Timely Status Update Based on Urgency of Information with Statistical Context

Lehan Wang, Jingzhou Sun, Sheng Zhou, Zhisheng Niu
Beijing National Research Center for Information Science and Technology
Department of Electronic Engineering, Tsinghua University, Beijing 100084, P.R. China
{wang-lh19,sunjz18}@mails.tsinghua.edu.cn, {sheng.zhou, niuzhs}@tsinghua.edu.cn

Abstract—Real-time status update has recently brought the widespread attentions, especially in the field of remote control. Accordingly, a metric named Urgency of Information (UoI) has been proposed to capture the timeliness of information and the evolution of the context at the same time. It is defined as the product of the context-aware weight and the cost function of the estimation error. However, the metric itself does not indicate how to determine the contextual weight. Based on UoI, this paper proposes a dynamic threshold-based scheduling policy which only requires the conditional expectation of the weight in the next time slot by Lyapunov method. When the environment is unknown, We also use the online algorithm SARSA to obtain a policy. The simulation results show that both policies are near-optimal when the update resources are relatively rich, and the SARSA-based policy works better under the tight resources constraint over the Lyapunov method.

I. INTRODUCTION

In the 5G era, real-time status update serves as one key application scenario, especially with the ultra-reliable and low-latency communications (uRLLC). Typical remote-control applications like tele-surgery [1], vehicular network [2] and so on, not only require low transmission delay but also the accuracy and the freshness of status information collected from the devices and sensors, so as to maintain the stability and the safety of the systems.

To measure the freshness of status information, Age of Information (AoI) was proposed in [3] as the time period between the current time stamp and the generation of the latest received packets. Recently, there have been many researches on AoI: in [4] the authors optimized the average peak AoI in a wireless network with multiple sensors and a single access point under the energy constraints. In [5], the authors managed to minimize the average AoI with standard automatic repeat request (ARQ) and hybrid ARQ protocols under resources constraint. Tail distribution of AoI was studied in [6] and the authors argued that optimizing the average AoI metric may not ensure the requirements of low delay and high reliability of certain applications, such as vehicle-to-vehicle (V2V) communications. In [7], the stationary distribution of AoI was analyzed in a single sensor network using queuing theory under different service disciplines. In [8], the authors derived the delay and the peak AoI violation probability by characterizing the probability-generating functions (PGFs) of delay and peak AoI at the steady state. The difference between optimizing the average AoI and the violation-probability has also been discussed in [8] by simulation. In [9]–[10], age penalty function was studied to measure the non-linear performance degradation caused by data staleness. Another line of work focuses on AoI-like metrics. For example, in [11], a metric named Age of Synchronization (AoS) was defined as the time elapsed since the local cache became desynchronized in cache systems. The authors argued that the AoI-optimal policy depends only on the sources popularity while the AoS near-optimal policy is related to both the update rate and the sources popularity.

On the other hand, the information timeliness should also take the context information of the monitored process into account. The context means all the environmental information that influences the significance of the information freshness of the whole system. Therefore, when a system is in an emergency, the status should be updated frequently to maintain the information freshness in this case. However, study on the influence of the context information is in its inception. In [12]–[13], a new metric named Urgency of Information (UoI) was proposed to capture the context dynamic and the information freshness at the same time. The authors proposed the context-aware weights $\omega_t$ to characterize the influence of the environment in the identical way. The UoI is defined as a product of the context-aware weights and the cost function of the estimation error $Q(t)$ of the current status, namely:

$$F(t) = \omega_t \delta(Q_t),$$

where the context-aware weights are modeled as a random variable independent over all the time slots in [13]. A large weight means that the system is in emergency and thus demands higher accuracy of status information, and vice versa.

For example, when a vehicle is approaching an intersection, more accurate information is needed to guarantee safety in this complicated crossroad environment, and thus the value of $\omega_t$ increases.

