Fractional Deep Reinforcement Learning for Age-Minimal Mobile Edge Computing

Lyudong Jin*[1], Ming Tang*[2], Meng Zhang†[1], Hao Wang[3]
[1]Zhejiang University, Haining, Zhejiang China
[2]Southern University of Science and Technology, Shenzhen, Guangdong China
[3]Monash University, Melbourne, Victoria Australia
lvdong.22@intl.zju.edu.cn,tangm3@sustech.edu.cn, mengzhang@intl.zju.edu.cn, hao.wang2@monash.edu

Abstract

Mobile edge computing (MEC) is a promising paradigm for real-time applications with intensive computational needs (e.g., autonomous driving), as it can reduce the processing delay. In this work, we focus on the timeliness of computational-intensive updates, measured by *Age-of-Information (AoI)*, and study how to jointly optimize the task updating and offloading policies for AoI with fractional form. Specifically, we consider edge load dynamics and formulate a task scheduling problem to minimize the expected time-average AoI. The uncertain edge load dynamics, the nature of the fractional objective, and hybrid continuous-discrete action space (due to the joint optimization) make this problem challenging and existing approaches not directly applicable. To this end, we propose a fractional reinforcement learning (RL) framework and prove its convergence. We further design a model-free fractional deep RL (DRL) algorithm, where each device makes scheduling decisions with the hybrid action space without knowing the system dynamics and decisions of other devices. Experimental results show that our proposed algorithms reduce the average AoI by up to 57.6% compared with several non-fractional benchmarks.

Introduction

Background and Motivations

The next-generation network demands mobile devices (e.g., smartphones and Internet-of-Things devices) to generate zillions of bytes of data and accomplish unprecedentedly computationally intensive tasks. Mobile devices, however, will be unable to timely process all their tasks locally due to their limited computational resources. To fulfill the low latency requirement, mobile edge computing (Mao et al. 2017) (MEC), also known as multi-access edge computing (Porambage et al. 2018), has become an emerging paradigm distributing computational tasks and services from the network core to the network edge. By enabling mobile devices to offload their computational tasks to nearby edge nodes, MEC can reduce the task processing delay.

On the other hand, the proliferation of real-time and computation-intensive applications (e.g., cyber-physical systems) has significantly boosted the demand for information freshness (Yates et al. 2021; Kaul, Yates, and Gruteser 2012; Shisher and Sun 2022; Pan et al. 2023; Pan, Sun, and Shroff 2023), in addition to low latency. For example, the real-time velocity and location knowledge of the surrounding vehicles is crucial in achieving safe and efficient autonomous driving. Another emerging example is metaverse applications, in which users anticipate real-time virtual reality services and real-time control over their avatars. In these applications, users' experience depends on how fresh the received information is rather than how long it takes to receive that information. Such a requirement motivates a new network performance metric, namely *Age of Information (AoI)* (Yates et al. 2021; Kaul, Yates, and Gruteser 2012; Shisher and Sun 2022). It measures the time elapsed since the most up-to-date data (computational results) was received.

While the majority of existing studies on MEC were concerned about delay reduction (e.g., (Wang et al. 2021; Tang and Wong 2022)), most of real-time applications mentioned above concern about fresh status updates, while delay itself does not directly reflect timeliness. Here we highlight the huge difference between delay and AoI. Specifically, task delay takes into account only the duration between when the task is generated and when the task output has been received by the mobile device. Thus, under less frequent updates (i.e., when tasks are generated in a lower frequency), task delays are naturally smaller. This is because infrequent updates lead to empty queues and hence reduced queuing delays of the tasks. In contrast, AoI takes into account both the task delay and the freshness of the task output. Thus, to minimize the AoI with computational-intensive tasks, the update frequency needs to be neither too high nor too low in order to reduce the delay of each task while ensuring the freshness of the most up-to-date task output. More importantly, such a difference between delay and AoI leads to a counter-intuitive important phenomenon in designing age-minimal scheduling policy: upon the reception of each update, the mobile device may need to *wait* for a certain amount of time to generate the next new task (Sun et al. 2017).

Therefore, the age-minimal MEC systems necessitate meticulous design of a scheduling policy for each mobile device, which should encompass two fundamental decisions. The first decision is the *updating* decision, i.e., upon completion of a task, how long should a mobile device wait for

*These authors contributed equally.
†Corresponding author.

generating the next one. The second is the task *offloading* decision, i.e., whether to offload the task or not? If yes, which edge node to choose? Although existing works on MEC have addressed the task offloading decision (e.g., (Tang and Wong 2022; Ma et al. 2022; Zhao et al. 2022; Zhu et al. 2022; He et al. 2022; Chen et al. 2022)) and some studies considered AoI (e.g., (Zhu et al. 2022; He et al. 2022; Chen et al. 2022)), they did not consider designing the task updating policy to improve data timeliness.

What's more, the age-minimal technique can also be extended to other ratio optimization problems including financial portfolio optimization (Keating and Shadwick 2002) and energy efficiency (EE) maximization problems for wireless communications (Omidkar et al. 2022).

In this paper, we aim to answer the following question:

Key Question. *How should mobile devices optimize their updating and offloading policies of hybrid action space in dynamic MEC systems in order to minimize their fractional objectives of AoI?*

Solution Approach and Contributions

In this work, we take into account system dynamics in MEC systems and aim at designing distributed AoI-minimal DRL algorithms to jointly optimize task updating and offloading. We first propose a novel fractional RL framework, incorporating reinforcement learning techniques and Dinkelbach's approach (for fractional programming) in (Dinkelbach 1967). We further propose a fractional Q-Learning algorithm and analyze its convergence. To address the hybrid action space, we further design a fractional DRL-based algorithm. Our main contributions are summarized as follows:

- *Joint Task Updating and Offloading Problem:* We formulate the joint task updating and offloading problem that takes into account unknown system dynamics. *To the best of our knowledge, this is the first work designing the joint updating and offloading policy for age-minimal MEC.*
- *Fractional RL Framework:* To overcome fractional objective of the average AoI, we propose a novel fractional RL framework. We further propose a fractional Q-Learning algorithm. We design a stopping condition, leading to a provable linear convergence rate without the need of increasing inner-loop steps.
- *Fractional DRL Algorithm:* We overcome unknown dynamics and hybrid action space of offloading and updating decisions and propose a fractional DRL-based distributed scheduling algorithm for age-minimal MEC, which extends the dueling double deep Q-network (D3QN) and deep deterministic policy gradient (DDPG) techniques into our proposed fractional RL framework.
- *Performance Evaluation:* Our algorithm significantly outperforms the benchmarks that neglect the fractional nature with an average AoI reduction by up to 57.6%. In addition, the joint optimization of offloading and updating can further reduce the AoI by up to 31.3%.

