Only Those Requested Count: Proactive Scheduling Policies for Minimizing Effective Age-of-Information

Bo Yin*, Shuai Zhang*, Yu Cheng*, Lin X. Cai*, Zhiyuan Jiang†, Sheng Zhou‡, and Zhisheng Niu‡
*Department of Electrical and Computer Engineering, Illinois Institute of Technology, Chicago 60616, USA
†Shanghai Institute for Advanced Communication and Data Science, Shanghai University, Shanghai 200444, China
‡Department of Electronic Engineering, Tsinghua University, Beijing 100084, China
Email: {byin, szhang104}@hawk.iit.edu; {cheng, lincai}@iit.edu; {zhiyuan, sheng.zhou, niuzhs}@tsinghua.edu.cn

***Abstract*—Motivated by the increasingly urgent demands for delivering fresh information, the age-of-information (AoI) has recently been introduced as an important metric for evaluating the timeliness performance of information update systems and has shed light on a number of research studies. Nevertheless, the most common goal of the existing works does not characterize the value of information freshness from the users' perspective. In this paper, we introduce the concept of effective AoI (EAoI) to quantify the freshness of the information users utilize for decision-making. We consider a general request-response model, which captures both proactive information update and timely information delivery, for investigating the scheduling problem with respect to EAoI minimization. By decomposing the scheduling problem into multiple computationally tractable subproblems, we propose request-aware scheduling policies for static and dynamic request models, respectively. The numerical results show that serving users requests proactively can reduce time-average EAoI in both scenarios.**

I. Introduction

Recent years have witnessed a significant advancement of networking technologies as well as the proliferation of mobile devices. The convergence of pervasive connectivity and ubiquitous computing has spawned a plethora of real-time applications, boosting the demand for timely information updates. With dense IoT deployment, it is becoming increasingly common that different types of time-sensitive data are collected, analyzed and delivered to end users based on their personalized interests. For example, a plug-in hybrid electric vehicle (PHEV) can request the information about road conditions, e.g., traffic information, and current energy prices at different charging stations to adjust their charging strategies as well as the routes to their target stations. Therefore, the delivery of fresh information has become a major concern.

Age-of-information (AoI) is a recently proposed metric that quantifies the freshness of the knowledge we have about a remote system [1]. Formally, it is defined as the time elapsed since the latest information update. In [2], the concept of AoI was first introduced to capture the timeliness requirement of safety applications in vehicular networks that maintain the current state information of nearby nodes. Since its inception, AoI has attracted great attention and been studied in a variety of areas, ranging from information sampling in sensor networks [3]–[7], scheduling discipline in queue management systems [8]–[14] to link scheduling optimization in wireless networking [15]–[22].

Although considerable efforts have been made to explore the optimal schedule of information updates in various contexts, prior studies on AoI minimization have a common goal: trying to keep the AoI low all the times. However, we argue that the freshness of information is not of equal importance at any time. Specifically, information is valuable only when it enables certain decisions or actions. In this paper, we will use the term effective AoI (EAoI) to represent the age of the information that is associated with decision-making. With this in mind, in order to improve the value of information freshness from the user's perspective, users' request patterns are supposed to be taken into account in the development of scheduling policy for information update. Thanks to recent advances in data mining or machine learning technologies, predicting user requests and serving them proactively has become a promising method for enhancing system performance in a range of applications [23]–[26]. However, how to exploit the predictable requests and leverage the proactive serving mechanism in the control of information updates remains open.

In this paper, we aim to close the gap and explore how to take advantage of the knowledge about the requests for providing users with fresh information. To this end, we consider the scenario of a server delivers the information requested by users in a timely manner. With the objective of minimizing the time-average EAoI, we propose scheduling policies for the server to proactively update users' information, given different types of knowledge about the arrivals of future requests. Our simulation results demonstrate the benefit of proactive scheduling with request awareness. The main contributions of this paper can be summarized as follows.

(1) We propose a general request-response model for studying the scheduling problem with respect to EAoI minimization. This model captures both proactive information update and timely information delivery. To the best of our knowledge, such request-aware model has not been investigated in the existing work on timeliness optimization. And we believe the idea of proactive scheduling can be applied on top of existing approaches in improving the information freshness.

(2) We develop two computationally tractable scheduling policies for addressing the EAoI minimization problem in the context of static and dynamic request models, respectively. For the static scenario, we apply the restless multi-armed bandit (RMAB) framework to model the problem above and derive the Whittle's index policy [27]. For the dynamic scenario, we employ the principle of receding horizon control (RHC) to leverage the knowledge about users' requests. Casting the EAoI minimization problem in the dynamic scenario as a weakly coupled dynamic programs (WCDP) problem, we decompose this problem by dual decomposition technique and propose a low-complexity heuristic.
(3) We show using extensive simulations that the proposed approaches outperform conventional scheduling algorithms in terms of minimizing the time-average EAoI. The numerical results also provide insights into the qualitative relationships between different design parameters and the system performance.

The remainder of this paper is organized as follows. We describe the system model in Section II. The scheduling policies for static request model and dynamic model are derived in Section III and Section IV, respectively. Numerical results are presented in Section V. Finally, we conclude the paper in Section VII.

II. System Model

We consider a general scenario where a single server serves N users with time-sensitive information. A time-slotted system is considered. In a time slot, a user may make a query into the server and utilize the information he/she obtains to take certain actions, e.g., stock trading based on the stock price information. We assume that the information update with respect to one user takes one time slot. Due to resource constraints, the server can only update the information of K users in one slot. At the beginning of each slot, the server selects K users and initialize the update process for their information. Let $a_n(t) \in \{0,1\}$ indicate the server's decision on user n in slot t, where $a_n(t) = 1$ if user n is selected at that slot. Potential update failures are considered, which are modeled by independent and identically distributed Bernoulli processes. That is, the information of user n, if selected, will get updated successfully with probability q_n. For example, the server may update the information of one user by pulling data from a remote IoT hub via a wireless connection. In this case, q_n can be considered as the coverage probability of this communication link.

