Minimizing Age of Information for Mobile Edge Computing Systems: A Nested Index Approach

Shuo Chen, Ning Yang*, Meng Zhang*, Jun Wang

Abstract—Exploiting the computational heterogeneity of mobile devices and edge nodes, *mobile edge computation* (MEC) provides an efficient approach to achieving real-time applications that are sensitive to information freshness, by offloading tasks from mobile devices to edge nodes. We use the metric *Age-of-Information* (AoI) to evaluate information freshness. An efficient solution to minimize the AoI for the MEC system with multiple users is non-trivial to obtain due to the random computing time. In this paper, we consider multiple users offloading tasks to heterogeneous edge servers in a MEC system. We first reformulate the problem as a *Restless Multi-Arm-Bandit* (RMAB) problem and establish a hierarchical *Markov Decision Process* (MDP) to characterize the updating of AoI for the MEC system. Based on the hierarchical MDP, we propose a nested index framework and design a nested index policy with provably asymptotic optimality. Finally, the closed form of the nested index is obtained, which enables the performance tradeoffs between computation complexity and accuracy. Our algorithm leads to an optimality gap reduction of up to 40%, compared to benchmarks. Our algorithm asymptotically approximates the lower bound as the system scalar gets large enough.

I. INTRODUCTION

A. Motivation

Large-scale cyber-physical applications necessitate real-time information. For example, Internet of Things (IoT) devices, constrained by limited computational resources, rely on cloud computing to boost performance, while sensor data from vehicles must be collected and processed to depict surroundings and facilitate navigation. Users demand prompt status updates. The *Age-of-Information* (AoI) is a recently introduced metric designed to assess the freshness of information, quantifying the time elapsed since the most recent message update (e.g., [1], [2]).

In numerous real-time applications, such as autonomous driving, the updated information is computationally demanding and necessitates processing. Offloading data to the cloud for computation can lead to data staleness and is computationally expensive. The *Mobile Edge Computing* (MEC) paradigm shifts servers from the cloud to the edge, bringing users closer to servers and thereby reducing transmission delay (e.g., [3]–[5]). Consequently, MEC emerges as a promising technology capable of reducing latency and enhancing information freshness.

The majority of existing studies [6]–[13] primarily concentrate on optimizing AoI in MEC systems with a single user or server, or under the assumption of fixed computation time and task size. However, in practical scenarios, multiple heterogeneous users and servers are prevalent, prompting further exploration of heterogeneous MEC systems. Nevertheless, minimizing AoI in MEC systems with heterogeneous servers presents two challenges: determining the optimal location for task offloading and deciding the time for this offloading. To this end, we first answer the following question:

Question 1. *How should one minimize the AoI in a MEC system with multiple heterogeneous users and servers?*

The task of minimizing AoI is frequently formulated as a *Restless Multi-Arm Bandit* (RMAB) problem, as it can be optimally solved by value iteration [14]. However, such strategies are prone to the curse of dimensionality, necessitating near-optimal solutions with low complexity. A promising method for addressing the RMAB problem is the index policy approach [15], which is particularly suitable for scheduling systems with multiple nodes. This approach can yield near-optimal results with relatively low computational complexity. The effectiveness of the index policy is attributed to two primary factors: the ability to decompose the original problem into several sub-problems with practicable optimal solutions and the potential to express the index in closed form for a specific *Markov Decision Process* (MDP) structure, thereby reducing computational complexity. Regrettably, neither of these factors can be easily assured: the optimal solution for the sub-problem may not exist, and obtaining the index function is non-trivial due to the presence of multiple state variables in MEC systems with heterogeneous users and servers. This leads us to the following question:

Question 2. *How should we design an index-based policy for RMAB problems with multi-dimensional state variables?*

B. Solution Approach

In response to this challenge, we suggest a framework where multiple heterogeneous users offload tasks to heterogeneous edge servers. We construct a multi-layer Markov Decision Process model aimed at minimizing the average AoI in MEC. The primary contributions of our research are as follows:

- *Problem Formulation:* We formulate the problem of minimizing average AoI for MEC by optimizing offloading

Shuo Chen and Ning Yang are with Institute of Automation, Chinese Academy of Sciences, Beijing, 100190, China. (e-mail: shuo.chen22@imperial.ac.uk, ning.yang@ia.ac.cn).

Meng Zhang is with the ZJU-UIUC Institute, Zhejiang University, Zhejiang, 314499, China. (e-mail: mengzhang@intl.zju.edu.cn).

Jun Wang is with the Department of Computer Science, University College London, WC1E 6BT, UK. (e-mail: jun.wang@cs.ucl.ac.uk).

(*Corresponding author: Ning Yang, Meng Zhang)

policies and reformulating it as an RMAB problem. *To the best of our knowledge, this represents the first formulation of an age-minimal MEC problem that takes into account multiple heterogeneous users and edge servers.*

- *Nested Index Approach:* We construct a multi-layer MDP model and, based on this, introduce a *nested index framework* to solve our RMAB problem. We demonstrate the indexability of the multi-layer MDP for our MEC system and design the corresponding index function. We propose a nested index algorithm with provable asymptotic optimality.
- *Numerical Results:* Our nested index algorithm results in an optimality gap reduction of up to 40% compared to benchmarks. Our algorithm converges to the lower bound as the system scale increases sufficiently.

II. RELATED WORK

A. Age-of-Information

Kaul *et al.* in [16] first proposed AoI as a metric to evaluate information freshness. The optimal AoI scheduling policy was to send messages from a source to the monitor through a single channel [17]. In [6], multiple sources could send updates over a single-hot network to a monitor, and they derived an approximate expression for the average AoI. In [7], they minimized AoI by considering multiple sources for queuing systems. In [8], they proposed a scheduling policy to minimize AoI in the wireless broadcast network with unreliable channels. In [9], they derived the structure of optimal policies for AoI minimizing problem and proved the optimality with reliable channel and unreliable channel assumptions. However, there was a lack of research on minimizing AoI in the more general MEC systems with heterogeneous multi-sources and edge servers.

B. Restless Multi-Arm Bandit

The RMAB problem arises when the state of an arm keeps changing whenever it is pulled or not [18]. In [10], they formulated the problem of minimizing AoI as an RMAB problem and demonstrated that Whittle's Index was optimal when the arms were stochastically identical in a single-hop network. They also mentioned that a classic MDP was always indexable and proved the indexability of certain RMAB problems. Hsu *et al.* [11] assumed that only one user could update at each time slot and obtained Whittle's Index in closed form. Hsu *et al.* [12] further studied the online and offline versions of the index approach and showed that the index policy was optimal when the arrival rate was constant. All the above index policies can only solve RMAB problems with one-dimensional state variables. However, in general, there exist more factors that affect decisions in wireless networks. Therefore, we need to consider multiple state variables for general wireless networks.

C. Mobile Edge Computing

In MEC scenarios, mobile edge servers are well equipped with sufficient computational resources and are close to users, enabling them to expedite the computation process. Yang *et al.*
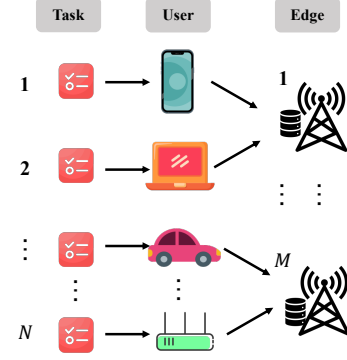

Fig. 1: An MEC system in which each user offloads tasks to a certain edge server and receives data after processing.

[19], [20] studied the resource management problem in MEC utilizing reinforcement learning approaches. In [13], an MDP-based policy was proposed to determine whether to offload a task and when to transmit it. Zou and Ozel in [21] studied the transmission and computation process for MEC systems as coupled two queues in tandem. The computing time is random in the MEC system. The optimal scheduling policy contains non-preemptive [17] and preemptive [21] structures, respectively. In [21], the optimal scheduling policy under preemptive structure had a threshold property, and it was a benefit for minimizing AoI to wait before offloading. For minimizing AoI problems with multiple sources (or users), they established the MDP model and index-based policies [15], [22], which had less complex and relatively efficient. Such an index-based policy was proved to be asymptotic for many single-hop wireless network scheduling. To summarize, random offloading time and indeterminate computation durations considering preemptive techniques posed significant offloading challenges in the MEC system.

III. SYSTEM MODEL

A. System Overview

We consider a MEC system with N users who generate computational tasks and offload them to M heterogeneous edge servers, as shown in Fig. 1. Let $n \in \mathcal{N}$, $\mathcal{N} = \{1, 2, ..., N\}$ be the index of users and $m \in \mathcal{M}$, $\mathcal{M} = \{1, 2, ..., M\}$ denote the index of edge servers. Let $t \in \mathcal{T}$ be the index of each time slot with $\mathcal{T} = \{1, 2, ..., T\}$. We consider the generate-at-will model [11], [17], i.e., the user can decide whether to generate a new task or not at each time slot t. The transmission time between users and edge servers is negligible, and once the computation of one task completes, its result is immediately sent back to the user. Each user can send a proportion of its task to any server for computing at each time slot [23].

We assume that the edge servers are heterogeneous and the computing time of tasks is stochastic. Each task of user n has a specific workload. When offloaded to a server, the task needs a specific number of CPU cycles to finish computing, and the computing time is based on both the workload and

CPU frequency of the chosen server. We assume there is a minimum computing time τ_n^{min} for the task of user n.