Literature Review

Mobile Edge Computing: Existing excellent works have conducted various research questions in MEC, including resource allocation (e.g., (Wang et al. 2022b)), service placement (e.g., (Taka, He, and Oki 2022)), and proactive caching (e.g., (Liu et al. 2022a)). Task offloading (Wang, Ye, and Lui 2022; Ma et al. 2022; Chen and Xie 2022), as another main research question in MEC, attracting considerable attention. To address the unknown system dynamics and reduce task delay, many existing works have proposed DRL-based approaches to optimize the task offloading in a centralized manner (e.g., (Huang, Bi, and Zhang 2020; Tuli et al. 2022)). As in our work, some existing works have proposed distributed DRL-based algorithms (e.g., (Tang and Wong 2022; Liu et al. 2022b; Zhao et al. 2022)) which do not require the global information. *Despite the success of these works in reducing the task delay, these approaches are NOT easily applicable to age-minimal MEC due to the aforementioned challenges of fractional objective and hybrid action space.*

Age of Information: Kaul *et al.* first introduced AoI in (Kaul, Yates, and Gruteser 2012). Assuming complete and known statistical information, the majority of this line of work mainly focused on the optimization and analysis of AoI in queueing systems and wireless networks (see (Zou, Ozel, and Subramaniam 2021; Chiariotti et al. 2021; Kuang et al. 2020), and a survey in (Yates et al. 2021)). Zou *et al.* in (Zou, Ozel, and Subramaniam 2021), Zhou *et al.* in (M. Zhou and Yates 2024, Early Access), Chiariotti *et al.* in (Chiariotti et al. 2021), and Kuang *et al.* in (Kuang et al. 2020). *The above studies analyzed simple single-device-single-server models and hence did not consider offloading.*

A few studies investigated DRL algorithm design to minimize AoI in various application scenarios, including wireless networks (e.g., (Ceran, Gündüz, and György 2021)), Internet-of-Things (e.g., (Akbari et al. 2021; Wang et al. 2022a)), vehicular networks (e.g., (Chen et al. 2020)), and UAV-aided networks (e.g., (Hu et al. 2020; Wu et al. 2021)). *This line of work mainly focused on optimal resource allocation and trajectory design.* Existing works considered AoI as the performance metric for task offloading in MEC and proposed DRL-based approaches to address the AoI minimization problem. Chen *et al.* in (Chen et al. 2022) considered AoI to capture the freshness of computation outcomes and proposed a multi-agent DRL algorithm. *However, these works focused on designing task offloading policy but did not optimize updating policy. Most importantly, all aforementioned approaches did not account for fractional RL and hence cannot directly tackle our considered problem.*

RL with Fractional Objectives: Research on RL with fractional objectives is currently limited. Ren *et. al.* introduced fractional MDP (Ren and Krogh 2005). Reference (Tanaka 2017) further studied partially observed MDPs with fractional rewards. However, RL methods were not considered in these studies. Suttle *et al.* (Suttle et al. 2021) proposed a two-timescale RL algorithm for the fractional cost, but it requires additional fixed reference states in the Q-learning update process to approximate the outer loop update and leave finite-time convergence analysis unsettled.

System Model

Consider M mobile devices and N edge nodes, which are in set $\mathcal{M} = \{1, 2, \ldots, M\}$ and set $\mathcal{N} = \{1, 2, \ldots N\}$, re-

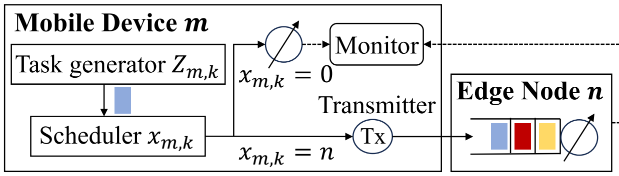

Figure 1: Illustration of an MEC system with mobile device $m \in \mathcal{M}$ and edge node $n \in \mathcal{N}$ where the tasks of different mobile devices are represented with different colors.

spectively. We consider an infinite-horizon continuous-time system model illustrated in Fig. 1.

Device Model

Each mobile device $m \in \mathcal{M}$ has a task generator, a scheduler, and a monitor. The task generator generates new tasks for processing while the scheduler determines where to process the tasks and the task output is sent to the monitor. We refer to a task output received by the monitor as an *update*.

Task Generator: We consider a *generate-at-will* model, as in (Sun et al. 2017), i.e., each task generator can decide when to generate the next new task. At the time when task $k-1$ of mobile device $m \in \mathcal{M}$ has been completed (denoted by time $t'_{m,k-1}$), the task generator observes the task delay $Y_{m,k-1}$ and makes a decision on $Z_{m,k}$, i.e., the waiting time for generating the *next* task k. Let $\boldsymbol{s}_m^{\text{U}}(k)$ and $\boldsymbol{a}_m^{\text{U}}(k)$ denote the state and action with task k of mobile device m respectively: $\boldsymbol{s}_m^{\text{U}}(k) = Y_{m,k-1},\ \boldsymbol{a}_m^{\text{U}}(k) = Z_{m,k},\ k \in \mathcal{K}, m \in \mathcal{M}$.

Let $\mathcal{S}^{\text{U}} = (0, \bar{Y}]$ and $\mathcal{A}^{\text{U}} \in [0, \bar{Z}]$ denote the state and action space, respectively. Let $\pi_m^{\text{U}} : \mathcal{S}^{\text{U}} \to \mathcal{A}^{\text{U}}$ denote the updating policy of mobile device $m \in \mathcal{M}$ that maps from $\mathcal{S}^{\text{U}}$ to $\mathcal{A}^{\text{U}}$. Specifically, let $t_{m,k}$ denote the time stamp when the task generator of mobile device m generates task k, after which the task is sent to the scheduler. The transmission time is considered negligible and $t_{m,k+1} = t'_{m,k} + Z_{m,k+1}$. From (Sun et al. 2017), the optimal waiting strategy may outperform the zero-wait policy, i.e., $Z_{m,k}$ may not necessarily be zero and requires proper optimization.

Scheduler: At the time when task k of mobile device $m \in \mathcal{M}$ is generated (i.e., time $t_{m,k}$), the task scheduler observes the queue lengths of edge nodes and makes the offloading decision denoted by $x_{m,k} \in \{0\} \cup \mathcal{N}$. Let $\boldsymbol{s}_m^{\text{O}}(k)$ denote the state vector associated with task k of mobile device m: $\boldsymbol{s}_m^{\text{O}}(k) = \boldsymbol{q}(t_{m,k}),\ k \in \mathcal{K}, m \in \mathcal{M}$, where $\boldsymbol{q}(t_{m,k}) = (q_n(t_{m,k}), n \in \mathcal{N})$ corresponds to the queue lengths of all edge nodes. We assume that edge nodes send their queue length information upon the requests of mobile devices. Since a generator generates a new task only after the previous task has been processed, the queue length is less than or equal to M. Thus, it can be encoded in $O(\log_2 M)$ bits, which incurs only small signaling overheads. Let $\mathcal{S}^{\text{O}} = \mathcal{M}^{1\times N}$ denote the state space. Let $\boldsymbol{a}_m^{\text{O}}(k)$ denote the action associated with task k of mobile device m. Thus, $\boldsymbol{a}_m^{\text{O}}(k) = x_{m,k},\ \ k \in \mathcal{K}, m \in \mathcal{M}$. Let $\mathcal{A}^{\text{O}} \in \{0\} \cup \mathcal{N}$ denote the offloading action space. Let π_m^{O} denote the task offloading policy of mobile device $m \in \mathcal{M}$.

If mobile device m processes task k locally, then let $\tau_{m,k}^{\text{local}}$ (in seconds) denote the service time of mobile device $m \in \mathcal{M}$ for processing task k. The value of $\tau_{m,k}^{\text{local}}$ depends on the size of task k and the real-time processing capacity of mobile device m (e.g., whether the device is busy in processing tasks of other applications), which are unknown *a priori*. If mobile device m offloads task k to edge node $n \in \mathcal{N}$, then let $\tau_{n,m,k}^{\text{tran}}$ (in seconds) denote the service time of mobile device $m \in \mathcal{M}$ for sending task k to edge node n. The value of $\tau_{n,m,k}^{\text{tran}}$ depends on the time-varying wireless channels and is unknown *a priori*. We assume that $\tau_{m,k}^{\text{local}}$ and $\tau_{n,m,k}^{\text{tran}}$ are random variables that follow exponential distribution (Tang et al. 2021; Liu et al. 2022b; Zhu and Gong 2022), respectively, which are independent of edge loads.