In this paper, we consider the scheduling policy under which the server updates the information requested by the users such that the users can make decisions based on the most up-to-date information. We employ the AoI metric to quantify the freshness of the information that the server has. Formally, let $h_n(t)$ denote the AoI with respect to user n at time t, its dynamic can be expressed as

$$h_n(t+1) = \begin{cases} 1 & \text{if } u_n(t) = 1 \\ h_n(t) + 1 & \text{otherwise} \end{cases} \tag{1}$$

where $u_n(t)$ is an indicator variable which represents whether or not user n's information gets updated at time slot t. Those variables are influenced by both the scheduling policy and the system randomness. Given that user n is selected at slot t, $u_n(t)$ follows Bernoulli distribution with mean q_n. According to Eq. (1), $h_n(t)$ grows linearly as time goes by and drops to 1 when a successful update occurs during the previous slot.

In order to deliver prompt response, the server serves the user requests in a proactive fashion. Specifically, given a user n who is not selected at slot t, his/her request will be responded immediately upon arrival. In this way, the EAoI with respect to this user equals to $h_n(t)$. On the other hand, if user n is selected at this slot, the server will deliver the most up-to-date information to this user at the end of slot t. Therefore, if the requested information gets updated successfully, the EAoI will be 1. Otherwise, the user has to make decisions based on information with age $h_n(t)+1$. Let $d_n(t)$ be the indicator variable that represents whether or not user n will make a query during slot t. According to the aforementioned service model, the EAoI with respect to user n at time t, denoted by $g_n(t)$, can be formulated as

$$g_n(t) = \begin{cases} d_n(t) & \text{if } u_n(t) = 1 \\ d_n(t) * (h_n(t) + a_n(t)) & \text{otherwise} \end{cases} \tag{2}$$

According to Eq. (2), $g_n(t)$ equals to 0 if user n does not make any query during that slot. The objective of the scheduling policy π is to minimize the expected time-average EAoI $J(\pi)$, i.e,

$$J(\pi) = \limsup_{T\to\infty} \frac{1}{TN} \mathbb{E}\left[\sum_{t=1}^{T}\sum_{n=1}^{N} g_n(t)\right] \tag{3}$$

It's worth noting that minimizing the time-average EAoI is in general different from minimizing the time-average AoI. In the following, we use a toy example to show that these two goals result in different scheduling policies. Consider a system that consists of three users and the server can perform one update per slot. The initial AoI with respect to user a, b, c are 3, 2, 1, respectively. Only one user will make query at a certain slot and the three users request information in cyclic order $[b, c, a]$. The information update processes are assumed to be error-free. Obviously, the optimal solution for minimizing the time-average EAoI is a round-robin schedule with the same cyclic order. Based on the results in [19], the age-greedy policy achieves the minimum time-average AoI in this simple setting. That is, information with the largest AoI gets updated at each slot, which leads to a round-robin schedule with cyclic order $[a, b, c]$. The dynamics of AoI (EAoI) with respect to each user are shown in Fig. 1. It can be seen in Fig. 1 that the time-average EAoI achieved by the age-greedy policy is the double of that of the optimal solution.

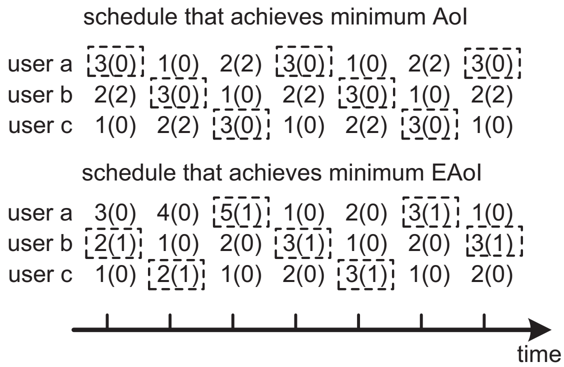


Fig. 1. Dynamics of AoI (EAoI) with respect to each user (the dashed box indicates the scheduled user)

The above example also shows that the knowledge of future requests plays an important role in the design of the scheduling policy. In this paper, the arrivals of user's requests are modeled by Bernoulli processes with rate $p_n(t)$. That is, at slot t, user n will make a query ($d_n(t) = 1$) with probability $p_n(t)$. Specifically, we focus on the following two request models.

(1) Static requests: In this case, variables $d_n(t)$ of different slots are independent and identically distributed. In other words, $p_n(t)$ is fixed in time. And we assume that such statistics are known to the server.
(2) Dynamic requests: In this case, $p_n(t)$ may change over time. In practical applications, near-term future user request patterns can be estimated by leveraging request history and certain context information, e.g., using machine learning methods. We assume that the server has access to a time series forecasting model which can provide the estimates of $p_n(t)$ in the next W slots, say $\{p_n(t), p_n(t+1), \ldots, p_n(t+W-1)\}$, for each user.

In what follows, we will develop scheduling policies for these two request models, respectively.

III. Scheduling Policy for Static Scenario

Generally, problem (3) can be formulated as a Markov Decision Process (MDP) with infinite horizon and countably infinite state space. Nevertheless, due to the curse of dimensionality, it is computationally prohibitive to derive the optimal policy by solving the MDP directly. In this section, we first formulate the scheduling problem in the static scenario as an RMAB problem. Following that, we develop the Whittle's index policy for that problem.

A. RMAB-based Problem Formulation

Applying the RMAB methodology to develop control policy requires formal definitions of state space, action space, transition and objective of the model. In the scheduling problem, users are considered as the arms in RMAB. The system state can be fully represented by the AoI of users, say $\mathbf{h}(t) = [h_1(t), h_2(t), \ldots, h_N(t)]$, $\mathbf{h}(t) \in \mathbb{N}^N$. The server, as the player, selects K out of N users and update their information. The action space can be characterized by $\mathbf{a}(t) = [a_1(t), a_2(t), \ldots, a_N(t)]$.

As the generalization to the classic multi-armed bandit (MAB), RMAB allows the arm to change its state even when it is not being operated. In general, the Markovian rule that the passive arms follow is different from that of the active arms. Take user n as an example, when not selected, the state transition is deterministic, i.e.,

$$\Pr(h_n(t+1) = h_n(t) + 1 | h_n(t)) = 1 \tag{4}$$

If selected, the state transition probability with respect to user n is described as

$$\begin{aligned} &\Pr(h_n(t+1) = 1 | h_n(t)) = q_n \\ &\Pr(h_n(t+1) = h_n(t) + 1 | h_n(t)) = 1 - q_n \end{aligned} \tag{5}$$

The objective of the RMAB problem is the same as (3), i.e., minimizing the expected time-average EAoI.

One important feature of the RMAB problem is that the arms are weakly coupled through the capacity constraint, i.e., only K users can be selected in each slot. The state transition of one arm has no impact on that of other arms. Such property facilitates the development of decomposable index policy, which has remarkably low computational complexity.

