number of objects.

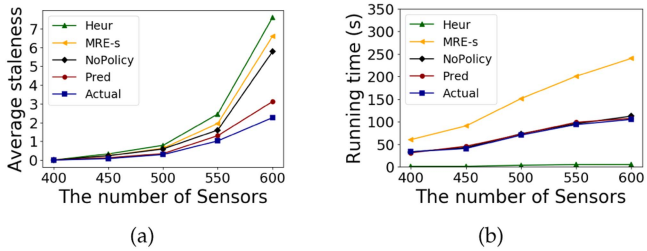


Fig. 4. Performance of different algorithms for the DT state staleness minimization problem by varying the number of objects.

The value in each figure is the mean of the results out of 50 network instances with the same size, and the running times of the algorithms are based on a machine with 3.6 GHz Intel i7 single-core CPU and 16 GB RAM. Unless otherwise specified, the parameters are adopted in the default setting.

B. Performance Evaluation of Different Algorithms for the DT State Staleness Minimization Problem

We first investigated the performance of different algorithms for the special DT state staleness minimization problem by varying the number of objects from 400 to 600. We can see from Fig. 3(a) that Algorithm 1 delivers a solution with a lower average DT state staleness than that of algorithm Random, and Fig. 3(b) illustrates their running time.

We then studied the performance of different algorithms for the DT state staleness minimization problem, by varying the number of sensors from 400 to 600. Fig. 4(a) illustrates the average staleness curves delivered by different algorithms. It can be seen that the average DT state staleness length delivered by Pred is much smaller than that by algorithm Heur and MRE-S. Specifically, when there are 500 sensors, the average DT state staleness length delivered by Pred is 0.34, which is approximately 44.92% less than that by MRE-S. With the increase on network size, the DT state staleness gap between Pred and algorithm MRE-S enlarges, indicating that Pred utilizes the UAV energy efficiently. In addition, even if the volume of data generated by each sensor is given, Actual only reduces the average DT state staleness among all objects (sensors) by 15.30% compared with Pred when there are 500

Authorized licensed use limited to: CITY UNIV OF HONG KONG. Downloaded on December 12,2024 at 01:07:45 UTC from IEEE Xplore. Restrictions apply.

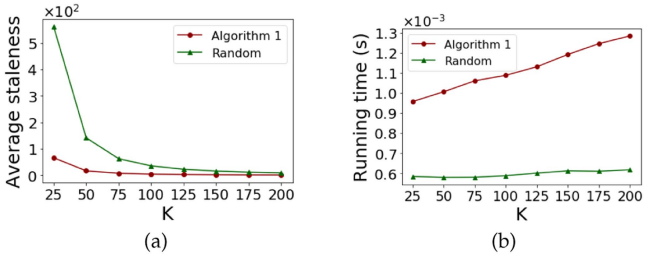


Fig. 5. Performance of different algorithms for the special DT state staleness minimization problem by varying the number K of synchronization objects.

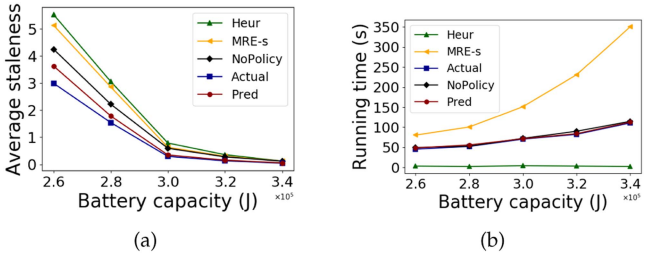


Fig. 6. Performance of different algorithms for the DT state staleness minimization problem, by varying the energy capacity $\mathcal{E}_{UAV}$ of the UAV.

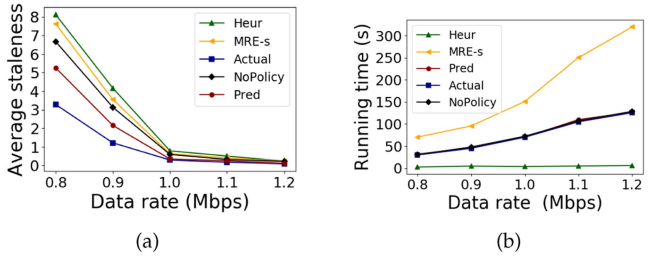


Fig. 7. Performance of different algorithms for the DT state staleness minimization problem by varying the uploading data rate B.