In this work, we adopt the similar remote control and tracking model in [13]. In this model, the objective is to minimize the average UoI of a single user over time under the resources constraint. We also assume that the exact value of the context-aware weight of the next time slot is not available. Therefore, the scheduling policy for the case where the context-weight in the next time slot is unknown is studied. We used Lyapunov method [14] to obtain the
dynamic threshold-based scheduling policy, which only needs
the conditional expectation of the weight in the next time slot.
When the conditional expectation is also unknown, we further
use SARSA [15] to obtain a policy. Both the proposed policy
was compared with the optimal policy derived by Constrained
Markov Decision Process (CMDP) [16] and show a near-
optimal performance when the update resources are relatively
rich in simulation.

The rest of this paper is organized as follows. In Section II,
the system model and the problem formulation are described.
The proposed dynamic threshold-based scheduling policy are
presented in Section III and the proposed SARSA-based
scheduling policy is obtained in Section IV. Finally, in Section
V, the simulation results are shown and analyzed while the
conclusions are drawn in Section VI.

II. SYSTEM MODEL AND PROBLEM FORMULATION

A. Remote Control and Tracking Model

We consider a vehicle remote control system, including a
single vehicle and a roadside controller, shown in Fig. 1. The
whole system is assumed as a discrete time linear control sys-
tem. At the beginning of each time slot $t$, the vehicle updates
its current status information $x_t$ (e.g. the current location of
the vehicle) to the controller. The controller then adjusts the
velocity of the vehicle in next time slot, i.e. $v_{t+1}$, to make
the vehicle towards a target status $(y_{t+1}$, which will only be
known in slot $t$) according to the received information $x_t$.
Then the status of the vehicle at time slot $t + 1$ should be:

$$x_{t+1} = ax_t + bv_{t+1} + n_{t+1},$$  (1)

where $n_{t+1}$ is the noise at time slot $t + 1$. Therefore the status
information must be updated when $(x_t - y_t)^2$ grows.

Based on the above analysis, the objective of the problem
can be formulated as:

$$\min \lim_{T \to \infty} \frac{1}{T} \sum_{t=0}^{T-1} E[\omega_t(x_t - y_t)^2],$$  (2)

where $\omega_t$ is the context-aware weight. However, in the prac-
tical system, the current status information $x_t$ is not always
available, because the vehicle may not be arranged to update
its information or the update fails due to the channel fading
in the time slot $t$. In this case, the roadside controller should
first estimate the status of the vehicle $x_t$ by the historical status
information. We define the estimation of $x_t$ as $\hat{x}_t$. Therefore
the original objective in (2) can be transformed into (3) [13]:

$$\min \lim_{T \to \infty} \frac{1}{T} \sum_{t=0}^{T-1} E[\omega_t(Q_t)^2],$$  (3)

where $Q_t = x_t - \hat{x}_t$ is the estimation error of the vehicle in
time slot $t$. Therefore, our objective is to minimize the average
UoI over time actually. Similar to [12], we further assume the
same period of a packet transmission is less than a time slot
and the estimation error only depends on the newest update
delivered by the vehicle. Then the recurrence relation of $Q_t$
will be:

$$Q_{t+1} = (1 - D_t)Q_t + A_t,$$  (4)

where $D_t$ is a random variable which takes value from \{0, 1\},
when $D_t = 1$, the vehicle successfully updates its status
information in time slot $t$. $A_t$ is the increment of the estimation
error at time slot $t$ and it obeys a Gaussian distribution with the
mean value of 0 and the variance of $\sigma^2$, i.e. $A_t \sim N(0, \sigma^2)$.

Based on the assumptions above, we can obtain that even if
the system status information is not updated for a long time,
the value of $Q_t$ may still be kept at a low level. It indicates that
the outdated information may still be useful to some extent
in some special cases. For example, if we assume that the
status information has Markovian property, then the outdated
information may help us to predict the current status. However,
the definition of AoI makes it grow linearly with time and can
not capture the nonlinear inaccuracy of the estimation.