B. Whittle's Index Policy

The Whittle's index policy is derived from the optimal solution to the relaxation of the RMAB problem, in which the capacity constraint is relaxed to long-term average version. The relaxed RMAB can be decomposed into N single-armed bandit processes, a.k.a., decoupled model. In the decoupled model, a constant cost, denoted by C, will be incurred to the arms that are selected at each slot. Those single-armed bandit processes are investigated separately and, at each slot, an index value can be calculated for each arm. The K arms with the highest indices will be selected by the player. With mild conditions, the Whittle's index policy is proven to be asymptotically optimal [28]. Near-optimal performances achieved by this approach have also been demonstrated in a variety of applications [29], [30].

In the decoupled model, the state space, action space and state transition probability are the same as the original RMAB. Regarding the objective function, the constant cost C is included. Since the N single-armed problem can be treated separately, we can focus on one of those N subproblems. Thereafter in this subsection, we omit the user index. The objective function with a policy ϕ can be written as

$$\hat{J}(\phi) = \limsup_{T\to\infty} \frac{1}{T} \mathbb{E}\left[\sum_{t=1}^{T} (g(t) + Ca(t))\right] \tag{6}$$

By verifying the conditions in [31], it can be shown that a deterministic and stationary optimal policy ϕ^* exists for the above single-armed problem.

In order to apply the Whittle's index policy, the arms of the RMAB must satisfy an indexability condition. Given the cost

C, let $\mathcal{U}(C)$ denote the set of states in which the user will not be selected by policy ϕ^*.

Definition 1. (Indexability) The arm associated with a user is indexable if $\mathcal{U}(C)$ increases monotonically from $\emptyset$ to $\mathbb{N}$ as C increases from 0 to ∞.

To demonstrate the indexability of the scheduling problem, we derive an optimal policy ϕ^* by solving the Bellman equations associated with the single-armed bandit problem, which are expressed as

$$\begin{aligned} f(h)+\mu = \min\{& f(h+1)+ph; \\ & C+q(f(1)+p)+(1-q)(f(h+1)+p(h+1))\} \end{aligned} \tag{7}$$

where $f(\cdot)$ is the differential cost-to-go function and we prescribe $f(1) = 0$. μ is the optimal value of $\hat{J}(\phi)$. The Bellman equations capture the conditions an optimal policy is supposed to satisfy. On the right hand side (RHS) of Eq. (7), the upper part is the expected outcome of idling, i.e., $a = 0$ while the lower part corresponds to the expected outcome when an update is scheduled. At each slot, the action that results in the outcome with the minimum expected value will be taken, with tie-breaking arbitrarily. For the convenience of analysis, we break ties in favor of updating the information.

Remark. It is worth noting that the Whittle's index policy is also derived under a similar scenario in [19], the objective of which is time-average AoI minimization. Besides the involvement of the request probability, the major difference between our formulation in (7) and its counterpart in [19] is the immediate cost. More precisely, regardless of the extra cost C, both actions result in the same immediate cost in [19]. Since proactive serving is considered in our formulation, the immediate costs incurred by two actions are different and the relation between these two values is equivocal, determined by the values of p, q, and h.

Lemma 1. *Given the cost C, there exists a threshold-type policy which is optimal solution to the single-armed bandit problem. That is, there exists a threshold H and it is optimal for the server to update user's information when $h \geq H$ while keeping idle when $1 \leq h < H$. Specifically, the threshold is given by*

$$H = \left\lceil \frac{1}{2} - \frac{1}{q} + \sqrt{(\frac{1}{q}+\frac{1}{2})^2 + \frac{2C}{pq}} \right\rceil \tag{8}$$

Proof: The proof of Lemma 1 is constructive. By applying the proof technique in [19], we demonstrate that the threshold-type policy with the threshold value in (8) is the solution to the Bellman equations. See Appendix for details. ■

With the optimal threshold H, we are ready to establish the indexability of the RMAB problem and derive the closed form of the Whittle's Index.

Theorem 1. *The arms of the RMAB problem formulated in section III-A are indexable.*

Proof: Under the threshold-type policy, $\mathcal{U}(C) = \{h : 1 \leq h < H\}$. Since the threshold H is monotonically increasing with C, $\mathcal{U}(C)$ is also monotonically increasing with C. In addition, $\mathcal{U}(C)$ increases to $\mathbb{N}$ as $C \to \infty$. When $C = 0$, $H = 1$, which implies that $\mathcal{U}(0) = \emptyset$. According to the definition of indexability, we conclude the proof. ■

Let $I(h)$ denote the Whittle's index in state h. The value of $I(h)$ is defined as the infimum cost C that makes both actions equally attractive in state h.

Theorem 2. *The Whittle's index of the single-armed bandit problem is given by*

$$I(h) = \frac{1}{2}p(qh+2)(h-1) \tag{9}$$

Proof: Since the $\mathcal{U}(C)$ is monotonically increasing with C, there exists a unique cost C that makes both actions equally beneficial in state h. Consider that the tie is broken in favor of updating the information, we have

$$h = \frac{1}{2} - \frac{1}{q} + \sqrt{(\frac{1}{q}+\frac{1}{2})^2 + \frac{2C}{pq}} \tag{10}$$

In this way, the update will be activated at this moment since $h = H$. On the other hand, a larger C yields $H = h + 1$, in which case idling is more desirable. With appropriate manipulation, we obtain the value of Whittle's index described in (9) and conclude the proof. ■

Under the Whittle's index policy, the top K users in terms of their index value $I_n(h_n(t))$ will be selected by the server, with ties breaking arbitrarily. It can be seen that the Whittle's index policy reduces to the age-greedy policy under symmetric setting, in which $p_n = p$ and $q_n = q, \forall n$. Additionally, when users' request pattern are homogeneous, say $p_n = p, \forall n$, the Whittle index policy reduces to a request-aware policy which in some sense coincides with the results in [19].

IV. Scheduling Policy for Dynamic Scenario

In the dynamic scenario, we consider that the near-term prediction about future request arrivals is available to the server for decision-making, where finite-horizon optimization arises naturally. In this section, we employ the principle of RHC to develop a dynamic scheduling policy. To this end, we formulate a WCDP problem as the finite-horizon surrogate of problem (3). Afterward, we seek for a computationally appealing heuristic to enable the RHC.