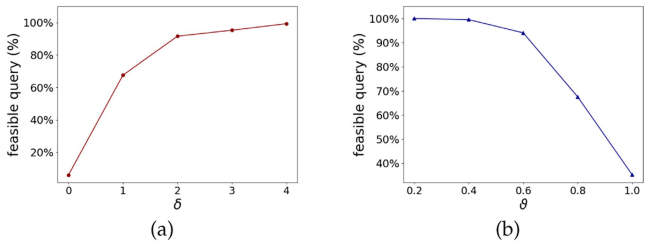


Fig. 8. The percentage of feasible evaluation plans for 1,000 queries randomly generated by varying δ and ϑ, respectively.

objects. This demonstrates the effectiveness of the proposed prediction algorithm. Moreover, algorithm `NoPolicy` increases the average DT state staleness by 70.84% compared with `Pred`. The rationale behind is that the UAV wastes more energy on hovering above sensors even if the data collection from sensors has been finished by algorithm `NoPolicy`.

C. The Impact of Parameters on the Performance of Different Algorithms

We first investigated the impact of parameter K on the performance of different algorithms for the special DT state staleness minimization problem, by varying its value from 25 to 200. It can be seen from Fig. 5(a) that with the increase of K, the average DT state staleness in the solutions delivered by both algorithms decreases, since more DTs can be synchronized with their objects. With the increase of K, the gap of the average DT state staleness between the two comparison algorithms becomes smaller. Fig. 5(b) plots the running time curves of the two mentioned algorithms.

We then evaluated the impacts of parameters $\mathcal{E}_{UAV}$ and B on the performance of the proposed algorithms for the DT state staleness minimization problem, by varying $\mathcal{E}_{UAV}$ from 2×10^5 to 4×10^5 $Joules$. Fig. 6(a) presents the impact of the energy capacity $\mathcal{E}_{UAV}$ on the average DT state staleness. It is noted that the DT state staleness reduces with the growth on the energy capacity, as the UAV can visit more sensors and the visited sensors can synchronize with their DTs per update round. It can also be seen that `Pred` is superior to algorithms `MRE-S`, `Heur` and `NoPolicy`, respectively. The difference of the DT state staleness between the comparison algorithms becomes smaller when the UAV has a larger energy capacity, and the UAV is more tolerant to inaccurate predictions on the volume of update data of objects (sensors) when the UAV has a large energy capacity. Fig. 6(b) plots the running time curves of the four comparison algorithms, which shows that `Pred` requires a shorter running time than `MRE-S`.

We finally studied the impact of the uploading data rate B of objects by varying the value of B from 0.8 Mbps to 1.2 Mbps. Fig. 7(a) shows that the average DT state staleness curves in the solutions delivered by all algorithms, the average DT state staleness decreases with the increase on the data rate of the UAV. This is due to the fact that the growth on the uploading data rate leads to less hovering time above sensors, and thus more sensors can be visited by the UAV per tour. When the uploading data rate is set at 0.8 Mbps, the average DT state staleness in the solution by algorithm `Heur` is 8.12, which is approximately 54.67% more than that by `Pred`. Fig. 7(b) describes the running time curves of different algorithms. Although `Pred` takes a longer running time than that of algorithm `Heur`, the average DT state staleness in its solution is much shorter than that by `Heur`.

D. Performance of Fidelity-Aware Query Services

To evaluate the efficiency of Algorithm 6, we constructed another evaluation tree, which is a minimum spanning tree rooted at source cloudlet $n_u \in \mathcal{N}$ initially. We then removed the leaves from the tree that do not contain the DTs of sensors in V' continuously until all leaves in the resulting tree are cloudlets that contain the DTs of sensors in V'. Denote by `MST` the resulting query evaluation tree for query $Q(t)$ of user u.

We first evaluated a fidelity-aware query by setting the value of δ as 0, 1, 2, 3, and 4, respectively, and for each value of δ, we randomly generate 1,000 queries and then examined the number of feasible evaluation plans among the generated queries. As shown in Fig. 8(a), the percentage of feasible evaluation plans

Authorized licensed use limited to: CITY UNIV OF HONG KONG. Downloaded on December 12,2024 at 01:07:45 UTC from IEEE Xplore. Restrictions apply.