Edge Node Model

Upon receiving a task offloaded by a mobile device, edge node $n \in \mathcal{N}$ enqueues the task for processing. The queue may store the tasks offloaded by multiple mobile devices, as shown in Fig. 1. Suppose the queue operates in a first-in-first-out (FIFO) manner (Yates et al. 2021).

Let $w_{n,m,k}^{\text{edge}}$ (in seconds) denote the time duration that task k of mobile device $m \in \mathcal{M}$ waits at the queue of edge node n. Let $\tau_{n,m,k}^{\text{edge}}$ (in seconds) denote the service time of edge node n for processing task k of mobile device m. The value of $\tau_{n,m,k}^{\text{edge}}$ depends on the size of task k and may be unknown *a priori*. We assume that $\tau_{n,m,k}^{\text{edge}}$ is a random variable following exponential distribution as well. In addition, the value of $w_{n,m,k}^{\text{edge}}$ depends on the processing time of the tasks placed in the queue (of edge node n) ahead of the task k of device m, where those tasks are possibly offloaded by mobile devices other than device m. Thus, since mobile device m does not know the offloading behaviors of other mobile devices *a priori*, it does not know the value of $w_{n,m,k}^{\text{edge}}$ beforehand.

Age of Information

The *age of information (AoI)* for mobile device m at time stamp t (Yates et al. 2021) is given by

$$\Delta_m(t) = t - U_m(t),\ \ \forall m \in \mathcal{M}, t \geq 0, \tag{1}$$

where $U_m(t) \triangleq \max_k[t_{m,k}|t'_{m,k} \leq t]$ stands for the time stamp of the most recently completed task.

We use $Y_{m,k} \triangleq t'_{m,k} - t_{m,k}$ to denote the delay of task k, i.e., the time it takes to complete task k. Thus,

$$Y_{m,k} = \begin{cases} \tau_{m,k}^{\text{local}}, & x_{m,k} = 0, \\ \tau_{n,m,k}^{\text{tran}} + w_{n,m,k}^{\text{edge}} + \tau_{n,m,k}^{\text{tran}}, & x_{m,k} = n \in \mathcal{N}. \end{cases} \tag{2}$$

We consider a *drop time* $\bar{Y}$ (in seconds). That is, we assume that if a task has not been completely processed within $\bar{Y}$ seconds, the task will be dropped (Tang and Wong 2022; Li, Zhou, and Chen 2020). Meantime, the AoI keeps increasing until the next task is completed.

To capture the overall performance of mobile device m, we define the trapezoid area associated with time interval

$[t_{m,k}, t_{m,k+1})$ (Yates et al. 2021):

$$A(Y_{m,k}, Z_{m,k+1}, Y_{m,k+1}) \triangleq \frac{1}{2}(Y_{m,k} + Z_{m,k+1} + Y_{m,k+1})^2 - \frac{1}{2}Y_{m,k+1}^2. \quad (3)$$

Based on (3), we can characterize the objective of mobile device m, i.e., to minimize the time-average AoI of each device $m \in \mathcal{M}$: (Yates et al. 2021)

$$\Delta_m^{(ave)} \triangleq \liminf_{M\to\infty} \frac{\sum_{m=1}^{M} A(Y_{m,k}, Z_{m,k+1}, Y_{m,k+1})}{\sum_{m=1}^{M}(Y_{m,k} + Z_{m,k+1})}. \quad (4)$$

Problem Formulation

Let $\boldsymbol{\pi}_m = (\pi_m^{\mathrm{U}}, \pi_m^{\mathrm{O}})$ denote the policy of mobile device $m \in \mathcal{M}$. This is a stationary policy that contains the mapping from $\mathcal{S}^{\mathrm{U}} \times \mathcal{S}^{\mathrm{O}}$ to $\mathcal{A}^{\mathrm{U}} \times \mathcal{A}^{\mathrm{O}}$. Given a stationary policy $\boldsymbol{\pi}_m$, the expected time-average AoI of mobile device $m \in \mathcal{M}$ is

$$\mathbb{E}[\Delta_m^{(ave)}|\boldsymbol{\pi}_m] \triangleq \frac{\mathbb{E}[A(Y_{m,k}, Z_{m,k+1}, Y_{m,k+1})|\boldsymbol{\pi}_m]}{\mathbb{E}[Y_{m,k} + Z_{m,k+1}|\boldsymbol{\pi}_m]}. \quad (5)$$

We take the expectation $\mathbb{E}[\cdot]$ over policy $\boldsymbol{\pi}_m$ and the time-varying system parameters, e.g., the time-varying processing duration as well as the edge load dynamics.

We aim at the optimal policy $\boldsymbol{\pi}_m^*$ for each mobile device $m \in \mathcal{M}$ to minimize its expected time-average AoI.

$$\boldsymbol{\pi}_m^* = \arg\underset{\boldsymbol{\pi}_m}{\text{minimize}} \quad \mathbb{E}\left[\Delta_m^{(ave)}|\boldsymbol{\pi}_m\right]. \quad (6)$$

The fractional objective in (5) introduces a major challenge in designing the optimal policy, which is significantly different from conventional RL and DRL algorithms. Specifically, the difficulty of directly expressing the immediate reward (cost) of each action for the fractional RL problem. Specifically, it seems to be straightforward to define the reward (or cost) function as either the instant AoI (i.e., $\Delta_m(t'_{m,k})$) (Chen et al. 2022; He et al. 2022) or the average AoI during certain time interval (e.g., $[t_{m,k}, t_{m,k+1})$). However, consider the time-average over infinite time horizon, neither minimizing $\mathbb{E}[\Delta_m(t'_{m,k})|\boldsymbol{\pi}_m]$ nor $\mathbb{E}[A(Y_{m,k}, Z_{m,k+1}, Y_{m,k+1})/(Y_{m,k} + Z_{m,k+1})|\boldsymbol{\pi}_m]$ is equivalent to minimizing (5).

Fractional RL Framework

In this section, we propose a fractional RL framework for solving Problem (6).We first present a two-step reformulation of Problem (6). We then introduce the fractional RL framework, under which we present a fractional Q-Learning algorithm with provable convergence guarantees.

Dinkelbach's Reformulation With the proposed Problem 6 we consider the Dinkelbach's reformulation and a discounted reformulation in the following. we define a reformulated AoI in an average-cost fashion:

$$\mathbb{E}[\Delta'_m|\boldsymbol{\pi}_m, \gamma] \triangleq \lim_{K\to\infty} \frac{1}{K}\sum_{k=1}^{K}\{\mathbb{E}[A(Y_{m,k}, Z_{m,k+1}, Y_{m,k+1})|\boldsymbol{\pi}_m] - \gamma\mathbb{E}[Y_{m,k} + Z_{m,k+1}|\boldsymbol{\pi}_m]\}. \quad (7)$$

Let γ^* be the optimal value of Problem (6). Leveraging Dinkelbach's method (Dinkelbach 1967), we have the following reformulated problem:

Lemma 1 ((Dinkelbach 1967))**.** *Problem* (6) *is equivalent to the following reformulated problem:*

$$\boldsymbol{\pi}_m^* = \arg\underset{\boldsymbol{\pi}_m}{\text{minimize}} \ \mathbb{E}[\Delta'_m|\boldsymbol{\pi}_m, \gamma^*], \quad \forall m \in \mathcal{M}, \quad (8)$$

where $\boldsymbol{\pi}_m^*$ is the optimal solution to Problem (6)*.*

Since $\mathbb{E}[\Delta'_m|\boldsymbol{\pi}_m, \gamma^*] \geq 0$ for any π and $\mathbb{E}[\Delta'_m|\boldsymbol{\pi}_m^*, \gamma^*] = 0$, $\boldsymbol{\pi}_m^*$ is also optimal to the Dinkelbach reformulation. This implies the reformulation equivalence is also established for our stationary policy space.