A. RHC-based Policy

Recall that a time series forecasting model is assumed in the dynamic request model. Usually, the forecasting model is characterized by an intricate function, e.g., support vector machine (SVM) and artificial neural network (ANN), which maps past observations to possible future outcomes. In this way, the forecasting model will adjust its predictions based on the recent observations at the end of each slot. Due to its great adaptability to parametric changes, the RHC framework offers a promising methodology to address the scheduling problem. Basically, an RHC-based policy involves repeatedly solving a constrained optimization problem over a moving prediction window and commits only the first step in the resulting optimal control sequence.

For ease of presentation, let $\mathbf{h}(0)$ and $\mathbf{h}(W)$ denote the AoI status of all users at the beginning of current slot and at the end of the next W-th slot, respectively. Correspondingly, let $\{\mathbf{p}(0), \mathbf{p}(1), \ldots, \mathbf{p}(W-1)\}$ denote the predictions generated by the forecasting model. Given $\mathbf{h}(0)$, the finite-horizon optimization problem regarding EAoI minimization can be described as

$$\begin{aligned} \text{minimize} \quad & \mathbb{E}\left[\sum_{t=0}^{W-1}\sum_{n=1}^{N} g_n(t) + \beta\|\mathbf{h}(W)-1\|^2\right] \\ \text{s.t.} \quad & \sum_{n=1}^{N} a_n(t) = K, \forall t \end{aligned} \tag{11}$$

The term $\beta\|\mathbf{h}(W)-1\|^2$ in the objective function of problem (11) represents the terminal cost, which in some sense regulates the AoI status at the end of the horizon.

Problem (11) is a WCDP problem, which can be considered as analogous to the RMAB problem formulated in section III-A. More precisely, these two problems are associated with the same action space and transition dynamics. In contrast to the RMAB, problem (11) involves finite state space. Let $\mathcal{H}_n(t)$ denote the state space associated with user n at slot t, $\mathcal{H}_n(t) = \{1, 2, \ldots, t\} \cup \{h_n(0)+t\}$. Although finite, the state space of the whole problem grows exponentially with N. To address this issue, in the next subsection, we employ the dual decomposition technique to develop a low-complexity heuristic for the scheduling problem.

B. Lagrangian Heuristic

Prior to applying the decomposition technique, we relax the constraints in (11) to constraints in form of expectation, say $\mathbb{E}[\sum_{n=1}^{N} a_n(t)] = K$. Let $\boldsymbol{\lambda} = [\lambda_0, \lambda_1, \ldots, \lambda_{W-1}]$, $\boldsymbol{\lambda} \in \mathbb{R}^W$, be the Lagrange multiplier vector corresponding to those relaxed constraints. The Lagrangian associated with the relaxed problem can be expressed as

$$\begin{aligned} \mathcal{L}(\pi, \boldsymbol{\lambda}) &= \mathbb{E}\left[\sum_{t=0}^{W-1}\sum_{n=1}^{N} g_n(t) + \beta\|\mathbf{h}(W)-1\|^2\right] \\ &\quad + \sum_{t=0}^{W-1} \lambda_t(\mathbb{E}[\sum_{n=1}^{N} a_n(t)] - K) \\ &= \sum_{n=1}^{N} \mathbb{E}\left[\sum_{t=0}^{W-1} (g_n(t) + \lambda_t a_n(t)) + \beta(h_n(W)-1)^2\right] \\ &\quad - K\sum_{t=0}^{W-1}\lambda_t \end{aligned} \tag{12}$$

And the Lagrange dual function can be written as

$$D(\boldsymbol{\lambda}) = \inf_{\pi} \mathcal{L}(\pi, \boldsymbol{\lambda}) \tag{13}$$

According to duality theory [32], $D(\boldsymbol{\lambda})$ is concave and provides a lower bound on the optimal value of the original problem. Moreover, $\boldsymbol{\lambda}^* = \operatorname{argmax}_{\boldsymbol{\lambda}} D(\boldsymbol{\lambda})$ plays an important role in the development of Lagrangian heuristic. Since $D(\boldsymbol{\lambda})$ is piecewise linear in $\boldsymbol{\lambda}$, $\boldsymbol{\lambda}^*$ can be calculated by using a subgradient method [33], i.e., updating $\boldsymbol{\lambda}$ iteratively by

$$\lambda_t^{(k+1)} = \lambda_t^{(k)} + \alpha^{(k)}(\sum_{n=1}^{N} \mathbb{E}[a_n^{(k)}(t)] - K), \quad \forall t \tag{14}$$

where α^k is the step size and $\sum_{n=1}^{N} \mathbb{E}[a_n^{(k)}(t)]$ is determined by the policy $\pi^{(k)} = \operatorname{argmin}_\pi \mathcal{L}(\pi, \boldsymbol{\lambda}^{(k)})$. In the following, we show that $\mathbb{E}[a_n^{(k)}(t)]$ can be evaluated efficiently.

For a given $\boldsymbol{\lambda}^{(k)}$, $\mathcal{L}(\pi, \boldsymbol{\lambda}^{(k)})$ is separable. Therefore, $\pi^{(k)}$ is characterized by a collection of policies $\{\pi_n^{(k)}\}$, where $\pi_n^{(k)}$ is the solution to the following problem

$$\text{minimize} \ \mathbb{E}\left[\sum_{t=0}^{W-1} (g_n(t) + \lambda_t^{(k)} a_n(t)) + \beta(h_n(W)-1)^2\right] \tag{15}$$

By the abuse of notation, we represent $\pi_n^{(k)}$ as a table that contains $\{a_n^{(k)}(h,t)\}$, $h \in \mathcal{H}_n(t)$, where $a_n^{(k)}(h,t)$ denote the decision in state h at slot t. Problem (15) can be considered as a finite-horizon MDP, which can be solved through backward induction [34], i.e.,

$$\begin{aligned} F_n(h,t) = \min\{&F_n(h+1,t+1) + p_n(t)h; \\ &\lambda_t^{(k)} + q_n(F_n(1,t+1) + p_n(t)) \\ &+ (1-q_n)(F_n(h+1,t+1) + p_n(t)(h+1))\} \end{aligned} \tag{16}$$

and $F_n(h,W) = \beta(h-1)^2$, where $F_n(\cdot,\cdot)$ is the cost-to-go function.