We use the notion AoI to measure the freshness of information. We denote $\Delta_n(t)$ as the AoI for user n at time t. The age of user n decreases to the age of the latest offloaded task when the computing finishes or increases by 1 otherwise. Let $G_n(t)$ denote the generation time of the most recent task offloaded by user n at time t. Then, the age of user n at time t if the computing finishes can be expressed as

$$\Delta_n(t) = t - G_n(t), \quad \forall n \in \mathcal{N}. \quad (1)$$

1) Offloading Decision: At time t, each user can choose one edge server to offload its tasks. When a task is offloaded, the computation starts at the beginning of each time slot. We denote $y_{nm}(t) \in \{0, 1\}$ as the offloading decision variable for user n at time t: $y_{nm}(t) = 1$ if user n decides to offload a task to server m. When $y_{nm}(t) = 0, \forall m \in \mathcal{M}$, no task is to offload or the current task is to be dropped.

Users' offloading decisions are subject to the following constraints:

$$\sum_{n \in \mathcal{N}} \sum_{m \in \mathcal{M}} y_{nm}(t) \leq M, \qquad \forall t \in \mathcal{T}, \quad (2a)$$

$$\sum_{m \in \mathcal{M}} y_{nm}(t) \leq 1, \qquad \forall n \in \mathcal{N}, t \in \mathcal{T}, \quad (2b)$$

$$y_{nm}(t) \in \{1, 0\}, \quad \forall n \in \mathcal{N}, m \in \mathcal{M}, t \in \mathcal{T}. \quad (2c)$$

Specifically, constraint (2a) means there are at most M servers to be chosen for offloading, and constraint (2b) means each user can offload its task to only one server at the same time. Constraint (2c) is a indicator function, which denotes whether a task is offload to server m at time t.

2) Shifted Geometric Distribution: The transition of AoI during computation obeys a shifted geometric distribution [24] with parameter $p_m = 1 - e^{-\lambda_m}$ for tasks offloaded to server m, where λ_m is the parameter of an exponential distribution. We consider a minimal computational time for each computational task, denoted by τ_n^{min}, i.e., only after τ_n^{min} time slots, each edge server m completes the computation of the task with a probability p_m within each time slot. Therefore, the transition probability of the AoI of each user n during the computation can be written as

$$\mathbb{P}\{\Delta_n(t+1) = \Delta_n(t) + 1 \mid$$
$$t - G_n(t) > \tau_n^{min}, y_{nm}(t) = 1\} = 1 - p_m, \quad (3a)$$
$$\mathbb{P}\{\Delta_n(t+1) = t - G_n(t) + 1 \mid$$
$$t - G_n(t) > \tau_n^{min}, y_{nm}(t) = 1\} = p_m, \quad (3b)$$
$$\mathbb{P}\{\Delta_n(t+1) = \Delta_n(t) + 1 \mid$$
$$t - G_n(t) \leq \tau_n^{min}, y_{nm}(t) = 1\} = 1, \quad (3c)$$
$$\mathbb{P}\{\Delta_n(t+1) = \Delta_n(t) + 1 \mid y_{nm}(t) = 0\} = 1. \quad (3d)$$

B. Problem Formulation

We aim to minimize the overall AoI of the MEC system. In the following, we formulate the AoI minimization problem.

Let $\pi \in \Pi$ denote the scheduling policy, which maps from the system state to the actions of all users.

We define the long-term average AoI [1], [12], [15] under policy π as:

$$\limsup_{T \to \infty} \frac{1}{TN} \sum_{t=1}^{T} \sum_{n \in \mathcal{N}} \mathbb{E}_{\boldsymbol{y} \sim \pi}[\Delta_n(t)], \quad (4)$$

where we consider the policy π is deterministic stationary [25].

We reformulate the minimization problem of the long-term average AoI into the following form:

$$\min_{\pi} \quad \limsup_{T \to \infty} \frac{1}{TN} \sum_{t=1}^{T} \sum_{n \in \mathcal{N}} \mathbb{E}_{\boldsymbol{y} \sim \pi}[\Delta_n(t)] \quad (5a)$$

$$\text{s.t.} \quad (2)$$

Based on the Lagrangian relaxation [26], we relax the instantaneous constraint (2a) to average constraint, then drop constraint (2b) by introducing dual variables $\nu_m, \forall m \in \mathcal{M}$ and deriving N sub-problems. Define $\pi_n \in \Pi_n$ as the policy which maps from the state of user n to the action of user n. Given dual variables, each sub-problem n is formulated as:

$$\min_{\pi_n} \quad \limsup_{T \to \infty} \frac{1}{TN} \sum_{t=1}^{T} \mathbb{E}_{\boldsymbol{y}_n \sim \pi_n} \Big[\Delta_n(t) + \sum_{m \in \mathcal{M}} \nu_m y_{nm}(t) \Big]$$
$$(6a)$$

$$\text{s.t.} \quad \sum_{m \in \mathcal{M}} y_{nm}(t) \leq 1, \qquad \forall n \in \mathcal{N}, t \in \mathcal{T}, \quad (6b)$$

$$y_{nm}(t) \in \{1, 0\}, \quad \forall n \in \mathcal{N}, m \in \mathcal{M}, \forall t \in \mathcal{T}. \quad (6c)$$

When the dual variables converge, the sum of solutions to each sub-problem (6) reaches the lower bound of the solution to problem (5), i.e., $\limsup_{T \to \infty} \frac{1}{TN} \sum_{t=1}^{T} \sum_{n \in \mathcal{N}} \mathbb{E}_{\boldsymbol{y} \sim \pi}[\Delta_n(t)] \geq \sum_{n \in \mathcal{N}} \limsup_{T \to \infty} \frac{1}{TN} \sum_{t=1}^{T} \mathbb{E}_{\boldsymbol{y}_n \sim \pi_n} \Big[\Delta_n(t) + \sum_{m \in \mathcal{M}} \nu_m y_{nm}(t) \Big]$. We will further study the optimal policy π_n^* for the decomposed problem (6) given $\{\nu_1, \nu_2, \dots, \nu_M\}$ by considering a multi-layer MDP.

C. Multi-Layer MDP

Definition 1 (L-Layer MDP)**.** *An L-layer MDP is a tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{C}, L \rangle$, where $\mathcal{S}$ denotes the state space, $\mathcal{A}$ is the action space, the transition function is $\mathcal{P} : \mathcal{S} \times \mathcal{A} \to PD(\mathcal{S})$, the cost function is $\mathcal{C} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$, and $L \in \mathbb{Z}_+$ is the number of layers. Denote $\mathcal{S}_l = \mathbb{N}^l$ as the state space at layer l, and $\mathcal{S} = \mathcal{S}_1 \cup \mathcal{S}_2 \cup \dots \cup \mathcal{S}_L$. An L-layer MDP fulfills the following conditions: $\mathcal{S}_l \subset \mathbb{N}^L$, and $\mathcal{S} = \mathcal{S}_1 \cup \mathcal{S}_2 \cup \dots \cup \mathcal{S}_L$*

- *$\forall 0 < l < L$ and $\forall s \in \mathcal{S}_l$, there exist some $a \in \mathcal{A}$ and $s' \in \mathcal{S}_{l+1}$ that satisfies $\mathbb{P}\{s' \mid s, a\} > 0$,*
- *$\forall 0 < l \leq L$ and $\forall s \in \mathcal{S}_l$, there exist some $a \in \mathcal{A}$ and $s'' \in \mathcal{S}_l$ that satisfies $\mathbb{P}\{s'' \mid s, a\} > 0$,*
- *$\forall L \geq l > 1$, there exists some $s \in \mathcal{S}_l$, $s''' \in \mathcal{S}_1$, and $a \in \mathcal{A}$ that satisfies $\mathbb{P}\{s''' \mid s, a\} > 0$,*

and we term the sub-space $\mathcal{S}_l$ as layer l.

The multi-layer MDP defines the transition probability among states at different layers. The state at layer l should be able to transit to states at layer l, $l+1$, and layer 1. In a multi-layer MDP, states only transit among neighbor layers, which gives insights into the analysis of multi-dimensional state variables.

Now, we specify the multi-layer MDP for the MEC system. Each sub-problem (6) can be formulated as a 2-layer MDP:

- **Action space:** Let $\mathcal{A} = \{0, 1\}^M$ be the action space for each user, and the action of user n, which is denoted as $\boldsymbol{y}_n(t) = \{y_{n1}(t), y_{n2}(t), \ldots, y_{nM}(t)\} \in \mathcal{A}$, contains the information of offloading decisions. The action vector $\boldsymbol{y}_n(t)$ is composed entirely of zero elements, with at most one element being one.

- **State space:** Let $\mathcal{S}_l$ denote the state space for each user at layer l. Recall that we denote $\Delta_n(t)$ as the age of user n and $G_n(t)$ denotes the generation time of the latest task of user n. User n who is idle is defined at layer 1, which has state $s_n(t) = \Delta_n(t) \in \mathcal{S}_1$. We consider users waiting for the result of the computation is at layer 2. Let $D_n(t) = \Delta_n(G_n(t))$ denote the age of user n when the latest task was generated. Thus we have state $s_n(t) = (\Delta_n(t), D_n(t)) \in \mathcal{S}_2$.