Discounted Reformulation Following Dinkelbach's reformulation, we reformulate the problem in (8) one step further by considering a discounted objective. Let $\delta \in (0, 1]$ be the discount factor, capturing how the objective is discounted in the future. We define

$$\mathbb{E}[\Delta_m^{\delta}|\boldsymbol{\pi}_m, \gamma] \triangleq \sum_{k=1}^{\infty} \delta^k \{\mathbb{E}[A(Y_{m,k}, Z_{m,k+1}, Y_{m,k+1})|\boldsymbol{\pi}_m] - \gamma\mathbb{E}[Y_{m,k} + Z_{m,k+1}|\boldsymbol{\pi}_m]\}, \forall\gamma \geq 0. \quad (9)$$

From (Puterman 2014), we can establish the asymptotic equivalence between the average formulation and the discounted formulation:

Lemma 2 (Asymptotic Equivalence (Puterman 2014))**.** *Given the optimal quotient value γ^*, Problems* (6) *and* (8) *are asymptotically equivalent to reformulation as follows:*

$$\boldsymbol{\pi}_m^* = \arg\underset{\boldsymbol{\pi}_m}{\text{minimize}} \ \lim_{\delta\to 1} \mathbb{E}[\Delta_m^{\delta}|\boldsymbol{\pi}_m, \gamma^*], \quad (10)$$

for all $m \in \mathcal{M}$, where $\boldsymbol{\pi}_m^*$ is the optimal solution to (8)*.*

Therefore, the discounted reformulation in (10) serves as a good approximation of (7) when δ approaches 1. Such an approximation provides us with a convention of designing new DRL algorithms for fractional MDP problems based on existing well-established DRL algorithms. We will stick to the discounted reformulation for the rest of this paper.

Fractional MDP

We study the following general fractional MDP framework and drop index m for the rest of this section.

Definition 1 (Fractional MDP)**.** *A fractional MDP is defined as $(\mathcal{S}, \mathcal{A}, P, c_N, c_D, \delta)$, where $\mathcal{S}$ and $\mathcal{A}$ are the finite sets of states and actions, respectively; P is the transition distribution; c_N and c_D are the cost functions*[1]*, and δ is a discount factor. We use $\mathcal{Z}$ to denote the joint state-action space, i.e., $\mathcal{Z} \triangleq \mathcal{S} \times \mathcal{A}$.*

From Definition 1 and Lemmas 1 and 2, we have that Problem (6) has the equivalent Dinkelbach's reformulation:

$$\boldsymbol{\pi}^* = \arg\underset{\boldsymbol{\pi}}{\text{minimize}} \lim_{K\to\infty} \mathbb{E}\left[\sum_{k=1}^{K} \delta^k (c_N - \gamma^* c_D)\middle| \boldsymbol{\pi}\right], \quad (11)$$

[1]Since we aim at minimizing the time-average AoI, we consider minimizing long-term expected cost in this work.

Algorithm 1: Fractional Q-Learning (FQL)

for $k = 1, 2, \ldots, K$ **do**
  Initialize $\boldsymbol{s}_m(1)$;
  for time slot $t \in \mathcal{T}$ **do**
    Observe the next state $\boldsymbol{s}_m(t+1)$;
    Observe a set of costs $\{c_m(t'),\ t' \in \widetilde{\mathcal{T}}_{m,t}\}$;
    for each task $k_m(t')$ with $t' \in \widetilde{\mathcal{T}}_{m,t}$ **do**
      Send $(\boldsymbol{s}_m(t'), \boldsymbol{a}_m(t'), c_m(t'), \boldsymbol{s}_m(t'+1))$ to n_m;
    end for
  end for
  $\gamma^{(k+1)} = N_{\gamma^k}(s, a_k)/D_{\gamma^k}(s, a_k)$, where $a_k = \arg\max_a Q_\gamma^T(s, a)$.
end for

where we can see from Lemmas 1 and 2 that γ^* satisfies

$$\gamma^* = \underset{\boldsymbol{\pi}}{\text{minimize}} \lim_{K\to\infty} \frac{\mathbb{E}\left[\sum_{k=0}^{K} \delta^k c_N \middle| \boldsymbol{\pi}\right]}{\mathbb{E}\left[\sum_{k=0}^{K} \delta^k c_D \middle| \boldsymbol{\pi}\right]}, \tag{12}$$

Note that Problem (11) is a classical MDP problem, including an immediate cost, given by $c_N(\boldsymbol{s}, \boldsymbol{a}) - \gamma^* c_D(\boldsymbol{s}, \boldsymbol{a})$. Thus, we can then apply a traditional RL algorithm to solve such a reformulated problem, such as Q-Learning or its variants (e.g., SQL in (Ghavamzadeh et al. 2011)).

However, the optimal quotient coefficient γ^* and the transition distribution P are unknown *a priori*. Therefore, one needs to design an algorithm that combines both fractional programming and RL algorithms to solve Problem (11) for a given γ and seek the value of γ^*. To achieve this, we start by introducing the following definitions: Given a quotient coefficient γ, the optimal Q-function is

$$Q_\gamma^*(\boldsymbol{s}, \boldsymbol{a}) \triangleq \min_{\boldsymbol{\pi}} Q_\gamma^{\boldsymbol{\pi}}(\boldsymbol{s}, \boldsymbol{a}),\ \forall (\boldsymbol{s}, \boldsymbol{a}) \in \mathcal{Z}, \tag{13}$$

where $Q_\gamma^{\boldsymbol{\pi}}(\boldsymbol{s}, \boldsymbol{a})$ is the action-state function that satisfies the following Bellman's equation: for all $(\boldsymbol{s}, \boldsymbol{a}) \in \mathcal{Z}$,

$$\begin{aligned} Q_\gamma^{\boldsymbol{\pi}}(\boldsymbol{s}, \boldsymbol{a}) \triangleq & c_N(\boldsymbol{s}, \boldsymbol{a}) - \gamma c_D(\boldsymbol{s}, \boldsymbol{a}) \\ & + \delta \mathbb{E}[Q_\gamma^{\boldsymbol{\pi}}(\boldsymbol{s'}, \boldsymbol{a'}) | \boldsymbol{s}, \boldsymbol{a}, \boldsymbol{\pi}]. \end{aligned} \tag{14}$$

In addition, we can further decompose the optimal Q-function in (13) into the following two parts: $Q_\gamma^*(\boldsymbol{s}, \boldsymbol{a}) = N_\gamma(\boldsymbol{s}, \boldsymbol{a}) - \gamma D_\gamma(\boldsymbol{s}, \boldsymbol{a})$ and, for all $(\boldsymbol{s}, \boldsymbol{a}) \in \mathcal{Z}$,

$$N_\gamma(\boldsymbol{s}, \boldsymbol{a}) = c_N(\boldsymbol{s}, \boldsymbol{a}) + \delta \mathbb{E}[N_\gamma(\boldsymbol{s'}, \boldsymbol{a'}) | \boldsymbol{s}, \boldsymbol{a}, \boldsymbol{\pi}^*], \tag{15}$$

$$D_\gamma(\boldsymbol{s}, \boldsymbol{a}) = c_D(\boldsymbol{s}, \boldsymbol{a}) + \delta \mathbb{E}[D_\gamma(\boldsymbol{s'}, \boldsymbol{a'}) | \boldsymbol{s}, \boldsymbol{a}, \boldsymbol{\pi}^*]. \tag{16}$$

Fractional Q-Learning Algorithm

In this subsection, we present a Fractional Q-Learning (FQL) algorithm in Algorithm 1, consisting of an inner loop with E episodes and an outer loop. The key idea is to approximate the Q-function $Q_{\gamma_i}^*$ by Q_i and then iterate $\{\gamma_i\}$.