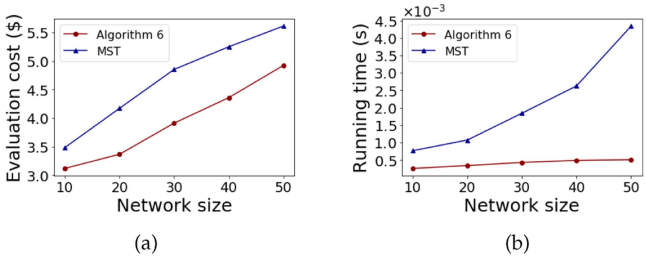


Fig. 9. Performance of different algorithms for the fidelity-aware query evaluation problem by varying network size $|\mathcal{N}|$.

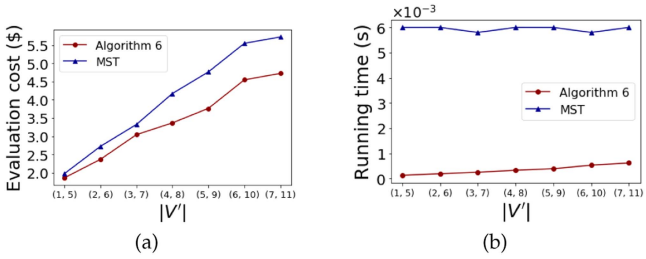


Fig. 10. Performance of different algorithms for the fidelity-aware query evaluation problem by varying the number $|V'|$ of querying sensors.

for queries increases with the growth on the value of δ. The rationale behind is that more DTs can meet specified fidelity requirements of the queries when δ is large. When δ equals 0, the queried DTs must be synchronized with their objects at the current time slot, only 5.97% evaluation plans among 1,000 query plans are feasible. The percentage of feasible evaluation plans can reach 99.32% when δ is set as 4.

We then studied the impact of the data fresh fidelity parameter ϑ on the query result for a query, by increasing its value from 0.2 to 1.0. It can be seen from Fig. 8(b) the percentage of feasible evaluation plans becomes decreasing with the increase on the value of ϑ. The reason is that fewer numbers of DTs involving a query can meet the data fresh fidelity requirement of the query.

We finally investigated the impacts of parameters: the network size $|\mathcal{N}|$, the number $|V'|$ of sensors per query, and the computing cost per unit data ξ, on the query evaluation cost for each fidelity-aware query. The result in the following figures is the mean of the results of 1,000 feasible queries.

We evaluated the impact of network size $|\mathcal{N}|$ by varying its value from 10 to 50. It can be seen from Fig. 9(a) that the query evaluation cost increases with the growth of network size. This can be justified by the growth on the DT data transmission cost in the MEC network. The query evaluation cost by algorithm `MST` is much higher than that by Algorithm 6, and the query evaluation tree delivered by the algorithm does not lead to the minimum transmission cost of DT data. Fig. 9(b) depicts the running time curves of different algorithms.

We investigated the impact of the number $|V'|$ of sensors requested per query, by varying its value range from $[1, 5]$ to $[7, 11]$. Fig. 10(a) shows that the query evaluation cost increases with the growth of $|V'|$. This is due to the additional computing and DT data transmission costs incurred by requesting the DT data of more sensors. Fig. 10(b) depicts the running time curves of the comparison algorithms. It can be seen that the running time of Algorithm 6 increases when more cloudlets are added to the partial evaluation tree, while the running time of algorithm `MST` does not change too much.

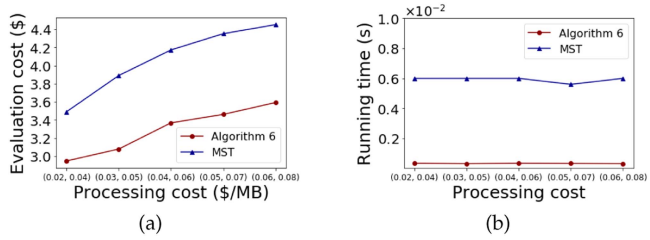


Fig. 11. Performance of different algorithms for the fidelity-aware query evaluation problem, by varying the processing cost per unit data ξ.

We also studied the impact of the processing cost per unit data ξ by varying its value range from $[\$0.02, \$0.04]$ to $[\$0.06, \$0.08]$. Fig. 11(a) depicts that the query evaluation trees delivered by the both algorithms have higher evaluation costs. When the processing cost per unit data is drawn from $[\$0.06, \$0.08]$, the cost of the query evaluation tree delivered by Algorithm 6 is 24.01% higher than that by itself when the processing cost per unit data is drawn from $[\$0.02, \$0.04]$. The running time curves of the two algorithms in Fig. 11(b) indicate that they do not have much changes with the growth of the processing cost.