Let $w_n^{(k)}(h,t)$ denote the probability that $h_n(t) = h$ under $\pi_n^{(k)}$. We can employ forward induction to calculate all the $w_n^{(k)}(h,t)$, i.e., $w_n^{(k)}(h,0) = 1$ and, for $h > 1, t > 0$,

$$w_n^{(k)}(h,t) = \begin{cases} (1-q_n)w_n^{(k)}(h-1,t-1) \\ \qquad \text{if } a_n^{(k)}(h-1,t-1) = 1; \\ w_n^{(k)}(h-1,t-1) \quad \text{otherwise} \end{cases} \tag{17}$$

and $w_n^{(k)}(1,t) = 1 - \sum_{h \in \mathcal{H}_n(t)\setminus 1} w_n^{(k)}(h,t)$. With $\{a_n^{(k)}(h,t)\}$ and $\{w_n^{(k)}(h,t)\}$, $\mathbb{E}[a_n^{(k)}(t)]$ is given by

$$\mathbb{E}[a_n^{(k)}(t)] = \sum_{h \in \mathcal{H}_n(t)} w_n^{(k)}(h,t) a_n^{(k)}(h,t) \tag{18}$$

The complexity for calculating $\mathbb{E}[a_n^{(k)}(t)]$ is $\mathcal{O}(W^2)$ and the evaluation of this value for different user can be parallelized.

Recall that only the first step is committed by the RHC-based policy, let $F_n^*(h_n(0)+1,1)$ and $F_n^*(1,1)$ respectively denote the value of cost-to-go function when $\boldsymbol{\lambda} = \boldsymbol{\lambda}^*$. $F_n(h_n(0),0)$ under a scheduling policy can be written as

$$F_n(h_n(0),0) = \begin{cases} F_n^*(h_n(0)+1,1) + p_n(0)h_n(0) \\ \qquad \text{if } a_n(0) = 0; \\ \lambda_0^* + q_n(F_n^*(1,1) + p_n(0)) \\ +(1-q_n)(F_n^*(h_n(0)+1,1)) \\ +(1-q_n)(p_n(0)(h_n(0)+1)) \\ \qquad \text{otherwise} \end{cases} \tag{19}$$

Our heuristic is a greedy policy with respect to $F_n(h_n(0), 0)$. That is, the action in $\{\{a_n(0)\} : \sum_{n=1}^{N} a_n(0) = K\}$ that result in minimum $\sum_{n=1}^{N} F_n(h_n(0), 0)$ will be taken, which yields an index policy with

$$I_n = q_n \left[F_n^*(h_n(0)+1, 1) + p_n(0)h_n(0) - F_n^*(1, 1)\right] - p_n(0) \tag{20}$$

With the heuristic above, the K users with the largest I_n will be selected. It's worth noting that the value of I_n also equals to the minimum λ_0 that makes actions $a_n(0) = 1$ and $a_n(0) = 0$ equally beneficial, which can be interpreted as the price that the server would be willing to pay to update the information of user n. The Lagrangian heuristic is summarized in Alg. 1.

Algorithm 1 RHC-based Heuristic

Input: $\mathbf{h}(0)$, $\{\mathbf{p}(0), \mathbf{p}(1), \ldots, \mathbf{p}(W-1)\}$
Output: scheduling decisions $\mathbf{a}(0)$
compute $\boldsymbol{\lambda}^*$ via subgradient update (14)
update the tabular cost-to-go functions $\{F_n(h,t)\}$ w.r.t. problem (15) with $\boldsymbol{\lambda}^*$
calculate the index of each user I_n according to (20)
set $a_n(0) = 1$ if user n is among the K users with the largest I_n (break ties arbitrarily); otherwise, set $a_n(0) = 0$

V. Numerical Results

In this section, simulations are conducted to evaluate the performance of the proactive scheduling policies in static scenario and dynamic scenario, respectively.

A. Static Scenario

In this scenario, we compare the performance in terms of time-average EAoI achieved by the request-aware Whittle's index policy with following two baseline solutions.

- Whittle's index policy in [19]: This policy leverages the knowledge of q_n and $h_n(t)$ in making scheduling decisions and can achieve near-optimal performance in terms of time-average AoI. To distinguish this policy with our request-aware policy, we call it request-oblivious policy in our numerical results.
- Myopic policy: This policy utilizes the knowledge of q_n, p_n, and $h_n(t)$ to make scheduling decisions. More precisely, under such policy, the update schedule that achieves the minimum immediate cost would be chosen by the server at each slot.

For each user's information, the probability of successful update q_n is randomly generated from a uniform distribution in the interval $[0.1, 1]$. To explore the impact of request pattern on the system performance, the request probability p_n is generated via three probabilistic models.

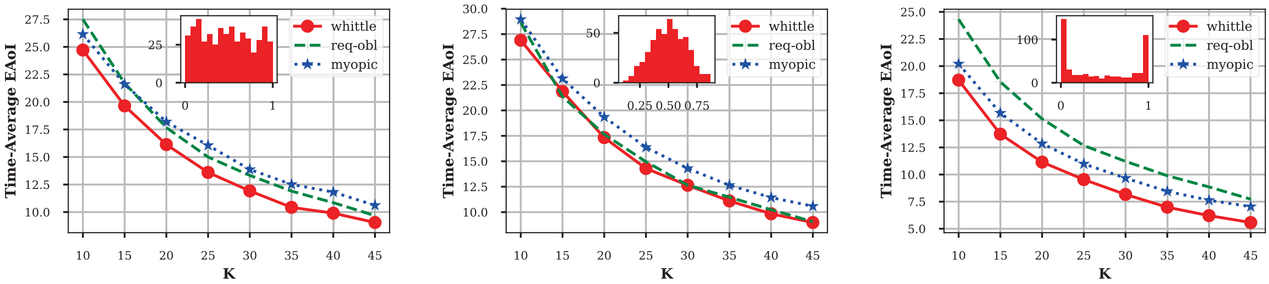
- Uniform distribution. In this case, p_n follows a uniform distribution in $[0.1, 1]$.
- Unimodal beta distribution. In this case, p_n of different users concentrate on a specific value, which in some sense represents the case of homogeneous user requests.
- Bimodal beta distribution. This model represents the case that there are two types of users. The first type of users request their information aggressively while the second type of users request their information infrequently.