One of the key innovations in Algorithm 1 is the design of the stopping condition, leading to the shrinking values of the uniform approximation errors of Q_i. This facilitates us to adapt the convergence proof in (Dinkelbach 1967) to our setting and prove the linear convergence rate of $\{\gamma_i\}$ without increasing the inner-loop time complexity.

We describe the details of the inner loop and the outer loop procedures of Algorithm 1 in the following:

- *Inner loop*: For each episode i, given a quotient coefficient γ_i, we perform an (arbitrary) Q-Learning algorithm (as the Speedy Q-Learning in (Ghavamzadeh et al. 2011)) to approximate function $Q_{\gamma_i}^*(\boldsymbol{s}, \boldsymbol{a})$ by $Q_i(\boldsymbol{s}, \boldsymbol{a})$. Let $\boldsymbol{s}_0$ denote the initial state of any arbitrary episode, and $\boldsymbol{a}_i \triangleq \arg\max_{\boldsymbol{a}} Q_i(\boldsymbol{s}_0, \boldsymbol{a})$ for all $i \in [E] \triangleq \{1, ..., E\}$. We consider a *stopping condition*

$$\epsilon_i < -\alpha Q_i(\boldsymbol{s}_0, \boldsymbol{a}_i),\ \forall i \in [E], \tag{17}$$

  so as to terminate each episode i with a bounded *uniform approximation error*: $\|Q_{\gamma_i}^* - Q_i\| \leq \epsilon_i,\ \forall i \in [E]$. Operator $\|\cdot\|$ is the supremum norm, which satisfies $\|g\| \triangleq \max_{(\boldsymbol{s}, \boldsymbol{a}) \in \mathcal{Z}} g(\boldsymbol{s}, \boldsymbol{a})$. Specifically, we obtain $Q_i(\boldsymbol{s}, \boldsymbol{a})$, $N_i(\boldsymbol{s}, \boldsymbol{a})$, and $D_i(\boldsymbol{s}, \boldsymbol{a})$, which satisfy, for all $(\boldsymbol{s}, \boldsymbol{a}) \in \mathcal{Z}$,

$$Q_i(\boldsymbol{s}, \boldsymbol{a}) = N_i(\boldsymbol{s}, \boldsymbol{a}) - \gamma_i D_i(\boldsymbol{s}, \boldsymbol{a}). \tag{18}$$

- *the outer loop to update the quotient coefficient:*

$$\gamma_{i+1} = \frac{N_i(\boldsymbol{s}_0, \boldsymbol{a}_i)}{D_i(\boldsymbol{s}_0, \boldsymbol{a}_i)}, \quad \forall i \in [E], \tag{19}$$

  which will be shown to converge to the optimal value γ^*.

Convergence Analysis

We are ready to present the key convergence results of our proposed FQL algorithm (Algorithm 1) as follows.

Convergence of the outer loop We start with analyzing the convergence of the outer loop:

Theorem 1 (Linear Convergence of Fractional Q-Learning)**.** *If we select $\{T_i\}$ such that the uniform approximation error $\|Q_{\gamma_i}^* - Q_i\| \leq \epsilon_i$ holds with $\epsilon_i < -\alpha Q_i(\boldsymbol{s}_0, \boldsymbol{a}_i)$ for some $\alpha \in (0, 1)$ and for all $i \in [E]$, then the sequence $\{\gamma_i\}$ generated by Algorithm 1 satisfies*

$$\frac{\gamma_{i+1} - \gamma^*}{\gamma_i - \gamma^*} \in (0, 1),\ \forall i \in [E] \text{ and } \lim_{i\to\infty} \frac{\gamma_{i+1} - \gamma^*}{\gamma_i - \gamma^*} = \alpha. \tag{20}$$

That is, $\{\gamma_i\}$ converges to γ^* linearly.

While the convergence proof in (Dinkelbach 1967) requires to obtain the *exact* solution in each episode, Theorem 1 generalizes this result to the case where we only obtain an *approximated (inexact)* solution in each episode. In addition to the proof techniques in (Dinkelbach 1967) and (Ghavamzadeh et al. 2011), our proof techniques include induction and exploiting the convexity of $Q_i(\boldsymbol{s}, \boldsymbol{a})$. We present a proof sketch of Theorem 1 in Appendix A[2].

The significance of Theorem 1 is two-fold. First, Theorem 1 shows that Algorithm 1 achieves a linear convergence rate, even though it only attains an approximation of $Q_\gamma^*(\boldsymbol{s}, \boldsymbol{a})$. Second, (17) is a well-behaved stopping condition.

[2]Please refer to (Jin et al. 2023) for our appendices.

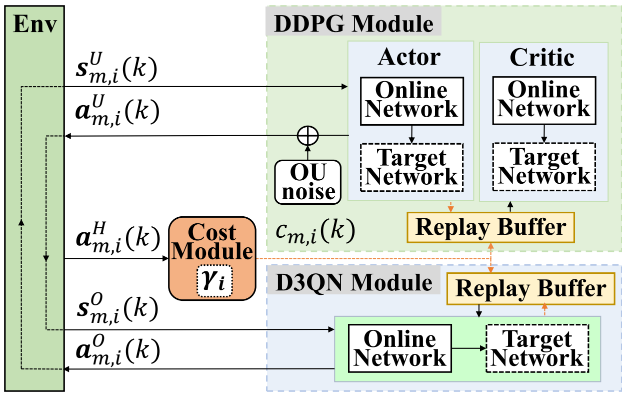

Figure 2: Illustration on the fractional DRL framework.

Time Complexity of the inner loop Although as $\{Q_i(\boldsymbol{s}_0, \boldsymbol{a}_i)\}$ is convergent to 0 and hence $\epsilon_i < -\alpha Q_i$ is getting more restrictive as i increases, the steps needed T_i in Algorithm 1 keep to be finite without increasing over episode i. See Appendix B in detail.

Fractional DRL Algorithm

In this section, we present a fractional DRL-based algorithm to approximate the Q-function in FQL algorithm and solve Problem (6) with DDPG (for continuous action space) (Lillicrap et al. 2015) and D3QN (for discrete action space) (Mnih et al. 2015) techniques for the task updating and offloading processes in the decentralized manner, which is illustrated in Fig. 2. Appendix C shows detailed settings of networks. Moreover, we design a cost function based on our fractional RL framework to ensure the convergence.