- **Transition function:** In the former section, we derive the transition probability in terms of $\Delta_n(t)$, while solely using $\Delta_n(t)$ can not fully characterize the transition of the states in multi-layer MDP. We use $q_{nm}^{ss'}$ to denote the transition probability of user n from state s to next state s' by choosing server m at time t, and we have

$$q_{nm}^{ss'} = \mathbb{P}\{s' \mid s, \ y_{nm}(t) = 1\}, \tag{7}$$

where $q_{nm}^{ss'}$ can be derived from the transition probability in Section III-A2.

- **Cost function:** We define the immediate cost as

$$C_n(s_n(t), m) \triangleq \Delta_n(t) + \nu_m, \tag{8}$$

which includes the current AoI and the server cost.

According to [25], it is simple to derive that there exists one deterministic stationary policy π_n^* that reaches optimal average AoI. However, value iteration when deriving the optimal policy suffers from the curse of dimensionality [27]. We need to seek an approach that owns less complexity and is near-optimal.

IV. INDEX-BASED POLICY

In this section, we introduce a nested index approach to our RMAB problem, which is proven to be an asymptotically optimal offloading policy. First, we define the nested index and prove that the 2-layer MDP for MEC systems fulfills the indexability condition. Next, we propose the nested index policy to schedule tasks. In addition, we also verify the asymptotic optimality of the proposed approach and obtain the nested index function in a closed form.

A. Nested Index

We first introduce the following definition of a *passive set* based on [18]. Define $\boldsymbol{\nu} \triangleq (\nu_1, \cdots, \nu_M)$ as the vector of *activating cost*, where each ν_m is the server cost of choosing server m for computation. We focus on sub-problem (6) with given cost $\boldsymbol{\nu}$. We introduce the cost-to-go function which is a prediction of cost to evaluate the value of the state s. Denote the optimal average cost of sub-problem n as γ_n^*, which is the minimum cost per stage. We can write the Bellman Equation of each sub-problem (6b) as:

$$\gamma_n^* + V_n(s, \boldsymbol{\nu}) = \min_{m \in \mathcal{M}, l \in \mathcal{L}} \left[C_n(s, m) + \sum_{s' \in \mathcal{S}_l} q_{nm}^{ss'} V_n(s', \boldsymbol{\nu}) \right], \tag{9}$$

where function $V_n(s, \boldsymbol{\nu})$ is the differential cost-to-go [27].

Let

$$\mu_{nm}(s, \boldsymbol{\nu}) = C_n(s, m) + \sum_{s' \in \mathcal{S}} q_{nm}^{ss'} V_n(s', \boldsymbol{\nu}) - \gamma_n^* \tag{10}$$

denote the expected cost of choosing server m given state s.

As the decision process for our multi-layer MDP is different from Whittle's Index, which involves multi-dimensional state variables and multiple feasible actions, it motivates us to consider a multi-layer index structure.

Definition 2 (Passive Set). *The passive set for user n to transit to layer l at server m given activating cost $\boldsymbol{\nu}$ is denoted as:*

$$\mathcal{P}_{nm}^l(\boldsymbol{\nu}) \triangleq \left\{ s \in \mathcal{S}_l \mid \min_{m' \in \mathcal{M}, m' \neq m} \mu_{nm'}(s, \boldsymbol{\nu}) \leq \mu_{nm}(s, \boldsymbol{\nu}) \right\}. \tag{11}$$

We denote $\mathcal{P}_{nm}(\boldsymbol{\nu}) \triangleq \cup_{l=1}^{L} \mathcal{P}_{nm}^l(\boldsymbol{\nu})$ as the overall passive set.

The passive set refers to the set of states at layer l that are sub-optimal for selecting server m for computing with activating costs $\boldsymbol{\nu}$.

In classic RMAB problems [11], [28], activating cost ν is a scalar. Whittle [18] stated that if the cardinality of the passive set increases monotonically from 0 to $+\infty$ as activating cost ν increases from 0 to $+\infty$, the problem is *indexable*. Each state is assigned a maximum activating cost, which makes the same cost-to-go to take action or not at this state. The activating cost also gives the urgency of such a state as Whittle stated [18]. A state with a higher activating cost has a higher priority for selection.

In MEC systems, we have multiple actions to choose from for each user, and we have heterogeneous servers with varying activating costs, which complicates the definition of *indexability*. Therefore, we need to design a more sophisticated index technique.

Definition 3 (Intra-Indexability). *In a Multi-Layer MDP $\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{C}, L \rangle$, given servers cost $\boldsymbol{\nu}$, if for any layer l, the cardinality of passive set $|\mathcal{P}_{nm}^l(\boldsymbol{\nu})|$ increases monotonically to the cardinality $|\mathcal{S}_l|$ of layer l as cost ν_m for server m increases from 0 to $+\infty$, then this multi-layer MDP is intra-indexable.*

The intra-indexability describes the relation between server costs and the optimal state to choose server m at each layer. Given layer l, there exists the largest server cost ν'_m that state $s_n(t)$ is no longer included in the passive set $\mathcal{P}^l_{nm}(\nu)$ at layer l. The monotonic property guarantees the uniqueness of such an activating cost. The cardinality passive set over all layers $|\mathcal{P}_{nm}(\nu)|$ is non-decreasing in ν if $|\mathcal{P}^l_{nm}(\nu)|$ is non-decreasing in ν, $\forall 1 \leq l \leq L$.

It is non-trivial to derive the optimal state at layer l for server m through the Bellman Equation Eq.(9). We can, however, conclude the structure-property of the optimal solution for each sub-problem (6). As there are multiple layers in the multi-layer MDP, the optimal solution has a *Multi-Layer-Threshold Type* (MLTT) structure:

Definition 4 (Multi-Layer-Threshold Policy). *Denote $\tilde{m} = \arg\max\limits_{m \in \mathcal{M}} p_m$ as the index of the server that owns the best computational performance. If the following two conditions hold:*

1) *If user n is at layer 1 with state $s_n(t) = (\Delta_n(t) = A)$:*
   - *for any server $m' \neq \tilde{m}, m' \in \mathcal{M}$, there exists $H_n(m', \tilde{m}, 0)$ that $\forall A \geq \max\limits_{m'}\{H_n(m', \tilde{m}, 0)\}$, $\pi_n(s_n(t)) = \tilde{m}$;*
   - *for any two states $s_n(t), s_n(t')$ which fulfill $\Delta_n(t) < \Delta_n(t')$, then $p_{\pi_n(s(t))} \leq p_{\pi_n(s(t'))}$;*
2) *If user n is at layer 2 with state $s_n(t) = (\Delta_n(t) = A, D_n(t) = D)$:*
   - *given D, there exists $H_n(m', \tilde{m}, D)$ that the $\forall A \geq H_n(m, m', D)$, $\pi_n(s_n(t)) = \tilde{m}$;*
   - *if $D_n(t) = D_n(t')$, $p_{\pi_n(s(t))} \leq p_{\pi_n(s(t'))}$;*

then the offloading policy π_n for the sub-problem (6) n has a Multi-Layer-Threshold Type (MLTT) structure.

The MLTT structure shows some common properties for optimal thresholds at both layer 1 and layer 2. The proposition proposes that a threshold exists for a user when choosing between two servers, and the threshold depends only on the current age of the user given the same age at generation. Due to the limited space, all the proofs are provided in the online appendix [29].

Proposition 1. *The optimal solution π_n^* to the sub-problem (6) is MLTT.*

The proof is shown in Appendix A.

The MLTT is a more strict property than the threshold policy, as different actions have their own optimal thresholds on the age of users, and the threshold is determined by the layer l and the generating age $D_n(t)$. By utilizing the MLTT property of the solution to problems in MEC systems, we can show the intra-indexability of the sub-problem (6).

Theorem 1. *The MDP sub-problem (6) is intra-indexable given cost ν.*

The proof is shown in Appendix B.

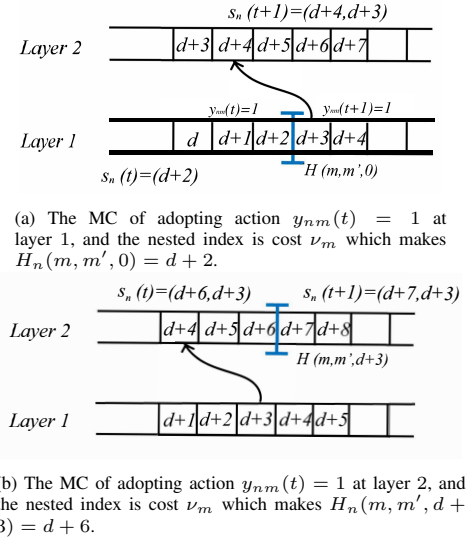Since the AoI minimizing problem in MEC systems is a 2-layer MDP, we design thresholds for actions at both layers.

(a) The MC of adopting action $y_{nm}(t) = 1$ at layer 1, and the nested index is cost ν_m which makes $H_n(m, m', 0) = d + 2$.

(b) The MC of adopting action $y_{nm}(t) = 1$ at layer 2, and the nested index is cost ν_m which makes $H_n(m, m', d + 3) = d + 6$.

Fig. 2: The relationship between the *nested-index* and the threshold in MC.

The intra-indexability property guarantees that there is one unique threshold for each server at each layer, and the size of the passive set for one server at each layer also increases as the server cost increases. Therefore, we can still use the index to represent the urgency of a state, and the comparison of states among layers is possible. We can then define the index for our 2-layer MDP.