Consider a system with $N = 500$, the time-average EAoI achieved by three scheduling policies with different values of K are illustrated in Fig. 2. The distributions of the values of p_n are also presented. As shown in Fig. 2, the Whittle's index policy always achieves better performance than the other two algorithms under all the three probabilistic models of p_n. Compared to the myopic policy, the Whittle's index policy can, on average, reduce the time-average EAoI by 13%, 11%, and 15% under those three probabilistic models of p_n, respectively. Consider that the bimodal beta distribution model captures the case in which two types of users with the highest disparity, this outcome indicates that the advantage of the Whittle's index policy over the myopic policy increases as the heterogeneity with respect to users' request patterns increases. Among those three scheduling policies, the request-oblivious algorithm suffers from such heterogeneity the most. In the case of uniform distribution model, the performance it can achieve is commensurate with that of the myopic policy while it provides slightly worse time-average EAoI compared to the Whittle's index policy under the unimodal beta distribution model. However, under the bimodal beta distribution model, its performance is far worse than that of the myopic policy.

Furthermore, we explore the performance of Whittle's index policy in systems with different scale. To this end, we run the scheduling algorithms with different numbers of users while the ratio $\kappa = \frac{N}{K}$ is fixed in each simulation. The corresponding simulation results are presented in Fig. 3. We can observe from Fig. 3 that, given the probabilistic model of p_n, the time-average EAoI achieved by the Whittle's index policy is mainly determined by κ. Specifically, the performance of the algorithm degrades as κ increases. On the other hand, with a fixed κ, the performance experiences slight fluctuation as the numbers of users increases from 200 to 550.

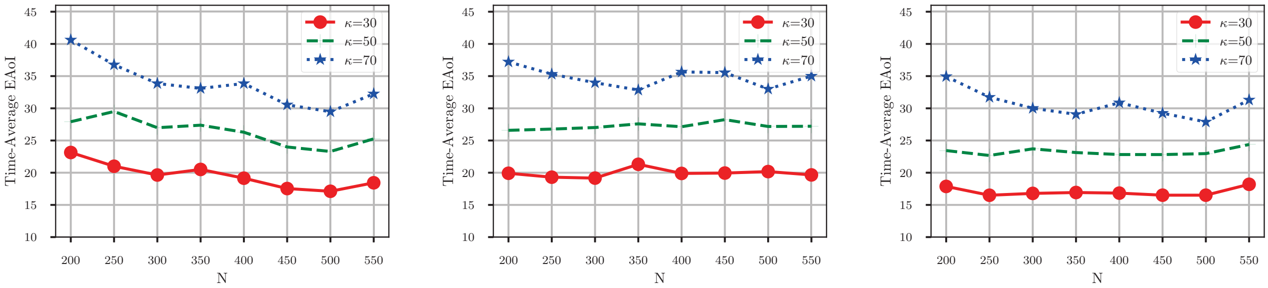
B. Dynamic Scenario

In this subsection, Markov model is used to characterize the time series forecasting model involved in the evaluations of the RHC-based policy. Roughly speaking, we assume that a user's request pattern satisfies the Markov property, given the request history in several previous slots. Specifically, in the simulations, an 8-state Markov model is generated for each user to capture the request pattern and a state is represented by the last three observations with respect to the arrivals of the requests. For example, let (111) denote the state of three consecutive requests observed. It will transition to either (111) or (110) based on the arrival of the request in the next slot, which is controlled by the transition probability. With the transition matrix of the Markov model, the estimates of the future arrivals can be readily calculated. The forecasting model updates these estimates at the beginning of every slot once the recent observation is obtained.



(a) uniform distribution (b) unimodal beta distribution (c) bimodal beta distribution

Fig. 2. Performance of different policies under three probabilistic models of p_n



(a) uniform distribution (b) unimodal beta distribution (c) bimodal beta distribution

Fig. 3. Performance of the Whittle's index policy in systems with different scale

We build a system with 100 users to evaluate the RHC-based policy. Recall that the WCDP problem is formulated as a finite-horizon surrogate of the original problem (3), a comprehensive empirical study is conducted to tune the hyperparameter β. For comparison, we also run the Whittle's index policy and the myopic policy in the dynamic scenario, where the estimates of the next step are considered as the request probability in the static scenario. Fig. 4 shows the traces of the time-average EAoI controlled by different policies. The length of the prediction window is set to 4 and the value of β used in the RHC-based policy is 0.023. As shown in Fig. 4, the RHC-based policy demonstrates better adaptability to the dynamic arrivals of the requests and achieves lower time-average EAoI.

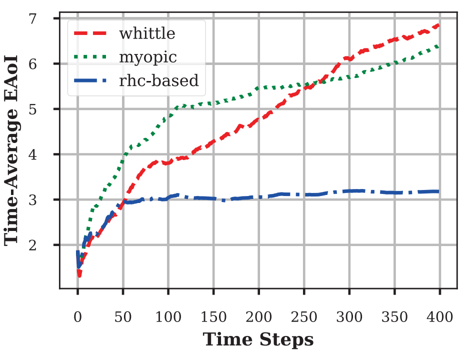


Fig. 4. Traces of time-average EAoI

The performances of these three policies under different values of K are illustrated in Fig. 5. Numerical results in Fig. 5 show that the RHC-based policy outperforms other two policies in reducing the time-average EAoI, achieving 12% average reduction compared to the Whittle's index policy.

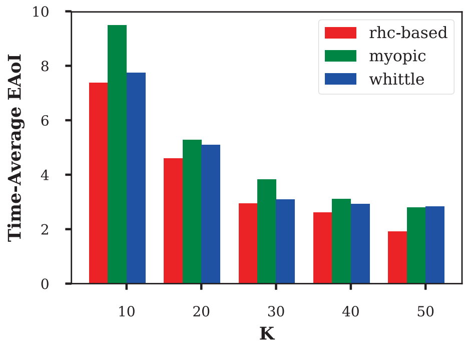


Fig. 5. Performance of the policies with different values of K

VI. Related Work

Among various research fields with respect to AoI minimization, the topic of link scheduling optimization is most related to our focus in this paper. In [15], the AoI minimization problem for scheduling a finite number of packets from multiple sources was proven to be NP-hard. Structural results of the optimal policy in special cases were also discussed. In [16], an age-based scheduler was proposed for applications that the status updates from different sources are synchronized. The work in [17] utilized MDP methods to address the scheduling problem with stochastic source updates in broadcast wireless networks while a round-robin policy with one-packet buffers policy (RR-ONE) was proven to achieve asymptotically optimal performance in terms of AoI in wireless uplink scheduling [18]. Consider periodic updates, the work in [19] proposed a series of scheduling policies for minimizing AoI in broadcast networks. Among them, the greedy policy was proven to be optimal for symmetric network settings while a Whittles index policy was shown to be sub-optimal in general cases. These results were further generalized in [20], where throughput constraint is considered. By modeling the scheduling problem with stochastic updates as an RMAB, Whittles index policies were respectively developed in [21] and [22]. The former focused on the system without buffer while the latter considered general buffering strategy. In addition, a decentralized index-prioritized random access policy was proposed in [22].