VIII. Conclusion

In this paper, we considered the budget-constrained DT state synchronization issue between DTs and their physical objects (sensors) in an MEC network. We proposed a novel DT synchronization framework through a real application example to demonstrate how to make use of the freshness of DT states for fidelity-aware query services. Specifically, we first formulated a novel DT state staleness minimization problem and showed its NP-hardness. We then proposed an optimal solution to a special case of the problem where exactly K DT states are synchronized with their objects per update round. Inspired by the solution of this special problem, we also developed an efficient algorithm for the problem by reducing it to the award collection maximization problem, assuming that the volume of the update data of each object since its last synchronization is given. Otherwise, we predicted the volume of the update data of each object by adopting the LSTM method to analyze historical traces of the update data on its DT data. Also, to demonstrate the importance of the DT state staleness of objects, we studied fidelity-aware query services built upon on the DT data, by proposing a cost-effective evaluation algorithm for such query evaluations. Finally, we evaluated the performance of the proposed algorithms through simulations. The simulation results showed that the proposed algorithms are promising. Particularly the average DT state staleness by the proposed algorithms is 44.92% less than the baselines, and the query evaluation cost is 19.42% less than the cost delivered by the baseline algorithm.

Authorized licensed use limited to: CITY UNIV OF HONG KONG. Downloaded on December 12,2024 at 01:07:45 UTC from IEEE Xplore. Restrictions apply.

ACKNOWLEDGMENTS

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

REFERENCES

[1] N. Bansal, A. Blum, S. Chawla, and A. Meyerson, "Approximation algorithms for deadline-TSP and vehicle routing with time-windows," in *Proc. 36th Annu. ACM Symp. Theory Comput.*, 2004, pp. 166–174.

[2] M. Bastopcu and S. Ulukus, "Age of information for updates with distortion: Constant and age-dependent distortion constraints," *IEEE/ACM Trans. Netw.*, vol. 29, no. 6, pp. 2425–2438, Dec. 2021.

[3] L. Corneo, C. Rohner, and P. Gunningberg, "Age of information-aware scheduling for timely and scalable Internet of Things applications," in *Proc. IEEE Conf. Comput. Commun.*, 2019, pp. 2476–2484.

[4] Digital twin market worth 15.66 billion USD by 2023, Aug. 2017. Accessed: Feb. 2022. [Online]. Available: https://www.marketsandmarkets.com/PressReleases/digital-twin.asp

[5] Digital twin | Siemens, 2018. Accessed: Feb. 2022. [Online]. Available: https://www.plm.automation.siemens.com/global/en/our-story/glossary/digital-twin/24465

[6] R. Dong, C. She, W. Hardjawana, Y. Li, and B. Vucetic, "Deep learning for hybrid 5G services in mobile edge computing systems: Learn from a digital twin," *IEEE Trans. Wireless Commun.*, vol. 18, no. 10, pp. 4692–4707, Oct. 2019.

[7] B. Fan, Z. Su, Y. Chen, Y. Wu, C. Xu, and T. Q. S. Quek, "Ubiquitous control over heterogeneous vehicles: A digital twin empowered edge AI approach," *IEEE Wireless Commun.*, vol. 30, no. 1, pp. 166–173, Feb. 2023.

[8] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[9] L. U. Khan, W. Saad, D. Niyato, Z. Han, and C. S. Hong, "Digital-twin-enabled 6G: Vision, architectural trends, and future directions," *IEEE Commun. Mag.*, vol. 60, no. 1, pp. 74–80, Jan. 2022.

[10] G. Lai, W. C. Chang, Y. Yang, and H. Liu, "Modeling long-and short-term temporal patterns with deep neural networks," in *Proc. 41st Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2018, pp. 95–104.

[11] J. Li et al., "AoI-aware user service satisfaction enhancement in digital twin-empowered edge computing," *IEEE/ACM Trans. Netw.*, vol. 32, no. 2, pp. 1677–1690, Apr. 2024.

[12] J. Li et al., "Wait for fresh data? Digital twin empowered IoT services in edge computing," in *Proc 20th Int. Conf. Mobile Ad Hoc Smart Syst.*, 2023, pp. 397–405.