B. Wireless Channel Model

If the vehicle successfully updates its status information
in time slot $t$, then the vehicle must be scheduled to update in
time slot $t$ and the status update packets should be successfully
transmitted to the controller. According to [12], the random
variable $D_t$ should be the product of $S_t$ and $U_t$, i.e. $D_t = S_tU_t$,
where $U_t = 1$ means the vehicle is scheduled to update
status information in time slot $t$.

We assume the uplink channel from the vehicle to the
roadside controller is a block fading channel. Therefore the
status update packets will be successfully transmitted with
probability $p$ [12]. Thus $S_t \in \{0, 1\}$ is defined to represent
whether the transmission is successful in time slot $t$. 

![Fig. 1. Remote Control and Tracking Model.](image-url)

Meanwhile, the scheduling policy of information updates is
also related to the situation and environment of the system. For
example, in this remote control system, when the vehicle is
in emergency, such as passing the intersection, overtaking, or
when the vehicle ahead stops in an emergency, the significance
of the accuracy of status information will be very high.

Also, when the car is in the ordinary situations, then the
importance of the status information will be relatively low.
Therefore, if the resources for the status updates are very
scarce in the system, then the policy should be designed to
save the limited resources for the emergency state to ensure
the safety of the whole system.
C. Update Frequency Constraint of the Model

Obviously, a higher status update frequency may improve the freshness of information, but will bring burden to both the transmitter and the receiver. Meanwhile, the limited wireless channel resources may not allow the unlimited increase of the update frequency. Thus the resource constraint of the system will be:

$$\lim_{T \to \infty} \frac{1}{T} \sum_{t=0}^{T-1} E[U_t] \leq \rho,$$

(5)

where $\rho$ is the maximum average update frequency and $\rho \in (0,1]$.

D. Problem Formulation

Based on the analysis above, the problem can be formulated as follow:

$$\min_{U_t} \lim_{T \to \infty} \frac{1}{T} E[\sum_{t=0}^{T-1} U_t Q_t^2]$$

s.t. $$\lim_{T \to \infty} \frac{1}{T} \sum_{t=0}^{T-1} E[U_t] \leq \rho.$$

(6)

In fact, our objective is to minimize average UoI over time under the constraint on the update resource. The cost function of the estimation error here is $\delta(Q_t) = Q_t^2$.

III. THE DYNAMIC THRESHOLD-BASED SCHEDULING POLICY

A. Lyapunov Method

Similar to [12], Lyapunov methods can be used to solve the optimization problem in (6). First of all, we define a virtual queue $H_t$ which can be used to satisfy the update frequency constraint [14]:

$$H_{t+1} = [H_t - \rho + U_t]^+.$$

(7)

With the virtual queue, we adopt the min drift-plus-penalty optimization to solve the problem. To proceed, the Lyapunov function is defined as:

$$L_t = \frac{1}{2} V H_t^2 + \theta Q_t^2,$$

(8)

where $\theta$ and $V$ are positive constant, i.e. $\theta > 0, V > 0$.

Furthermore, the penalty of the system at time slot $t$ is defined as UoI in next slot $t+1$:

$$f_t = \omega_{t+1} Q_{t+1}^2.$$

(9)

Thus the Lyapunov drift-plus-penalty is $E[L_{t+1} - L_t + f_t]$. According to the min drift-plus-penalty algorithm, if the drift-plus-penalty is bounded by a constant $C$, then the resources constraint will be satisfied while the objective in (6), i.e. $\lim_{T \to \infty} \frac{1}{T} E[\sum_{t=0}^{T-1} U_t Q_t^2]$, will also be bounded by the constant $C$. Therefore, we can solve the optimization problem in (6) by finding and minimizing the upper bound of $E[L_{t+1} - L_t + f_t]$.

Meanwhile, we aim to make decisions based on the weight in the current time slot. Therefore, $H_t, Q_t, \omega_t$ are chosen as the condition of the drift-plus-penalty.