Definition 5 (Nested Index). *Let ν_{-m} denote the activating cost of edge servers except for server m. The nested index for taking $y_n(t) = m$ at state $s_n(t)$ is defined as*

$$I_{nm}(s_n(t), \nu) \triangleq$$
$$\max\left[0, \inf\left\{\nu_m \mid \min_{m' \in \mathcal{M}} \mu_{nm'}(s_n(t), [\nu_{-m}, \nu_m])\right.\right. \qquad (12)$$
$$\left.\left. < \mu_{nm}(s_n(t), [\nu_{-m}, \nu_m])\right\}\right].$$

The nested index allows us to characterize the urgency of each state. Compared with the partial index [15], the nested index requires server costs other than server m, and the shows the emergency of transiting to different layers.

Fig. 2 illustrates the relationship between the *nested index* and the MDP. Fig. 2a compares the decision of offloading a task to server m at neighbor time slots for user n at layer 1. The optimal threshold $H_n(m, m', 0)$ decreases as the server cost ν_m increases from 0 to ∞. The infimum of ν_m that makes $s_n(t)$, i.e., $\Delta_n(t)$, the optimal state to offload a task to server m is the corresponding nested index. Fig. 2b illustrates the *nested index* at layer 2. For simplicity, we consider user n with the same generating age at time t and $t+1$, i.e., $D_n(t) = D_n(t+1)$. The nested index can also be derived by adjusting ν_m when it is optimal to offload a task to server m at time t. The *nested index* gives the emergency of offloading tasks for a user at state $s_n(t)$.

Algorithm 1 A Nested Index Policy
___
1: Initialize parameters N, M, L, β;
2: Initialize $s_n(0)$ for each user n and server cost $\boldsymbol{\nu}_0$;
3: **for** $t \leq T$ **do**
4:   Compute nested index $I_{nm}(s_n(t), \boldsymbol{\nu}_{t-1})$ for each user n and layer l by Eq. (12);
5:   Solve the maximization problem (13) and obtains u_{nm};
6:   Schedule tasks according to $y_{nm}(t) = u_{nm}$;
7:   Get state updates from edge servers and actions;
8:   Update cost $\boldsymbol{\nu}_t \leftarrow (1-\beta)\boldsymbol{\nu}_{t-1} + \beta\boldsymbol{\nu}'_{t-1}$, where $\boldsymbol{\nu}_t$ is the server cost at time t, and $\boldsymbol{\nu}'_t$ is the optimal dual variable associate with problem (13b).
9: **end for**
___

B. Nested Index Policy

Based on the *nested index* derived at each time slot, the central actuator can schedule the tasks of all users following the nested index policy. Define u_{nm} as the decision variable for user n on server m, and $w_{nm} = I_{nm}(s_n(t), \boldsymbol{\nu}(t))$ is the decision weight given by the nested index at time t. We will solve the following binary decision scheduling problem at each time slot $t \in \mathcal{T}$:

$$\max_{\boldsymbol{u}} \sum_{n \in \mathcal{N}} \sum_{m \in \mathcal{M}} I_{nm}(s_n(t), \boldsymbol{\nu}_{t-1}) u_{nm} \tag{13a}$$

$$s.t. \sum_{n \in \mathcal{N}} u_{nm} \leq 1, \qquad \forall m \in \mathcal{M}, \tag{13b}$$

$$\sum_{m \in \mathcal{M}} u_{nm} \leq 1, \qquad \forall n \in \mathcal{N}, \tag{13c}$$

$$u_{nm} \in \{0, 1\}, \forall n \in \mathcal{N}, \qquad \forall m \in \mathcal{M}. \tag{13d}$$

We then make offloading decisions according to $y_{nm}(t) = u_{nm}$ at the solution. The decision variable u_{nm} represents the policy that maps from the current state $s_n(t)$. The mapping process from the current state $s_n(t)$ to the action variable $y_{nm}(t)$ is named the *nested index policy*.

In Algorithm 1, we compute nested index $I_{nm}(s_n(t), \boldsymbol{\nu}_{t-1})$ for each user via Eq. (12) and obtain the optimal solution $y_{nm}(t)$ for the problem (13) which is a simple linear programming problem. Next, the solution for problem (13) is mapped to offloading decisions and computing decisions to schedule tasks. We also get state updates from edge servers, offloading decisions, and computing decisions. Finally, we update activating cost $\boldsymbol{\nu}_t$. We execute lines 4-8 process at each time slot until the nested policy converges.

The computation of the index value can be very complex, and we gave an approximation of the nested index given s_n, $\boldsymbol{\nu}_{-m}$, $\forall\, 0 < m < M$.

Proposition 2. *Given* $s_n(t) = (\Delta_n(t), D_n(t))$, *the index function satisfies*

$$I_{nm}(s_n(t), \boldsymbol{\nu}) = \nu_{m-1} + \Delta_n(t) - \gamma_n^*. \tag{14}$$

The index for server m *can be derived by solving* $I_{nm}(s_n(t), \boldsymbol{\nu}) = \nu_m$.

The proof is given in Appendix C. The index for other layers can be similarly derived within finite steps of computation. We derive the optimal average cost γ_n^* by the technique similar to that used in [9], which involves solving a set of a finite number of equations. This reduces the complexity when computing the index function and makes our algorithm more feasible.

C. Fluid Limit Model

We use a fluid limit argument to show the optimality for the index policy in Algorithm 1 as in [30]. The fixed point solution is the solution for the fluid limit model of the original problem. We will show that the fixed point of the fluid limit model for problem (13) is equivalent to that of problem (6).

We define the fluid fixed point and the fluid limit model as follows. Let $z_{ns} \in [0, 1]$ denote the fraction of user n in state s, where $\sum_{s \in \mathcal{S}} z_{ns} = 1$. Let $x_{nm}^s \in [0, 1]$ denote the fraction of user n combined with state s at server m given by the optimal solution of the relaxed problem (6). Let $\boldsymbol{\nu}^*$ is the associated dual variable when it converges, and $(\boldsymbol{x}^*, \boldsymbol{z}^*, \boldsymbol{\nu}^*)$ represent the *fluid fixed point* of the following *fluid limit reformulation* of problem (6):

$$\min_{\boldsymbol{x}, \boldsymbol{z}} \sum_{n \in \mathcal{N}} \sum_{s \in \mathcal{S}} \sum_{m \in \mathcal{M}} z_{ns} C_{ns} x_{nm}^s \tag{15a}$$

$$s.t. \sum_{n \in \mathcal{N}} \sum_{s \in \mathcal{S}} z_{ns} x_{nm}^s \leq 1, \qquad \forall m \in \mathcal{M}, \tag{15b}$$

$$\sum_{m \in \mathcal{M}} x_{nm}^s \leq 1, \qquad \forall n \in \mathcal{N}, \forall s \in \mathcal{S}, \tag{15c}$$

$$\sum_{s \in \mathcal{S}} z_{ns} = 1, \qquad \forall n \in \mathcal{N}, \tag{15d}$$

$$z_{ns}, x_{nm}^s \in [0, 1], \forall n \in \mathcal{N}, \quad \forall m \in \mathcal{M}, \forall s \in \mathcal{S}, \tag{15e}$$

$$\sum_{s' \in \mathcal{S}} z_{ns} \sum_{m \in \mathcal{M}} x_{nm}^s q_{nm}^{ss'} = \sum_{s' \in \mathcal{S}} z_{ns'} \sum_{m \in \mathcal{M}} x_{nm}^{s'} q_{nm}^{s's},$$
$$\forall n \in \mathcal{N}, \forall s \in \mathcal{S}. \tag{15f}$$

where (15f) is a fluid balance constraint [30].

We can similarly derive the *fluid limit reformulation* problem (16) for the scheduling problem (13). Denote z'_{nm} as the fraction of user n in state s, v_{nm}^s as the fraction of user n assigned with server m under the index policy, and $\boldsymbol{\nu}'^*$ as the dual variable for the relaxed problem of the scheduling problem (13) when it converges. Let $(\boldsymbol{v}^*, \boldsymbol{z}'^*, \boldsymbol{\nu}'^*)$ be the fixed point solution, and we have:

$$\max_{\boldsymbol{v}, \boldsymbol{z}'} \sum_{n \in \mathcal{N}} \sum_{s \in \mathcal{S}} \sum_{m \in \mathcal{M}} z'_{ns} w_{nm}^s v_{nm}^s \tag{16a}$$

$$s.t. \sum_{n \in \mathcal{N}} \sum_{s \in \mathcal{S}} z'_{ns} v_{nm}^s \leq 1, \qquad \forall m \in \mathcal{M}, \tag{16b}$$

$$\sum_{m \in \mathcal{M}} v_{nm}^s \leq 1, \qquad \forall n \in \mathcal{N}, \forall s \in \mathcal{S}. \tag{16c}$$

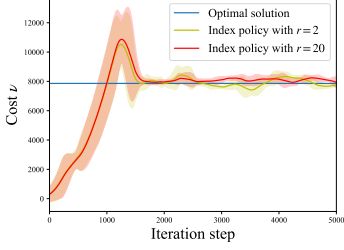Then, we can evaluate the performance of our *nested index* policy based on the fluid limit model for both problems.

Fig. 3: Dual cost update for the proposed index-based policy, which is compared with the dual cost update of the relaxed problem.