Different from the existing studies on AoI minimization, we introduce a novel concept EAoI in this paper, which characterizes the value of information freshness from the user's perspective. The work in [35] is the closest related work to the idea of EAoI minimization, which proposed a request replication scheme for a user to improve the EAoI of a single query. In contrast, this work focuses on the update management scheme of the server that aims to improve the time-average EAoI with respect to multiple users. Compared to the existing work, our model captures both proactive information update and timely information delivery. Based on that, our methods leverage users' request pattern in the design of scheduling policies, which is neglected by the existing solutions.

VII. Conclusion

In this paper, we have introduced the idea of EAoI and proposed a generic request-response model for studying the EAoI minimization problem. With the estimate of users' future requests, the server updates users' information proactively, attempting to deliver as fresh as possible information to users. For static request model, we have formulated the EAoI minimization problem as an RMAB and derived the Whittle's index policy. In addition, by casting the EAoI minimization problem as a WCDP, we have developed a computationally tractable RHC-based heuristic for the dynamic scenario. Extensive simulations have been conducted to demonstrate the advantages of the proposed approaches.

Appendix
Proof of Lemma 1

We assume that an optimal policy for the single-armed bandit is threshold-type with the threshold H. The server will initialize the update when $h \geq H$ while keeping idle when $1 \leq h < H$. Combine the Bellman equations associated with this problem and $f(1) = 0$, we obtain

$$\begin{aligned} f(h)+\mu = \min\{&f(h+1)+ph; \\ &C+qp+(1-q)(f(h+1)+p(h+1))\} \end{aligned} \tag{21}$$

In the case of $h \geq H$, the optimality condition of the Bellman equations requires that initializing the update incurs lower cost, i.e.,

$$f(h+1) \geq \frac{C+p-qph}{q}, \quad \forall h \geq H. \tag{22}$$

In addition, the Bellman equations in this case reduce to

$$f(h) = -\mu + C + p + (1-q)f(h+1) + (1-q)ph \tag{23}$$

With Eq. (23), $f(h), \forall h \geq H$ can be solved recursively with the condition $\lim_{m\to\infty}(1-q)^m f(h+m) = 0$, i.e.,

$$f(h) = \frac{-\mu + C + p}{q} + \frac{(1-q)^2 p}{q^2} + \frac{p(1-q)h}{q} \tag{24}$$

According to Eq. (24), $f(h)$ increases linearly with h when $h \geq H$. Therefore, the condition used in the recursion is valid. Furthermore, since the RHS of condition (22) decreases with the rise of h, to check the consistency of the results with (22), it is sufficient to show

$$f(H+1) \geq \frac{C+p-qpH}{q} \tag{25}$$

Substituting (24) into (25) yields

$$-\mu + pH \geq \frac{(2q-1)p}{q} - p \tag{26}$$

Likewise, in the case of $1 \leq h < H$, the solution of the Bellman equations are supposed to satisfy the following condition,

$$f(h+1) < \frac{C+p-qph}{q}, \quad \forall 1 \leq h < H. \tag{27}$$

And the Bellman equations in this case reduce to

$$f(h) = -\mu + f(h+1) + ph \tag{28}$$

For condition (27), consider the boundary case $h = H-1$, we obtain

$$f(H) < \frac{C+p-qp(H-1)}{q} \tag{29}$$

which further yields

$$-\mu + pH < \frac{(2q-1)p}{q} \tag{30}$$

If condition (30) holds, according to Eq. (28), we have

$$\begin{aligned} f(h+1) - f(h) &= \mu - ph \geq \mu - p(H-1) \\ &> \frac{(1-2q)p}{q} + p = \frac{(1-q)p}{q} \geq 0 \end{aligned} \tag{31}$$

Thus, $f(h+1)$ is monotonically increasing with h when $1 \leq h < H$ if condition (30) holds. Note that the RHS of condition (27) is monotonically decreasing with h, condition (30) is the sufficient condition of condition (27).

Given conditions (26) and (30), we introduce an auxiliary parameter $\gamma \in (0, 1]$ such that

$$-\mu + pH = \frac{(2q-1)p}{q} - p\gamma \tag{32}$$

Solving Eq. (28) recursively yields

$$f(H-m)=f(H)-m\mu+\frac{mp(2H-m-1)}{2} \tag{33}$$

Let $m=H-1$, we obtain

$$f(H)-(H-1)\left[\mu-pH+\frac{pH}{2}\right]=0 \tag{34}$$

Substituting $f(H)$ from Eq. (24) into Eq. (34) and using Eq. (32) to eliminate μ, we have

$$\frac{p}{2}H^2-(\frac{3}{2}-\gamma-\frac{1}{q})pH+\frac{p\gamma-2p-C}{q}+p-p\gamma=0 \tag{35}$$

which gives the unique positive root

$$H(\gamma)=\frac{3}{2}-\gamma-\frac{1}{q}+\sqrt{\gamma(\gamma-1)+(\frac{1}{q}+\frac{1}{2})^2+\frac{2C}{pq}} \tag{36}$$

The threshold H is monotonically decreasing over $\gamma\in(0,1]$ since its first derivative is negative. Moreover, it can be easily checked that

$$\lim_{\gamma\to 0}H(\gamma)=H(1)+1 \tag{37}$$

which implies the existence of a unique threshold H^*, within the range $[H(1),H(1)+1)$, whose value is an integer. Specifically, such threshold can be expressed as

$$H^*=\lceil H(1)\rceil=\left\lceil\frac{1}{2}-\frac{1}{q}+\sqrt{(\frac{1}{q}+\frac{1}{2})^2+\frac{2C}{pq}}\right\rceil \tag{38}$$

Since the γ^* that is associated with H^* falls in the interval $(0,1]$, conditions (26) and (30) hold. The threshold-type scheduling policy is consistent with the Bellman equations, which implies its optimality.