Cost Module

As in the proposed fractional RL framework, we consider a set of episodes $i \in [E]$ and introduce a quotient coefficient γ_i for episode i. Consider mobile device m. Let $\boldsymbol{a}^{\mathrm{H}}_{m,i}(k) \triangleq \{(\boldsymbol{a}^{\mathrm{U}}_{m,i}(l), \boldsymbol{a}^{\mathrm{O}}_{m,i}(l))\}_{l=1}^{k}$ denote the set of updating and offloading actions of mobile device $m \in \mathcal{M}$ made until task k in episode i, where "H" refers to "history". Recall that $\boldsymbol{s}_0$ is the initial state of any arbitrary episode. We define $N_i(\boldsymbol{s}_0, \boldsymbol{a}^{\mathrm{H}}_{m,i})$ and $D_i(\boldsymbol{s}_0, \boldsymbol{a}^{\mathrm{H}}_{m,i})$ as follows:

$$N_i(\boldsymbol{s}_0, \boldsymbol{a}^{\mathrm{H}}_{m,i}(k)) = \sum_{l=1}^{k} A(Y^i_{m,l}, Z^i_{m,l+1}, Y^i_{m,l+1}), \quad (21)$$

$$D_i(\boldsymbol{s}_0, \boldsymbol{a}^{\mathrm{H}}_{m,i}(k)) = \sum_{l=1}^{k} (Y^i_{m,l} + Z^i_{m,l+1}), \quad (22)$$

where $Y^i_{m,l}$ and $Z^i_{m,l+1}$ are the delay of task l and the wait interval for generating the next task $l+1$, respectively, for mobile device m. Note that $Y^i_{m,l}$ is a function of $\boldsymbol{a}^{\mathrm{H}}_{m,i}(l)$, and $Z^i_{m,l+1}$ is a function of $\boldsymbol{a}^{\mathrm{H}}_{m,i}(l)$ and $\boldsymbol{a}^{\mathrm{U}}_{m,i}(l+1)$. The cost module keeps track of $\boldsymbol{a}^{\mathrm{H}}_{m,i}(k)$ or equivalently $Y^i_{m,l}$ and $Z^i_{m,l}$ for all $l = 1, ..., k$ across the training process.

In step k of episode i, a cost is determined and sent to the DDPG and D3QN modules. This process corresponds to the inner loop of the proposed fractional RL framework and is defined based on (21): for all $i \in [E], m \in \mathcal{M}, k \in \mathcal{K}$,

$$\begin{aligned} c_{m,i}(k) = &A(Y^i_{m,k}, Z^i_{m,k+1}, Y^i_{m,k+1}) \\ &- \gamma_{m,i} \cdot (Y^i_{m,k} + Z^i_{m,k+1}), \end{aligned} \quad (23)$$

where $A(Y^i_{m,k}, Z^i_{m,k+1}, Y^i_{m,k+1})$ stands for the area of a trapezoid in (3). The cost in (23) corresponds to an (immediate) cost function as in the fractional MDP problem in (11).

Finally, at the end of each episode i, the cost module updates $\gamma_{m,i+1}$ using (21) and (22):

$$\gamma_{m,i+1} = \frac{N_i(\boldsymbol{s}_0, \boldsymbol{a}^{\mathrm{H}}_{m,i}(T))}{D_i(\boldsymbol{s}_0, \boldsymbol{a}^{\mathrm{H}}_{m,i}(T))}, \quad i \in [E], \quad (24)$$

where T is the stepped needed set to be the same for every episode, as motivated in Appendix B. Eq. (24) corresponds to the update procedure of the quotient coefficient in (19) as in the outer loop of the fractional RL framework.

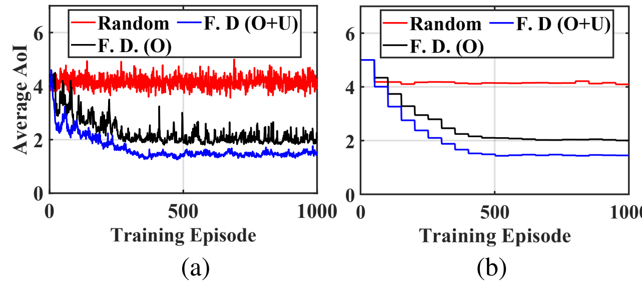

Figure 3: Convergence of (a) average AoI and (b) the average value of quotient coefficients $\gamma_{m,i}$ across devices, where $\gamma_{m,i}$ is updated every 50 episodes.

Performance Evaluation

We perform experiments to evaluate our proposed fractional DRL algorithm. We consider two edge nodes and 20 mobile devices learning their own policies simultaneously. Unless otherwise specified, we follow the experimental settings in (Tang and Wong 2022, Table I). We present more detailed experiment settings in Appendix D.

We denote proposed F. D. (O+U), which is short for "fractional DRL with offloading and updating policies". This is compared with several benchmark methods:

- *Random scheduling*: The updating and offloading decisions are randomly generated within action space.
- *PGOA (Yang et al. 2018)*: This corresponds to a best response algorithm for potential game in MEC systems.
- *Non-Fractional DRL* (denoted by Non-F. D.) (Xu et al. 2022): This benchmark adopts D3QN network to learn the offloading policy. In contrast to our proposed framework, this benchmark is non-fractional. That is, its objective approximates the ratio-of-expectation average AoI in (5) by an *expectation-of-ratio* expression: $\text{minimize}_{\boldsymbol{\pi}_m} \ \mathbb{E}\left[\frac{A(Y_{m,k}, Z_{m,k+1}, Y_{m,k+1})}{Y_{m,k}+Z_{m,k+1}} \middle| \boldsymbol{\pi}_m\right]$. Such an approximation can circumvent the fractional challenge but incurs large accuracy loss.

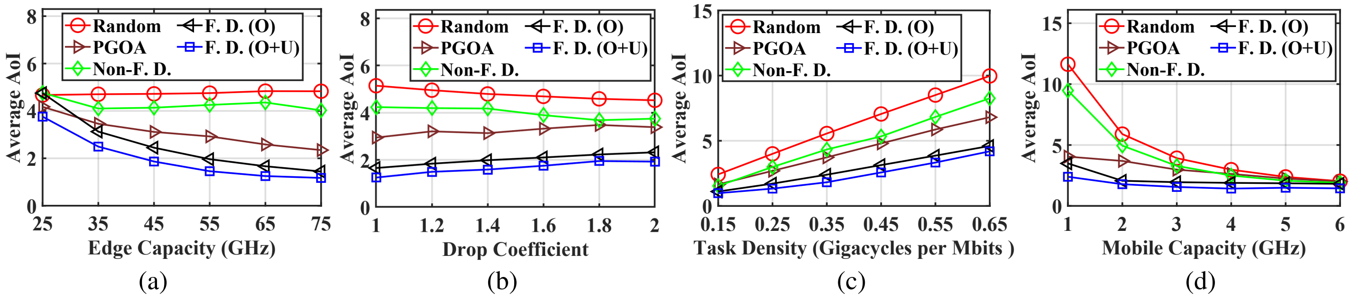

Figure 4: Performance under different (a) processing capacities of edges, (b) drop coefficients, (c) task densities, and (d) processing capacities of mobile devices.

- *Fractional DRL with Offloading Only* (denoted by F. D. (O)): We propose this algorithm by simplifying our fractional DRL algorithm through considering only the offloading policy. The updating policy (i.e., $Z_{m,k}$) is set to zero, as in many of the existing works (Xie, Wang, and Weng 2022; He et al. 2022; Xu et al. 2022).