Fig. 4: Average AoI performance during computing.

Proposition 3. *The fixed point solution to problem (15) is equivalent to the solution (the fluid fixed point) to problem (16), i.e., we have*

$$(v^*, z'^*, \nu'^*) = (x^*, z^*, \nu^*). \tag{17}$$

The proof is shown in Appendix D. The equivalence of fixed point solution builds the connection between problem (13) and problem (6b). Though problem (13) follows an instantaneous constraint (13b), its fixed point solution still reaches the optimality at the fluid limit, which contributes to the *asymptotic optimality* of our policy in Algorithm 1.

By scaling a system by r, we scale the number of users N^r and servers M^r by r proportionally, i.e., let $N^r = r \cdot N$, $M^r = r \cdot M$ while keeping $\frac{N^r}{M^r}$ a constant.[1]

Lemma 1. *Under a mild global attractor assumption, the expected objective V_r^π for problem (13) under the nested index policy π achieves the optimal objective V^* for the fluid limit model of problem (6b) asymptotically, i.e.,*

$$\lim_{r \to +\infty} V_r^\pi = V^*. \tag{18}$$

We refer to [30] for the details of the global attractor assumption. The equivalence of the fixed point solution shows the accordance of the fluid limit model for both problems. Under a mild global attractor assumption, the objective of problem (5) under our *nested index* converges to the optimal cost of problem (15).

V. NUMERICAL RESULTS

In this section, we perform numerical studies to evaluate the performance of our nested index algorithm and verify the convergence property. We simulate a MEC system with initial tasks of $N = 50$ users that can be divided into 6 groups, with $\boldsymbol{\tau}^{min} = [2, 4, 8, 16, 32, 64]$, and successful updating probability of $\boldsymbol{p} = [0.8, 0.7, 0.6, 0.5, 0.3, 0.1]$, each of which has the number of $[5, 10, 5, 5, 10, 15]$ users respectively. We set $\beta = 50$ and simulated $T = 10000$ slots. In the simulation, we mainly test the performance of the *nested index* when solving a multi-layer MDP.

[1]The system parameters $\boldsymbol{\tau}$ and $\boldsymbol{p}$ are also scaled proportionally, i.e., $\boldsymbol{\tau}^r = [\boldsymbol{\tau}, \boldsymbol{\tau}, \ldots, \boldsymbol{\tau}] \in \mathbb{Z}_+^{1 \times N^r}$ and $\boldsymbol{p}^r = [\boldsymbol{p}', \boldsymbol{p}', \ldots, \boldsymbol{p}'] \in [0,1]^{M^r \times N^r}$, where $\boldsymbol{p}' = [\boldsymbol{p}^T, \boldsymbol{p}^T, \ldots, \boldsymbol{p}^T]^T \in [0,1]^{M^r \times N}$.
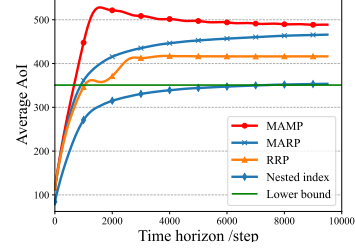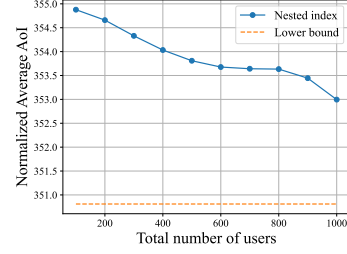

Fig. 5: Average AoI vs. system scale.

A. Convergence of the Cost Update

We compare the dynamics of the cost updates of the proposed nested index policy with that of the optimal solution to the problem (6). The optimal solution in Fig. 3 represents the dynamics of the server cost ν of the relaxed problem. We obtained a new cost $\nu(t)$ at each time slot by dual gradient ascent. In Fig. 3, the cost of the server smoothly converges to a small neighborhood of the optimal cost.

Next, we verify the server cost dynamics of the proposed index-based policy when the system scalar increases. Fig. 3 represents the dynamics of the cost update at scale $r = 2$ and $r = 20$. With the increase of the scalar r, the cost for the proposed index-based policy approaches is close to the optimal value of the dual cost.

B. Average AoI

We evaluate the average AoI performance of our proposed policy. We use the optimal solution to problem (6) as the lower bound of the index policy [30] for comparison.

We consider three benchmark policies in the experiments: *Max-Age Matching Policy* (MAMP), *Max-Age Reducing Policy* (MARP), which are both greedy policies, and *Rounded Relax Policy* (RRP).

- The MAMP chooses users with the highest current AoI.
- The MARP takes the transition probability of MC into consideration. Define the weight in this policy as $w_n = \Delta_n(t) + \frac{1}{p_n}(\Delta_n(t) - D_n(t))$, which represents the probable approximation of optimality gap reduction.
- The RRP is derived from the solution of the relaxed problem. The RRP chooses users uniformly at random to satisfy the feasibility when violating the constraint (2a).

Fig. 4 evaluates the average AoI under different policies such as nested index policy, MAMP, MARP, RRP, and lower

bound of problem (6) with $r = 20$. The greedy policy MAMP and MARP is 40% worse than the *nested index* policy and RRP is 21% worse than our approach when $r = 20$. The normalized system AoI gets closer to that of the optimal AoI for the relaxed problem with the increase of the system scalar r. Fig. 5 shows the normalized AoI of the system, i.e., the average age per user. The normalized AoI decreases almost monotonically as r increases even, which supports the asymptotic optimality of our proposed policy.

VI. CONCLUSION

In this study, we explored the minimization of AoI in a MEC system with heterogeneous servers and users. We formulated the problem as a two-layer MDP and introduced a novel nested index. We devised a scheduling policy that employs the nested index, ensuring the asymptotic optimality of the average expected AoI of the MEC system as the system scale expands. We also derived the computation of the nested index, which exhibits lower computational complexity. Through simulation, we demonstrated that our algorithm converges and delivers near-optimal performance.

REFERENCES

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

[2] V. Tripathi, R. Talak, and E. Modiano, "Age optimal information gathering and dissemination on graphs," *IEEE Transactions on Mobile Computing*, pp. 1–1, 2021.

[3] Q. Mi, N. Yang, H. Zhang, H. Zhang, and J. Wang, "Joint caching and transmission in the mobile edge network: An multi-agent learning approach," in *2021 IEEE Global Communications Conference (GLOBECOM)*, 2021, pp. 1–6.

[4] M. Tang and V. W. Wong, "Deep reinforcement learning for task offloading in mobile edge computing systems," *IEEE Trans. Mobile Computing*, 2020.

[5] Z. Ning, P. Dong, X. Wang, X. Hu, L. Guo, B. Hu, Y. Guo, T. Qiu, and R. Y. K. Kwok, "Mobile edge computing enabled 5g health monitoring for internet of medical things: A decentralized game theoretic approach," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 2, pp. 463–478, 2021.

[6] I. Kadota and E. Modiano, "Age of information in random access networks with stochastic arrivals," in *IEEE INFOCOM 2021-IEEE Conference on Computer Communications*. IEEE, 2021, pp. 1–10.

[7] R. D. Yates and S. K. Kaul, "The age of information: Real-time status updating by multiple sources," *IEEE Transactions on Information Theory*, vol. 65, no. 3, pp. 1807–1827, 2019.

[8] I. Kadota, A. Sinha, E. Uysal-Biyikoglu, R. Singh, and E. Modiano, "Scheduling policies for minimizing age of information in broadcast wireless networks," *IEEE/ACM Transactions on Networking*, vol. 26, no. 6, pp. 2637–2650, 2018.

[9] V. Tripathi and E. Modiano, "A whittle index approach to minimizing functions of age of information," in *2019 57th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. IEEE, 2019, pp. 1160–1167.

[10] K. Liu and Q. Zhao, "Indexability of restless bandit problems and optimality of whittle index for dynamic multichannel access," *IEEE Transactions on Information Theory*, vol. 56, no. 11, pp. 5547–5567, 2010.

[11] Y.-P. Hsu, "Age of information: Whittle index for scheduling stochastic arrivals," in *2018 IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2018, pp. 2634–2638.

[12] Y.-P. Hsu, E. Modiano, and L. Duan, "Scheduling algorithms for minimizing age of information in wireless broadcast networks with random arrivals," *IEEE Transactions on Mobile Computing*, vol. 19, no. 12, pp. 2903–2915, 2019.

[13] J. Liu, Y. Mao, J. Zhang, and K. B. Letaief, "Delay-optimal computation task scheduling for mobile-edge computing systems," in *2016 IEEE international symposium on information theory (ISIT)*. IEEE, 2016, pp. 1451–1455.

[14] J. Gittins, K. Glazebrook, and R. Weber, *Multi-armed bandit allocation indices*. John Wiley & Sons, 2011.

[15] Y. Zou, K. T. Kim, X. Lin, and M. Chiang, "Minimizing age-of-information in heterogeneous multi-channel systems: A new partial-index approach," in *Proceedings of the twenty-second international symposium on theory, algorithmic foundations, and protocol design for mobile networks and mobile computing*, 2021, pp. 11–20.

[16] S. Kaul, M. Gruteser, V. Rai, and J. Kenney, "Minimizing age of information in vehicular networks," in *2011 8th Annual IEEE communications society conference on sensor, mesh and ad hoc communications and networks*. IEEE, 2011, pp. 350–358.