[13] J. Li et al., "AoI-aware, digital twin-empowered IoT query services in mobile edge computing," *IEEE/ACM Trans. Netw.*, vol. 32, no. 4, pp. 3636–3650, Aug. 2024.

[14] J. Li et al., "Digital twin-enabled service provisioning in edge computing via continual learning," *IEEE Trans. Mobile Comput.*, vol. 23, no. 6, pp. 7335–7350, Jun. 2024.

[15] J. Li et al., "Digital twin-assisted, SFC-enabled service provisioning in mobile edge computing," *IEEE Trans. Mobile Comput.*, vol. 23, no. 1, pp. 393–408, Jan. 2024.

[16] 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, 2022, Art. no. 17.

[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. Serv. Comput.*, vol. 16, no. 2, pp. 1417–1430, Mar./Apr. 2023.

[18] J. Li, W. Liang, J. Wang, and X. Jia, "Accumulative fidelity maximization of inference services in DT-assisted edge computing," in *Proc. 1st IEEE Int. Conf. Meta Comput.*, 2024.

[19] J. Li, J. Wang, Q. Chen, Y. Li, and A. Y. Zomaya, "Digital twin-enabled service satisfaction enhancement in edge computing," in *Proc IEEE Conf. Comput. Commun.*, 2023, pp. 1–10.

[20] Y. Li et al., "Data collection maximization in IoT sensor networks via an energy-constrained UAV," *IEEE Trans. Mobile Comput.*, vol. 22, no. 1 pp. 159–174, Jan. 2023.

[21] W. Liang, Z. Xu, W. Xu, J. Shi, G. Mao, and S. Das, "Approximation algorithms for charging reward maximization in rechargeable sensor networks via a mobile charger," *IEEE/ACM Trans. Netw.*, vol. 25, no. 5, pp. 3161–3174, Oct. 2017.

[22] X. Liang, W. Liang, Z. Xu, Y. Zhang, and X. Jia, "Multiple service model refreshments in digital twin-empowered edge computing," *IEEE Trans. Serv. Comput.*, early access, Dec. 12, 2023, doi: 10.1109/TSC.2023.3341988.

[23] H. Liao et al., "Ultra-low AoI digital twin-assisted resource allocation for multi-mode power IoT in distribution grid energy management," *IEEE J. Sel. Areas Commun.*, vol. 41, no. 10, pp. 3122–3132, Oct. 2023.

[24] 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.

[25] Q. Liu, H. Zeng, and M. Chen, "Minimizing AoI with throughput requirements in multi-path network communication," *IEEE/ACM Trans. Netw.*, vol. 30, no. 3, pp. 1203–1216, Jun. 2022.

[26] Y. Lu, X. Huang, K. Zhang, S. Maharjan, and Y. Zhang, "Communication-efficient federated learning and permissioned blockchain for digital twin edge networks," *IEEE Internet Things J.*, vol. 8, no. 4, pp. 2276–2288, Feb. 2021.

[27] Y. Lu, S. Maharjan, and Y. Zhang, "Adaptive edge association for wireless digital twin networks in 6G," *IEEE Internet Things J.*, vol. 8, no. 22, pp. 16219–16230, Nov. 2021.

[28] R. Minerva, G. M. Lee, and N. Crespi, "Digital twin in the IoT context: A survey on technical features, scenarios, and architectural models," *Proc. IEEE*, vol. 108, no. 10, pp. 1785–1824, Oct. 2020.

[29] A. Paul, D. Freund, A. Ferber, D. Shmoys, and D. Williamson, "Prize-collecting TSP with a budget constraint," in *Proc. 25th Annu. Eur. Symp. Algorithms*, Wadern, Germany: Schloss Dagstuhl-Leibniz- Zentrum fuer Informatik, 2017, pp. 62:1–62:14.

[30] C. Singhal and S. De, *Resource Allocation in Next-Generation Broadband Wireless Access Networks*. Hershey, PA, USA: IGI Global Publishers, 2017.

[31] F. B. Sorbelli, A. Navarra, L. Palazzetti, C. M. Pinotti, and G. Prencipe, "Wireless IoT sensors data collection reward maximization by leveraging multiple energy-and storage-constrained UAVs," *J. Comput. Syst. Sci.*, vol. 139, 2024, Art. no. 103475.

[32] 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.

[33] 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.