ACKNOWLEDGMENT

This work was supported in part by the NSF of USA under Grant CNS-1816908, ECCS-1610874, ECCS-1554576, and the National Natural Science Foundation of China under Grants 61628107, 91638204, 61861136003, 61871254, 61571265, 61621091.

REFERENCES

[1] A. Kosta, N. Pappas, V. Angelakis *et al.*, "Age of information: A new concept, metric, and tool," *Foundations and Trends® in Networking*, vol. 12, no. 3, pp. 162–259, 2017.
[2] S. Kaul, M. Gruteser, V. Rai, and J. Kenney, "Minimizing age of information in vehicular networks," in *Proc. of IEEE SECON*, 2011, pp. 350–358.
[3] S. Kaul, R. Yates, and M. Gruteser, "Real-time status: How often should one update?" in *Proc. of IEEE INFOCOM*, 2012, pp. 2731–2735.
[4] R. D. Yates and S. Kaul, "Real-time status updating: Multiple sources," in *Proc. of IEEE ISIT*, 2012, pp. 2666–2670.
[5] L. Huang and E. Modiano, "Optimizing age-of-information in a multi-class queueing system," in *Proc. of IEEE ISIT*, 2015, pp. 1681–1685.
[6] R. D. Yates, "Lazy is timely: Status updates by an energy harvesting source," in *Proc. of IEEE ISIT*, 2015, pp. 3008–3012.
[7] Y. Sun, E. Uysal-Biyikoglu, R. Yates, C. E. Koksal, and N. B. Shroff, "Update or wait: How to keep your data fresh," in *Proc. of IEEE INFOCOM*, 2016, pp. 1–9.
[8] M. Costa, M. Codreanu, and A. Ephremides, "Age of information with packet management," in *Proc. of IEEE ISIT*, 2014, pp. 1583–1587.
[9] B. Li, A. Eryilmaz, and R. Srikant, "On the universality of age-based scheduling in wireless networks," in *Proc. of IEEE INFOCOM*, 2015, pp. 1302–1310.
[10] R. D. Yates and S. K. Kaul, "The age of information: Real-time status updating by multiple sources," *arXiv preprint arXiv:1608.08622*, 2016.
[11] A. M. Bedewy, Y. Sun, and N. B. Shroff, "Optimizing data freshness, throughput, and delay in multi-server information-update systems," in *Proc. of IEEE ISIT*, 2016, pp. 2569–2573.
[12] ——, "Age-optimal information updates in multihop networks," in *Proc. of IEEE ISIT*, 2017, pp. 576–580.
[13] R. D. Yates, M. Tavan, Y. Hu, and D. Raychaudhuri, "Timely cloud gaming," in *Proc. of IEEE INFOCOM*, 2017, pp. 1–9.
[14] Y. Sun, E. Uysal-Biyikoglu, and S. Kompella, "Age-optimal updates of multiple information flows," *arXiv preprint arXiv:1801.02394*, 2018.
[15] Q. He, D. Yuan, and A. Ephremides, "Optimal link scheduling for age minimization in wireless systems," *IEEE Transactions on Information Theory*, 2017.
[16] C. Joo and A. Eryilmaz, "Wireless scheduling for information freshness and synchrony: Drift-based design and heavy-traffic analysis," in *Proc. of IEEE WiOpt*, 2017, pp. 1–8.
[17] Y.-P. Hsu, E. Modiano, and L. Duan, "Age of information: Design and analysis of optimal scheduling algorithms," in *Proc. of IEEE ISIT*, 2017, pp. 561–565.
[18] Z. Jiang, B. Krishnamachari, X. Zheng, S. Zhou, and Z. Niu, "Timely status update in massive iot systems: Decentralized scheduling for wireless uplinks," *arXiv preprint arXiv:1801.03975*, 2018.
[19] I. Kadota, A. Sinha, E. Uysal-Biyikoglu, R. Singh, and E. Modiano, "Scheduling policies for minimizing age of information in broadcast wireless networks," *arXiv preprint arXiv:1801.01803*, 2018.
[20] I. Kadota, A. Sinha, and E. Modiano, "Optimizing age of information in wireless networks with throughput constraints," in *Proc. of IEEE INFOCOM*, 2018.
[21] Y.-P. Hsu, "Age of information: Whittle index for scheduling stochastic arrivals," *arXiv preprint arXiv:1801.03422*, 2018.
[22] Z. Jiang, B. Krishnamachari, S. Zhou, and Z. Niu, "Can decentralized status update achieve universally near-optimal age-of-information in wireless multiaccess channels?" *arXiv preprint arXiv:1803.08189*, 2018.
[23] K. Chen and L. Huang, "Timely-throughput optimal scheduling with prediction," in *Proc. of IEEE INFOCOM*, 2018.
[24] B. Yin, Y. Cheng, L. X. Cai, and X. Cao, "Online SLA-aware multi-resource allocation for deadline sensitive jobs in edge-clouds," in *Proc. of IEEE GLOBECOM*, 2017, pp. 1–6.
[25] L. Liu, B. Yin, S. Zhang, X. Cao, and Y. Cheng, "Deep learning meets wireless network optimization: Identify critical links," accepted to *IEEE Transactions on Network Science and Engineering*, to appear.
[26] X. Cao, R. Ma, L. Liu, H. Shi, Y. Cheng, and C. Sun, "A machine learning based algorithm for joint scheduling and power control in wireless networks," accepted to *IEEE Internet of Things Journal*, to appear.
[27] P. Whittle, "Restless bandits: Activity allocation in a changing world," *Journal of applied probability*, vol. 25, no. A, pp. 287–298, 1988.
[28] R. R. Weber and G. Weiss, "On an index policy for restless bandits," *Journal of Applied Probability*, vol. 27, no. 3, pp. 637–648, 1990.
[29] 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.
[30] Z. Yu, Y. Xu, and L. Tong, "Deadline scheduling as restless bandits," *IEEE Transactions on Automatic Control*, 2018.
[31] 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.
[32] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.
[33] J. T. Hawkins, "A langrangian decomposition approach to weakly coupled dynamic optimization problems and its applications," Ph.D. dissertation, Massachusetts Institute of Technology, 2003.
[34] M. L. Puterman, *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.
[35] Y. Sang, B. Li, and B. Ji, "The power of waiting for more than one response in minimizing the age-of-information," in *Proc. of IEEE GLOBECOM*, 2017, pp. 1–6.