[17] Y. Sun, E. Uysal-Biyikoglu, R. D. Yates, C. E. Koksal, and N. B. Shroff, "Update or wait: How to keep your data fresh," *IEEE Transactions on Information Theory*, vol. 63, no. 11, pp. 7492–7508, 2017.

[18] P. Whittle, "Restless bandits: Activity allocation in a changing world," *Journal of applied probability*, vol. 25, no. A, pp. 287–298, 1988.

[19] N. Yang, H. Zhang, K. Long, H.-Y. Hsieh, and J. Liu, "Deep neural network for resource management in noma networks," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 1, pp. 876–886, 2020.

[20] N. Yang, H. Zhang, and R. Berry, "Partially observable multi-agent deep reinforcement learning for cognitive resource management," in *GLOBECOM 2020 - 2020 IEEE Global Communications Conference*, 2020, pp. 1–6.

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

[22] K. Liu and Q. Zhao, "Indexability of restless bandit problems and optimality of whittle index for dynamic multichannel access," *IEEE Transactions on Information Theory*, vol. 56, no. 11, pp. 5547–5567, 2010.

[23] J. Ren, G. Yu, Y. Cai, Y. He, and F. Qu, "Partial offloading for latency minimization in mobile-edge computing," in *GLOBECOM 2017 - 2017 IEEE Global Communications Conference*, 2017, pp. 1–6.

[24] A. Arafa, R. D. Yates, and H. V. Poor, "Timely cloud computing: Preemption and waiting," in *2019 57th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, 2019, pp. 528–535.

[25] L. I. Sennott, "Average cost optimal stationary policies in infinite state markov decision processes with unbounded costs," *Operations Research*, vol. 37, no. 4, pp. 626–633, 1989.

[26] S. Boyd, S. P. Boyd, and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[27] D. P. Bertsekas, *Dynamic programming and optimal control. Volume 2*, fourth edition. ed. Belmont, Mass: Athena Scientific, 2012.

[28] A. Maatouk, S. Kriouile, M. Assad, and A. Ephremides, "On the optimality of the whittle's index policy for minimizing the age of information," *IEEE Transactions on Wireless Communications*, vol. 20, no. 2, pp. 1263–1277, 2020.

[29] C. Shuo, Y. Ning, and Z. Meng, "Minimizing age of information for mobile edge computing systems: A nested index approach – online appendix." [Online]. Available: https://www.dropbox.com/s/3v4tw2olpq4g8b6/appendix.pdf?dl=0

[30] I. M. Verloop, "Asymptotically optimal priority policies for indexable and nonindexable restless bandits," *The Annals of Applied Probability*, vol. 26, no. 4, pp. 1947–1995, 2016.

[31] D. R. Jiang and W. B. Powell, "An approximate dynamic programming algorithm for monotone value functions," *Operations research*, vol. 63, no. 6, pp. 1489–1511, 2015.

Minimizing Age of Information for Mobile Edge Computing Systems: A Nested Index Approach – Online Appendix

Shuo Chen, Ning Yang, Meng Zhang, Jun Wang

APPENDIX A

PROOF OF PROPOSITION 1

We drop subscript n to simplify the notation, as the proposition holds for each sub-problem. We suppose π^* is the optimal policy under the optimal server cost $\boldsymbol{\nu}^*$. We first show the monotonicity of the optimal cost-to-go function under π^*.

Lemma 2. *Denote the state of a user at layer 2 as $s = (A, D)$. The cost-to-go function under π^* is non-decreasing in A, i.e.,*

$$A \leq A', D = D' \Rightarrow V((A, D), \boldsymbol{\nu}) \leq V((A', D), \boldsymbol{\nu}),$$
$$V((A, D), \boldsymbol{\nu}) - V((A - D), \boldsymbol{\nu}) \quad (19)$$
$$\leq V((A', D), \boldsymbol{\nu}) - V((A' - D), \boldsymbol{\nu}).$$

The proof exploits Proposition 3.1 in [33], which extends the average cost within finite steps to the infinite horizon. We show the first inequality fulfills the condition of Proposition 3.1 in [33], and use backward induction to derive the second inequality. We consider the following state transition function $f : \mathcal{S} \times \mathcal{A} \times \mathcal{W} \to \mathcal{S}$, i.e., $s_n(t+1) = f(s_n(t), \boldsymbol{y}_n(t), w_t)$, where $w_t \in \mathcal{W}$ is the information process at time t [33]. Given $s' = (A', D')$ and $s \preccurlyeq s'$, we have:

1) For every $\boldsymbol{a} \in \mathcal{A}$ and $w \in \mathcal{W}$ the state transition function satisfies $f(s, \boldsymbol{a}, w) \preccurlyeq f(s', \boldsymbol{a}, w)$. Since at state s the next state could be either $(A+1, D)$ or $(A+1-D)$. We have $(A+1, D) \preccurlyeq (A'+1, D')$ and $A - D \leq A' - D$ at the same time given $D = D'$;
2) Denote $g_m(s) = C(s, \boldsymbol{a}) = \nu_m + (1 - p_m)A + p_m(A - D) = \nu_m + A - p_m \cdot D$ as the per stage cost. We also have $g_m(s) \leq g_m(s')$;
3) $w_{t+1} \in \mathcal{W}$ is independent of state $s \in \mathcal{S}$.

By Proposition 3.1 in [33], we conclude the inequality $V_T((A, D), \boldsymbol{\nu}) \leq V_T((A', D), \boldsymbol{\nu})$ of the T-stage cost minimization problem, where $V_T(\cdot)$ is the T-stage value function. By utilizing the convergence of the value iteration [28], [33], we can derive the first inequality in (19).

For the second inequality in (19), we also consider the T-stage value function. Given $V_{t+1}((A, D), \boldsymbol{\nu}) - V_{t+1}((A - D), \boldsymbol{\nu}) \leq V_{t+1}((A', D), \boldsymbol{\nu}) - V_{t+1}((A' - D), \boldsymbol{\nu})$. We have

$$\mathbb{E}\left[V_{t+1}(f((A, D), \boldsymbol{a}, w_{t+1}), \boldsymbol{\nu}) \mid S_t = (A, D), a_t = \boldsymbol{a}\right]$$
$$-\mathbb{E}\left[V_{t+1}(f((A - D), \boldsymbol{b}, w_{t+1}), \boldsymbol{\nu}) \mid S_t = (A - D), a_t = \boldsymbol{b}\right]$$
$$\leq \mathbb{E}\left[V_{t+1}(f((A', D), \boldsymbol{a}, w_{t+1}), \boldsymbol{\nu}) \mid S_t = (A', D), a_t = \boldsymbol{a}\right]$$
$$-\mathbb{E}\left[V_{t+1}(f((A' - D), \boldsymbol{b}, w_{t+1}), \boldsymbol{\nu}) \mid S_t = (A' - D), a_t = \boldsymbol{b}\right]. \quad (20)$$

We also have

$$V_t(s, \boldsymbol{\nu}) = \min_{\boldsymbol{a} \in \mathcal{A}} \left[C(s, \boldsymbol{a}) \right.$$
$$\left. + \mathbb{E}\left[V_{t+1}(f(s, \boldsymbol{a}, w_{t+1}), \boldsymbol{\nu}) \mid S_t = s, a_t = \boldsymbol{a}\right] \right]. \quad (21)$$

Since

$$C((A, D), \boldsymbol{a}) - C((A - D), \boldsymbol{b})$$
$$= \nu_m - \nu_{m'} + (1 - p_m)D \quad (22)$$
$$= C((A', D), \boldsymbol{a}) - C((A' - D), \boldsymbol{b}), \forall \boldsymbol{a}, \boldsymbol{b} \in \mathcal{A},$$

we can derive $V_t((A, D), \boldsymbol{\nu}) - V_t((A - D), \boldsymbol{\nu}) \leq V_t((A', D), \boldsymbol{\nu}) - V_t((A' - D), \boldsymbol{\nu})$ by adding (21) to (20). By using backward induction, it holds for all t.

Now, we will verify the MLTT property for sub-problem (6). For any user with state $s = A$ at layer 1, we compare the expected cost-to-go of choosing server m and m' when $p_m > p_{m'}$:

$$\mu_{m'}(A, \boldsymbol{\nu}) - \mu_m(A, \boldsymbol{\nu}) = \tau^{min} \cdot \nu_{m'}$$
$$+(1 - p_{m'})[A + \tau^{min} + V((A + \tau^{min} + 1, A), \boldsymbol{\nu})]$$
$$+p_{m'}V(\tau^{min}, \boldsymbol{\nu})$$
$$-\tau^{min} \cdot \nu_m \quad (23)$$
$$-(1 - p_m)[A + \tau_n^{min} + V((A + \tau^{min} + 1, A), \boldsymbol{\nu})]$$
$$-p_m V(\tau^{min}, \boldsymbol{\nu}).$$

Eq. (23) is a function of variable A, and can be rewritten as:

$$\mu_{m'}(A, \boldsymbol{\nu}) - \mu_m(A, \boldsymbol{\nu}) =$$
$$\cdots + (p_m - p_{m'})[A + V((A + \tau^{min} + 1, A), \boldsymbol{\nu}) \quad (24)$$
$$-V(\tau^{min} + 1, \boldsymbol{\nu})],$$

where we omit terms irrelevant to A. Since $p_m > p_{m'}$, Eq. (24) is strictly monotonically increasing in A. Therefore, there must exist a threshold $H(m, m', 0) = A^{m,m'}$, where we have $\mu_m(\cdot) = \mu_{m'}(\cdot)$.