[34] C. Wang, Z. Cai, and Y. Li, "Sustainable blockchain-based digital twin management architecture for IoT devices," *IEEE Internet Things J.*, vol. 10, no. 8, pp. 6535–6548, Apr. 2023.

[35] X. Wang, Z. Ning, S. Guo, M. Wen, and 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.

[36] Y. Wu, K. Zhang, and Y. Zhang, "Digital twin networks: A survey," *IEEE Internet Things J.*, vol. 8, no. 18, pp. 13789–13804, Sep. 2021.

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

[38] J. Zhang et al., "Minimizing the number of deployed UAVs for delay-bounded data collection of IoT devices," in *Proc. IEEE Conf. Comput. Commun.*, 2021, pp. 1–10.

[39] R. Zhang, Z. Xie, D. Yu, W. Liang, and X. Cheng, "Digital twin-assisted federated learning service provisioning over mobile edge networks," *IEEE Trans. Comput.*, vol. 73, no. 2, pp. 586–598, Feb. 2024.

[40] S. Zhang, L. Wang, H. Luo, X. Ma, and S. Zhou, "AoI-delay trade-off in mobile edge caching with freshness-aware content refreshing," *IEEE Trans. Wireless Commun.*, vol. 20, no. 8, pp. 5329–5342, Aug. 2021.

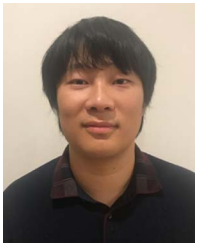
[41] Y. Zhang and W. Liang, "Cost-aware digital twin migration in mobile edge computing via deep reinforcement learning," in *Proc. 23rd IFIP/IEEE Netw. Conf.*, 2024, pp. 441–447.

[42] Y. Zhang, W. Liang, Z. Xu, and X. Jia, "Mobility-aware service provisioning in edge computing via digital twin replica placements," *IEEE Trans. Mobile Comput.*, early access, Apr. 30, 2024, doi: 10.1109/TMC.2024.3394839.

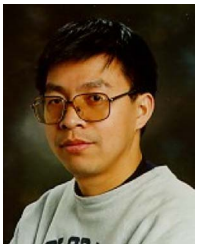
[43] Y. Zhang, W. Liang, Z. Xu, W. Xu, and M. Chen, "AoI-aware inference services in edge computing via digital twin network slicing," *IEEE Trans. Serv. Comput.*, early access, Aug. 02, 2024, doi: 10.1109/TSC.2024.3436705.

Authorized licensed use limited to: CITY UNIV OF HONG KONG. Downloaded on December 12,2024 at 01:07:45 UTC from IEEE Xplore. Restrictions apply.

[44] Y. Zhang, W. Liang, W. Xu, Z. Xu, and X. Jia, "Cost minimization of digital twin placements in mobile edge computing," *ACM Trans. Sensor Netw.*, vol. 20, no. 3, pp. 1–26, 2024.



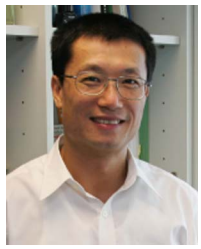
Yuchen Li received the BSc and PhD degrees in computer science from the Australian National University, in 2018 and 2023, respectively. His research interests include the Internet of Things, mobile edge computing, and algorithm design.



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

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

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



Xiaohua Jia (Fellow, IEEE) received the BSc and ME degrees from the University of Science and Technology of China, in 1984 and 1987, respectively, and the DSc degree in information science from the University of Tokyo, in 1991. He is currently a chair professor with the Department of Computer Science, City University of Hong Kong. His research interests include cloud computing and distributed systems, computer networks, wireless sensor networks, and mobile wireless networks. He is an editor of *IEEE Transactions on Parallel and Distributed Systems* (2006–2009), *Journal of World Wide Web*, *Wireless Networks*, *Journal of Combinatorial Optimization*, and so on. He is the general chair of ACM MobiHoc 2008, TPC co-chair of IEEE MASS 2009, area-chair of IEEE INFOCOM 2010, TPC co-chair of IEEE GlobeCom 2010, Ad Hoc and Sensor Networking Symposium, and Panel co-chair of IEEE INFOCOM 2011.

Authorized licensed use limited to: CITY UNIV OF HONG KONG. Downloaded on December 12,2024 at 01:07:45 UTC from IEEE Xplore. Restrictions apply.