The performance difference between Non-F. D. and F. D. (O) shows the significance of our proposed fractional DRL. The difference between Fr. D. (O) and F. D. (O+U) shows the necessity of joint offloading and waiting optimization.

Convergence: Fig. 3 illustrates the convergence of our proposed F. D. (O) and F. D. (O+U) algorithms. Unlike non-fractional approaches, our proposed approach involves the convergence of not only the neural network (see Fig. 3(a)) but also the quotient coefficient γ (see Fig. 3(b)). As a result, the convergence curve of AoI may sometimes change non-monotonically. In Fig. 3(a), both F. D. (O) and F. D. (O+U) converge after roughly 350 episodes. As for the converged AoI, F. D. (O+U) outperforms F. D. (O) by 31.3%.

Edge Capacity: Fig. 4(a) evaluates the performance of our proposed schemes under different node processing capacities. First, the proposed fractional DRL-based algorithm consistently achieves lower average AoI, compared against those non-fractional benchmarks. Such an advantage is more significant as the per-node processing capacity is larger. When processing capacity is 75 GHz, F. D. (O+U) can achieve an average AoI reduction of 54.8% and 73.6%, compared against PGOA and Non-F. D., respectively. Second, when compared with F. D. (O) algorithm, F. D. (O+U) can further reduce the average AoI up to 19.9% when the processing capacity of edge nodes is 55 GHz. This further shows that a well-designed updating policy also plays an important role of further improving the performance, especially there are relatively high edge loads.

Drop Coefficient: In Fig. 4(b), we consider different drop coefficients, i.e., the ratio of the drop time $\bar{Y}$ to the average time of processing a task. The performance gaps between the proposed schemes and benchmarks are large when the drop ratio gets smal (i.e., tasks are more delay-sensitive). When the drop coefficient is 1.0, the F. D. (O+U) algorithm reduces the average AoI by 57.6%, compared with the Non-Fractional DRL algorithm.

Task Density: In Fig. 4(c), we evaluate algorithm performance under different task densities, which affect the expected processing time of tasks at both edge nodes and mobile devices. Specifically, our proposed F. D. (O) and F. D. (O+U) schemes outperform all the benchmarks. In addition, the performance gaps increase as the task density increases, which shows the benefit of our proposed algorithm under large task densities. When the task density is 0.65, F. D. (O+U) achieves an average reductions of 38.4% and 49.3%, compared against PGOA and Non-Fractional DRL, respectively. Meanwhile, F. D. (O+U) algorithm outperforms F. D. (O) by up to 24.2%.

Mobile Capacity: In Fig. 4(d), as the processing capacity of mobile devices decreases, the gap between F. D. (O) and Non-F. D. significantly increases, indicating the necessity of our fractional scheme. When mobile capacity is 2 GHz, F. D. (O+U) algorithm can achieve an average AoI reduction of 48.8% compared to PGOA.

To summarize, *our proposed schemes significantly outperform non-fractional benchmarks, especially under large task density, delay-sensitive tasks, and small mobile device processing capacity. Meanwhile, the joint optimization over offloading and updating can further increase the system performance by up to* 31.3%. We present additional convergence and performance evaluation under different networks hyperparamters, distribution of processing duration and scale of mobile devices in Appendix D.

Conclusion

This paper has studied the computational task scheduling (including offloading and updating) problem for age-minimal MEC. To address the underlying challenges of unknown load dynamics and the fractional objective, we have proposed a fractional RL framework with a provable linear convergence rate. We further designed a fractional DRL algorithm that incorporates D3QN and DDPG techniques to tackle hybrid action space. Experimental results show that proposed fractional algorithms significantly reduce the average AoI, compared against several benchmarks. Meanwhile, the joint optimization of offloading and updating can further reduce the average AoI, validating the effectiveness of our proposed scheme. There are several future directions, including incorporating multi-agent RL with recurrent neural networks for non-stationarity and social optimal scheduling.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (Projects 62202427 and 62202214), in part by Guangdong Basic and Applied Basic Research Foundation under Grant 2023A1515012819, and in part by the Australian Research Council (ARC) Discovery Early Career Researcher Award (DECRA) under Grant DE230100046.

References

Akbari, M.; et al. 2021. Age of Information Aware VNF Scheduling in Industrial IoT Using Deep Reinforcement Learning. *IEEE J. Sel. Areas Commun.*, 39(8): 2487–2500.

Ceran, E. T.; Gündüz, D.; and György, A. 2021. A reinforcement learning approach to age of information in multi-user networks with HARQ. *IEEE J. Sel. Areas Commun.*, 39(5): 1412–1426.

Chen, J.; and Xie, H. 2022. An Online Learning Approach to Sequential User-Centric Selection Problems. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(66): 6231–6238.

Chen, X.; et al. 2020. Age of Information Aware Radio Resource Management in Vehicular Networks: A Proactive Deep Reinforcement Learning Perspective. *IEEE Trans. Wireless Commun.*, 19(4).

Chen, X.; et al. 2022. Information Freshness-Aware Task Offloading in Air-Ground Integrated Edge Computing Systems. *IEEE J. Sel. Areas Commun.*, 40(1): 243–258.

Chiariotti, F.; Vikhrova, O.; Soret, B.; and Popovski, P. 2021. Peak Age of Information Distribution for Edge Computing With Wireless Links. *IEEE Trans. Commun.*, 69(5): 3176–3191.

Dinkelbach, W. 1967. On nonlinear fractional programming. *Management science*, 13(7): 492–498.

Ghavamzadeh, M.; Kappen, H.; Azar, M.; and Munos, R. 2011. Speedy Q-Learning. In *Proc. Neural Info. Process. Syst. (NIPS)*, volume 24.

He, X.; Wang, S.; Wang, X.; Xu, S.; and Ren, J. 2022. Age-Based Scheduling for Monitoring and Control Applications in Mobile Edge Computing Systems. In *Proc. IEEE INFOCOM*.

Hu, J.; Zhang, H.; Song, L.; Schober, R.; and Poor, H. V. 2020. Cooperative Internet of UAVs: Distributed Trajectory Design by Multi-Agent Deep Reinforcement Learning. *IEEE Trans. Commun.*

Huang, L.; Bi, S.; and Zhang, Y.-J. A. 2020. Deep Reinforcement Learning for Online Computation Offloading in Wireless Powered Mobile-Edge Computing Networks. *IEEE Trans. Mobile Comput.*, 19(11): 2581 – 2593.

Jin, L.; Tang, M.; Zhang, M.; and Wang, H. 2023. Fractional Deep Reinforcement Learning for Age-Minimal Mobile Edge Computing.

Kaul, S.; Yates, R.; and Gruteser, M. 2012. Real-time status: How often should one update? In *Proc. IEEE INFOCOM*, 2731–2735.

Keating, C.; and Shadwick, W. F. 2002. A universal performance measure. *Journal of performance measurement*, 6(3): 59–84.

Kuang, Q.; Gong, J.; Chen, X.; and Ma, X. 2020. Analysis on computation-intensive status update in mobile edge computing. *IEEE Transactions on Vehicular Technology*, 69(4): 4353–4366.

Li, J.; Zhou, Y.; and Chen, H. 2020. Age of Information for Multicast Transmission With Fixed and Random Deadlines in IoT Systems. *IEEE Internet of Things Journal*, 7(9): 8178–8191.