For any user with state $s = (A, D)$ at layer 2, we have

$$\mu_{m'}((A, D), \boldsymbol{\nu}) - \mu_m((A, D), \boldsymbol{\nu}) =$$
$$\cdots + (p_m - p_{m'})[A + V((A + 1, D), \boldsymbol{\nu}) \quad (25)$$
$$-V(A - D + \tau^{min} + 1, \boldsymbol{\nu})].$$

The difference between the two cost-to-go is also monotonically decreasing in A, hence the sub-problem (6) is MLTT.

APPENDIX B

PROOF OF THEOREM 1

For simplicity, we specify the order of p_m, i.e., $p_{m-1} \leq p_m, \forall 1 < m \leq M$. To prove the intra-indexability, we have to introduce the following lemma:

Lemma 3. *Given server cost $\boldsymbol{\nu}$ and state $s = (A, D)$ at layer 2, denote $\boldsymbol{\nu}' = [\nu_1, \cdots, \nu_m + \Delta, \cdots, \nu_M], \forall \Delta \geq 0$. The difference between two cost-to-go functions given $\boldsymbol{\nu}$ nad $\boldsymbol{\nu}'$ can be upper-bounded by*

$$V_n(s, \boldsymbol{\nu}') - V_n(s, \boldsymbol{\nu}) < \frac{\Delta}{p_m^2}, \forall 1 \leq m \leq M. \quad (26)$$

Denote $p_{M+1} \triangleq 1$. Under the condition in lemma 3, the difference between two cost-to-go function can be lower-bounded by

$$V_n(s, \boldsymbol{\nu}') - V_n(s, \boldsymbol{\nu}) > -\frac{\Delta}{p_{m+1}^2},$$
$$\forall, 1 \leq m \leq M, A \geq H_n(m, m+1, D). \quad (27)$$

The proof is shown as follows. The minimizing AoI problem can be seen as a stochastic shortest path (SSP) problem [28].

Recall the optimal cost for user n is denoted as γ_n^*. The cost-to-go function can be rewritten as:

$$V_n(s, \boldsymbol{\nu}) = \min_\pi \mathbb{E}_\pi[\text{the cost from state } s \text{ to the recurrent state}$$
$$\text{for the first time}] -$$
$$\mathbb{E}_\pi[\text{the cost from state } s \text{ to the recurrent state}$$
$$\text{with stage cost } \gamma_n^*]. \tag{28}$$

Unlike appendix B in [18], the recurrent state for reference is not defined as the state with minimum AoI, but we can set any state at layer 1 as a reference recurrent state with zero cost-to-go. The expected cost from $s = (A, D)$ to recurrent state ζ given policy π_n and server cost ν can be denoted as $\text{Cost}_{s\zeta}^{\pi_n}(\boldsymbol{\nu})$, and the expected step can be denoted as $N_{s\zeta}^{\pi_n}$. Considering recurrent states $\zeta = (j), \forall j \geq A - D$, we have [18]

$$V_n(s, \boldsymbol{\nu}') - V_n(s, \boldsymbol{\nu}) \leq \text{Cost}_{s\zeta}^{\pi_n}(\boldsymbol{\nu}') - \text{Cost}_{s\zeta}^{\pi_n}(\boldsymbol{\nu}). \tag{29}$$

where

$$\text{Cost}_{s\zeta}^{\pi_n}(\boldsymbol{\nu}') - \text{Cost}_{s\zeta}^{\pi_n}(\boldsymbol{\nu})$$
$$= \Delta \cdot [\text{the expected time of hitting state } \tilde{s}, \forall \pi_n(\tilde{s}) = m$$
$$\text{when transiting from s to } \zeta \text{ under policy } \pi_n]. \tag{30}$$

According to Proposition 1, there exits $H_n(m-1, m, D), \forall D$. We can derive the upper bound by considering the following policy π_n': (a) for all $s = (A, D), D < H_n(m-1, m, 0)$ and $A >= \max_d H_n(m-1, m, d)$, $\pi_n'(s) = m$ and for any other $s', \pi_n'(s') = 1$. By using such a policy π_n', we have $p_{\pi_n'(s)} <= p_{\pi_n(s)}$, i.e., the probability of finishing computation at each state gets lower, and for all s and $|\{s \mid \pi_n'(s) = m\}| > |\{s \mid \pi_n(s) = m\}|$. Therefore, we have

$$\text{Cost}_{s\zeta}^{\pi_n}(\boldsymbol{\nu}') - \text{Cost}_{s\zeta}^{\pi_n}(\boldsymbol{\nu}) \leq \Delta \cdot N_{s\zeta}^{\pi_n'}. \tag{31}$$

We have

$$N_{s\zeta}^{\pi_n'} = p_m \cdot 1 + \sum_{i=2}^\infty p_m(1 - p_m)^{i-1}(i + N_{s\zeta}^{\pi_n'}),$$
$$N_{s\zeta}^{\pi_n'} = \frac{1}{p_m^2}. \tag{32}$$

Therefore,

$$V_n(s, \boldsymbol{\nu}') - V_n(s, \boldsymbol{\nu}) \leq \frac{\Delta}{p_m^2}. \tag{33}$$

Similar to Lemma 4.4 in [18], we also have

$$V_n(s, \boldsymbol{\nu}') - V_n(s, \boldsymbol{\nu}) \geq (\gamma_n^* - \gamma_n^{*'}) \cdot N_{s\zeta}^{\pi_n} \geq -\frac{\Delta}{p_{m+1}^2}, \tag{34}$$
$$\forall A \geq H_n(m, m+1, D),$$

where $\gamma^{*'}$ is the optimal stage cost under server cost $\boldsymbol{\nu}'$.

To prove the sub-problem (6) is intra-indexable, we have to show the following two claims:

- If $s = (A, D) \in \mathcal{P}_{nm}^l(\boldsymbol{\nu})$, then $s \in \mathcal{P}_{nm}^l(\boldsymbol{\nu}')$ must hold for $\boldsymbol{\nu}' = [\nu_1, \cdots, \nu_m + \Delta, \cdots, \nu_M], \forall \Delta \geq 0, m \in \mathcal{M}$.
- If $\nu_m \to +\infty$, then $\lim_{\nu_m \to +\infty} \mathcal{P}_{nm}^l(\boldsymbol{\nu}') = \mathcal{S}_l$.

We have shown that the optimal policy for the MDP is MLTT. Recall that the threshold splits server $m-1$ and m given generate age d is denoted as $H_n(m-1, m, d)$. Then, the passive set for layer 2 can be expressed as [18]:

$$\mathcal{P}_{nm}^l(\boldsymbol{\nu}) = \bigcup_{d=1}^\infty \{(a, d) \mid$$
$$a \in \{1, \cdots, H_n(m-1, m, d) - 1\} \cup \{H_n(m, m+1, d), \cdots\}\},$$
$$\forall 1 < m < M, \tag{35}$$

$$\mathcal{P}_{nM}^l(\boldsymbol{\nu}) = \bigcup_{d=1}^\infty \{(a, d) \mid a \in \{1, \cdots, H_n(M-1, M, d) - 1\}\}. \tag{36}$$

To show statement (i), we want to show $\mathcal{P}_{nm}^l(\boldsymbol{\nu}) \subseteq \mathcal{P}_{nm}^l(\boldsymbol{\nu}')$, we can instead show that for all $d \geq 1$:

(a) $H_n(m-1, m, d) \leq H_n'(m-1, m, d), \forall m \leq M$;
(b) $H_n(m, m+1, d) \geq H_n'(m, m+1, d), \forall m \geq M$;

We first show (a). For contradiction, suppose $H_n(m-1, m, d) > H_n'(m-1, m, d)$. At state $(H_n'(m-1, m, d), d)$, taking server $m-1$ has a smaller expected cost, and we have:

$$(p_m - p_{m-1})[H_n' + V_n(H_n', \boldsymbol{\nu}) - H_n' + d$$
$$- V_n((H_n' + 1 - d, H_n' + 1 - d), \boldsymbol{\nu})] \leq \nu_m - \nu_{m-1}, \tag{37}$$

where we denote $H_n = H_n(m-1, m, d) + 1, d)$ and $H_n' = H_n'(m-1, m, d) + 1, d)$ for simplicity. If given $\boldsymbol{\nu}'$, taking server $m-1$ has a smaller expected cost, i.e.,

$$(p_m - p_{m-1})[H_n' + V_n((H_n' + 1, d), \boldsymbol{\nu}') - H_n' + d$$
$$- V_n((H_n' + 1 - d, H_n' + 1 - d), \boldsymbol{\nu}')] \leq \nu_m' - \nu_{m-1}'. \tag{38}$$

Let Eq. (38) - Eq. (37), we have

$$V_n((H_n' + 1, d), \boldsymbol{\nu}') - V_n((H_n' + 1, d), \boldsymbol{\nu})$$
$$- [V_n((H_n' + 1 - d, H_n' + 1 - d), \boldsymbol{\nu}')$$
$$- V_n((H_n' + 1 - d, H_n' + 1 - d), \boldsymbol{\nu})] \tag{39}$$
$$\geq \frac{\Delta}{p_m - p_{m-1}},$$

when $p_m - p_{m-1} \leq p_m^2$, it contradicts the upper bound in Lemma 3.

Then, we will show the sufficiency of condition (b). Suppose $H_n(m-1, m, d) \leq H_n'(m-1, m, d)$. At state $s = (H_n'(m-1, m, d) - 1, d)$, policy π_n prefer a server $m-1 > m$ over server m, i.e., $H_n'(m-1, m, d) \geq H_n(m-1, m, d) \geq H_n(m, m+1, d)$, and we have

$$(p_{m+1} - p_m)[H_n' + V_n((H_n' + 1, d), \boldsymbol{\nu})] \leq \nu_{m+1} - \nu_m. \tag{40}$$

Similarly, policy π_n' prefers server m at state $s = (H_n'(m, m+1, d) - 1, d)$, i.e.,

$$(p_{m+1} - p_m)[H_n' + V_n((H_n' + 1, d), \boldsymbol{\nu}')]$$
$$\leq \nu_{m+1}' - \nu_m' = -\Delta. \tag{41}$$

For $p_{m+1} - p_m > 0$, we have

$$V_n((H_n' + 1, d), \boldsymbol{\nu}') - V_n((H_n' + 1, d), \boldsymbol{\nu})$$
$$\leq -\frac{\Delta}{p_{m+1} - p_m}, \tag{42}$$

when $p_{m+1}-p_m \leq p_{m+1}^2$, contradicts the lower bound derived in lemma 4. Therefore, the cardinality of passive set $\mathcal{P}_{nm}(\boldsymbol{\nu})$ grow monotonically to $|\mathcal{S}|$ as server cost ν_m increases from 0 to $+\infty$. Here we consider a strict condition that $p_m - p_{m-1} \leq p_m^2, \forall 1 < m \leq M$ due to that we obtain a rough upper bound on $N_{s\zeta}^{\pi'_n}$. In future studies, we will seek a tighter bound on $N_{s\zeta}^{\pi'_n}$.

APPENDIX C

PROOF OF PROPOSITION 2

Appendix A shows that the problem (6) satisfies the MLTT property. For simplicity, we denote the minimal age to offload tasks to server m for a user at layer 1 as H_m^* and the optimal age to offload tasks to server m for a user at layer 2 given generated age d as $H_m^*(d)$, and the minimum computational time is 1. Let the cost-to-go for the recurrent state $s = (1)$ to be 0, i.e., . We first focus on the cost-to-go for one state at layer 1 with age $H_{m-1}^* \leq A \leq H_m^*$:

$$
\begin{aligned}
V(A, \boldsymbol{\nu}) =& A - \gamma^* + \nu_{m-1} + p_{m-1} \cdot 0 \\
& + (1 - p_{m-1})V((A+1, A), \boldsymbol{\nu}).
\end{aligned}
\tag{43}
$$

For $A_M^* \leq A$, we assert that the optimal server for state $s = (A+1, A)$ is also M, and we have

$$
\begin{aligned}
V(A, \boldsymbol{\nu}) =& A - \gamma^* + \nu_M \\
& + (1 - p_M)V((A+1, A), \boldsymbol{\nu}) \\
=& \frac{A - \gamma^* + \nu_M}{p_M} + \frac{1 - p_M}{p_M^2}.
\end{aligned}
\tag{44}
$$

For $H_{m-1}^* \leq A \leq H_m^*$, we have

$$
\begin{aligned}
V(A, \boldsymbol{\nu}) =& \sum_{i=1}^{A_m^*(A) - A*_{m-i}} (1 - p_{m-1})^{i-1} \\
& \cdot (A*_{m-1} + i - 1 + p_{m-1}V(i, \boldsymbol{\nu}) + \nu_{m-1} - \gamma^*) \\
& + (1 - p_{m-1})^{A_m^*(A) - 1}V((A_m^*(A), A), \boldsymbol{\nu}),
\end{aligned}
\tag{45}
$$

and we have

$$
\begin{aligned}
V((A_m^*(A), A), \boldsymbol{\nu}) =& \sum_{i=1}^{A_{m+1}^*(A) - A_m^*(A)} (1 - p_m)^{i-1} \\
& \cdot (A_m^*(A) + i - 1 + p_m V(i, \boldsymbol{\nu}) + \nu_m - \gamma^*) \\
& + (1 - p_m)^{A_{m+1}^*(A) - A_m^*(A) - 1}V((A_{m+1}^*(A), A), \boldsymbol{\nu}),
\end{aligned}
\tag{46}
$$

$$
V((A_M^*(A), A), \boldsymbol{\nu}) = \frac{A_M^*(A) - \gamma^* + \nu_M}{p_M} + \frac{1 - p_M}{p_M^2}.
\tag{47}
$$

For $A < H_1^*$, we have

$$
V(A, \boldsymbol{\nu}) = A + V(A+1, \boldsymbol{\nu}) - \gamma^*.
\tag{48}
$$

$$
V(1, \boldsymbol{\nu}) = 0.
\tag{49}
$$

Since $p_1, p_2, \ldots, p_M$ and $\boldsymbol{\nu}$ is given, we can derive the closed-form relationship of γ^* with $H_m^*, H_m^*(d), \forall m \in \mathcal{M}, d \leq A_M^*$ from Eq. (43) to Eq. (49). The *optimal threshold* follows:

$$
\begin{aligned}
\nu_{m-1} + V((A, D), \boldsymbol{\nu}) &\leq \nu_m + V((A, D-1), \boldsymbol{\nu}) \\
\nu_m + V((A, D-1), \boldsymbol{\nu}) &\leq \nu_m + V((A+1, D), \boldsymbol{\nu}) \\
\nu_m + V((A+1, D), \boldsymbol{\nu}) &\leq \nu_{m-1} + V((A+1, D+1), \boldsymbol{\nu}).
\end{aligned}
\tag{50}
$$

Subtracting $\nu_m + V((A, D-1), \boldsymbol{\nu}) - \gamma^*$ to every term above:

$$
\nu_{m-1} - \nu_m + A \leq \gamma^* \leq \nu_{m-1} - \nu_m + A + 1.
\tag{51}
$$

Let $\nu_{m-1} - \nu_m + A = \gamma^*$, we derive the expression of ν_m, which is the index function [12].

APPENDIX D

PROOF OF PROPOSITION 3

In Appendix B, we have shown that the subproblem (6) satisfies the *intra-indexability*. The asymptotic optimum will hold for the original problem under the nested index policy if *precise division* property satisfies [18]. The *precise division* property is defined as follows:

Definition 6 (Precise Division). *Given state s and the server cost $\boldsymbol{\nu}$, suppose the sub-problem is intra-indexable. We say the preference for server m is precisely divisible at layer l by the nested-index $I_{nm}(s, \boldsymbol{\nu}, l)$ if the following holds:*
  (i) If $I_{nm}(s, \boldsymbol{\nu}) = \nu_m$, then $\mu_{nm}(s, \boldsymbol{\nu}) \leq \mu_{nm'}(s, \boldsymbol{\nu})$.
  (ii) If $I_{nm}(s, \boldsymbol{\nu}) > \nu_m$, then $\mu_{nm}(s, \boldsymbol{\nu}) < \mu_{nm'}(s, \boldsymbol{\nu})$.
  (iii) Otherwise, there exists $m' \neq m$ s.t. $\mu_{nm}(s, \boldsymbol{\nu}) > \mu_{nm'}(s, \boldsymbol{\nu})$.

The precise division property established the connection between the index value and the optimal policy. Then, we will show that the sub-problem (6) satisfies the *precise division* property.

Proposition 4. *Given state $\mathcal{S}$ and server cost $\boldsymbol{\nu}$, the sub-problem (6) satisfies the precise division property defined in Definition 7.*

The precise division property is more strict than intra-indexability, due to the transition of optimal action happens when the index value coincides with the server cost. The author in [18] proves this proposition by showing the difference $V_n(s, \boldsymbol{\nu}') - V_n(s, \boldsymbol{\nu})$ is uniformly bounded:

Lemma 4. *Denote $p_{M+1} \triangleq 1$. Under the condition in lemma 3, the difference between two cost-to-go function can be lower-bounded by*

$$
V_n(s, \boldsymbol{\nu}') - V_n(s, \boldsymbol{\nu}) > -\frac{\Delta}{p_{m+1}}, \forall, 1 \leq m \leq M.
\tag{52}
$$

According to Case 3, Lemma 4.6 in [18], we have to show the monotonicity of $h(\pi'_n, s = (A, D), \boldsymbol{\nu}')$ in A, where

$$
h(\pi'_n, s = (A, D), \boldsymbol{\nu}') = \mu_{\pi'_n(s)}(s, \boldsymbol{\nu}') - \mu_m(s, \boldsymbol{\nu}') + \Delta - \gamma_n^{*\prime} + \gamma_n^*.
\tag{53}
$$

We have

$$\mu_{\pi'_n(s)}(s, \boldsymbol{\nu}') - \mu_m(s, \boldsymbol{\nu}') = \nu_{\pi'_n(s)} - \nu_m$$

$$+ (p_m - p_{\pi'_n(s)})(D + V((A+1, D), \boldsymbol{\nu}') - V((A+1-D), \boldsymbol{\nu}')), \tag{54}$$

where we have shown $V((A+1, D), \boldsymbol{\nu}') - V((A+1-D), \boldsymbol{\nu}')$ is monotonically increasing with A in Appendix A.