Lillicrap, T. P.; Hunt, J. J.; Pritzel, A.; Heess, N.; Erez, T.; Tassa, Y.; Silver, D.; and Wierstra, D. 2015. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*.

Liu, S.; Zheng, C.; Huang, Y.; and Quek, T. Q. S. 2022a. Distributed Reinforcement Learning for Privacy-Preserving Dynamic Edge Caching. *IEEE J. Sel. Areas Commun.*, 40(3): 749–760.

Liu, T.; et al. 2022b. Deep Reinforcement Learning based Approach for Online Service Placement and Computation Resource Allocation in Edge Computing. *IEEE Trans. Mobile Comput.*

M. Zhou, H. H. Y., M. Zhang; and Yates, R. D. 2024, Early Access. Age-minimal CPU Scheduling. In *Proc. IEEE INFOCOM*.

Ma, H.; Huang, P.; Zhou, Z.; Zhang, X.; and Chen, X. 2022. GreenEdge: Joint Green Energy Scheduling and Dynamic Task Offloading in Multi-Tier Edge Computing Systems. *IEEE Transactions on Vehicular Technology*, 71(4): 4322–4335.

Mao, Y.; You, C.; Zhang, J.; Huang, K.; and Letaief, K. B. 2017. A survey on mobile edge computing: The communication perspective. *IEEE Commun. Surveys & Tuts.*, 19(4): 2322–2358.

Mnih, V.; et al. 2015. Human-level control through deep reinforcement learning. *Nature*, 518(7540): 529–533.

Omidkar, A.; Khalili, A.; Nguyen, H. H.; and Shafiei, H. 2022. Reinforcement-Learning-Based Resource Allocation for Energy-Harvesting-Aided D2D Communications in IoT Networks. *IEEE Internet Things J*, 9(17): 16521–16531.

Pan, J.; Bedewy, A. M.; Sun, Y.; and Shroff, N. B. 2023. Optimal sampling for data freshness: Unreliable transmissions with random two-way delay. *IEEE/ACM Transactions on Networking*, 31(1): 408–420.

Pan, J.; Sun, Y.; and Shroff, N. B. 2023. Sampling for remote estimation of the Wiener process over an unreliable channel. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 7(3): 1–41.

Porambage, P.; et al. 2018. Survey on multi-access edge computing for Internet of things realization. *IEEE Commun. Surveys & Tuts.*, 20(4): 2961–2991.

Puterman, M. L. 2014. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons.

Ren, Z.; and Krogh, B. 2005. Markov decision Processes with fractional costs. *IEEE Transactions on Automatic Control*, 50(5): 646–650.

Shisher, M. K. C.; and Sun, Y. 2022. How Does Data Freshness Affect Real-Time Supervised Learning? In *Proc. ACM MobiHoc*. Seoul, Republic of Korea.

Sun, Y.; Uysal-Biyikoglu, E.; Yates, R. D.; Koksal, C. E.; and Shroff, N. B. 2017. Update or Wait: How to Keep Your Data Fresh. *IEEE Trans. Inform. Theory*, 63(11): 7492–7508.

Suttle, W.; Zhang, K.; Yang, Z.; Liu, J.; and Kraemer, D. 2021. Reinforcement Learning for Cost-Aware Markov Decision Processes. In Meila, M.; and Zhang, T., eds., *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, 9989–9999. PMLR.

Taka, H.; He, F.; and Oki, E. 2022. Service Placement and User Assignment in Multi-Access Edge Computing with Base-Station Failure. In *Proc. IEEE/ACM IWQoS*.

Tanaka, T. 2017. A partially observable discrete time Markov decision process with a fractional discounted reward. *Journal of Information and Optimization Sciences*, 38(1): 21–37.

Tang, M.; and Wong, V. W. 2022. Deep Reinforcement Learning for Task Offloading in Mobile Edge Computing Systems. *IEEE Trans. Mobile Comput.*, 21(6): 1985–1997.

Tang, Z.; Sun, Z.; Yang, N.; and Zhou, X. 2021. Age of Information Analysis of Multi-user Mobile Edge Computing Systems. In *Proc. IEEE GLOBECOM*.

Tuli, S.; et al. 2022. Dynamic Scheduling for Stochastic Edge-Cloud Computing Environments Using A3C Learning and Residual Recurrent Neural Networks. *IEEE Trans. Mobile Comput.*, 21(3).

Wang, S.; Guo, Y.; Zhang, N.; Yang, P.; Zhou, A.; and Shen, X. 2021. Delay-Aware Microservice Coordination in Mobile Edge Computing: A Reinforcement Learning Approach. *IEEE Trans. Mobile Comput.*, 20(3): 939 – 951.

Wang, S.; et al. 2022a. Distributed Reinforcement Learning for Age of Information Minimization in Real-Time IoT Systems. *IEEE J. Sel. Top. Signal Process.*, 16(3): 501–515.

Wang, X.; Ye, J.; and Lui, J. C. 2022. Decentralized Task Offloading in Edge Computing: A Multi-User Multi-Armed Bandit Approach. In *Proc. IEEE Conference on Computer Communications (INFOCOM)*.

Wang, Z.; Wei, Y.; Richard Yu, F.; and Han, Z. 2022b. Utility Optimization for Resource Allocation in Multi-Access Edge Network Slicing: A Twin-Actor Deep Deterministic Policy Gradient Approach. *IEEE Trans. Wireless Commun.*, 1–1.

Wu, F.; Zhang, H.; Wu, J.; Han, Z.; Poor, H. V.; and Song, L. 2021. UAV-to-Device Underlay Communications: Age of Information Minimization by Multi-Agent Deep Reinforcement Learning. *IEEE Transactions on Communications*, 69(7): 4461–4475.

Xie, X.; Wang, H.; and Weng, M. 2022. A Reinforcement Learning Approach for Optimizing the Age-of-Computing-Enabled IoT. *IEEE Internet of Things Journal*, 9(4): 2778–2786.

Xu, C.; et al. 2022. AoI-centric Task Scheduling for Autonomous Driving Systems. In *Proc. IEEE INFOCOM*.

Yang, L.; Zhang, H.; Li, X.; Ji, H.; and Leung, V. 2018. A Distributed Computation Offloading Strategy in Small-Cell Networks Integrated With Mobile Edge Computing. *IEEE/ACM Trans. Netw.*, 26(6).

Yates, R. D.; Sun, Y.; Brown, D. R.; Kaul, S. K.; Modiano, E.; and Ulukus, S. 2021. Age of Information: An Introduction and Survey. *IEEE J. Sel. Areas Commun.*, 39(5): 1183–1210.

Zhao, N.; Ye, Z.; Pei, Y.; Liang, Y.-C.; and Niyato, D. 2022. Multi-Agent Deep Reinforcement Learning for Task Offloading in UAV-assisted Mobile Edge Computing. *IEEE Trans. Wireless Commun.*

Zhu, J.; and Gong, J. 2022. Online Scheduling of Transmission and Processing for AoI Minimization with Edge Computing. In *Proc. IEEE INFOCOM WKSHPS*.

Zhu, Z.; Wan, S.; Fan, P.; and Letaief, K. B. 2022. Federated Multiagent Actor–Critic Learning for Age Sensitive Mobile-Edge Computing. *IEEE Internet of Things Journal*, 9(2): 1053–1067.

Zou, P.; Ozel, O.; and Subramaniam, S. 2021. Optimizing Information Freshness Through Computation–Transmission Tradeoff and Queue Management in Edge Computing. *IEEE/ACM Trans. Netw.*, 29(2).