Then with the recurrence relation of $Q_t$ (4) we can derive the conditional expectation of the penalty $f_t$:

$$E[f_t(Q_t, \omega_t, H_t)] = E[\omega_{t+1} Q_{t+1}^2 | Q_t, \omega_t, H_t]$$

$$= E[\omega_{t+1} \{(1 - U_t S_t) Q_t + A_t^2\} | Q_t, \omega_t, H_t]$$

$$= Q_t^2 E[\omega_{t+1} | Q_t, \omega_t, H_t, Q_t] - p Q_t^2 E(\omega_{t+1} U_t | \omega_t, H_t, Q_t)$$

$$+ \sigma^2 E(\omega_{t+1} \omega_t | Q_t, \omega_t, H_t).$$

(10)

In addition, the sum of the conditional expectation of the Lyapunov drift-plus-penalty can be bounded by:

$$E[L_{t+1} - L_t + f_t(Q_t, \omega_t, H_t)] \leq \theta \sigma^2 + \frac{1}{2} V - V \rho H_t$$

$$+ E[f_t(Q_t, \omega_{t+1}, H_t)] + (V H_t - \theta p Q_t^2) E[U_t | Q_t, \omega_{t+1}, H_t].$$

(11)

By (10), we can further obtain:

$$E[L_{t+1} - L_t + f_t(Q_t, \omega_t, H_t)] \leq \theta \sigma^2 + \frac{1}{2} V - V \rho H_t$$

$$+ Q_t^2 E(\omega_{t+1} | \omega_t, H_t, Q_t) - p Q_t^2 E(\omega_{t+1} U_t | \omega_t, H_t, Q_t)$$

$$+ \sigma^2 E(\omega_{t+1} | \omega_t, H_t, Q_t) + (V H_t - \theta p Q_t^2) E[U_t | Q_t, \omega_t, H_t].$$

(12)

Obviously, if we assume that the transition probability from $\omega_t$ to $\omega_{t+1}$ is independent with $U_t$ and $E[\omega_{t+1} | Q_t, \omega_t, H_t] = E[\omega_{t+1} | \omega_t]$, then we have:

$$E(\omega_{t+1} U_t | \omega_t, H_t, Q_t) = E(\omega_{t+1} | \omega_t, H_t, Q_t) E(U_t | \omega_t, H_t, Q_t).$$

Therefore the drift-plus-penalty will be bounded:

$$E[L_{t+1} - L_t + f_t(Q_t, \omega_t, H_t)]$$

$$\leq \theta \sigma^2 + \frac{1}{2} V - V \rho H_t + Q_t^2 E(\omega_{t+1} | \omega_t)$$

$$- p Q_t^2 E(\omega_{t+1} | \omega_t) E[U_t | Q_t, \omega_t, H_t] + \sigma^2 E(\omega_{t+1} | \omega_t)$$

$$+ (V H_t - \theta p Q_t^2) E[U_t | Q_t, \omega_t, H_t].$$

(13)

With the results above, we can get the scheduling policy by minimizing the upper bound of drift-plus-penalty in (13):

$$\min_{U_t} \{V H_t - p(\theta + E[\omega_{t+1} | \omega_t]) Q_t^2 U_t \}$$

s.t. $E[U_t] \leq \rho.$

(14)

To ensure that the drift plus penalty is bounded in any instances, the value of $\theta$ should be carefully chosen. To this end, a randomized policy is used. In this policy, the vehicle updates its status in any time slot with the probability of $\rho$, the maximum average update frequency. Obviously, the randomized policy can also satisfy the resource constraint. In this randomized policy, $E[U_t | Q_t, \omega_t, H_t]$ will be the constant $\rho$, then :

$$E[L_{t+1} - L_t + f_t(Q_t, \omega_t, H_t)] \leq (\theta + E(\omega_{t+1} | \omega_t)) \sigma^2 + \frac{1}{2} V$$

$$+ Q_t^2 (1 - p \rho) E(\omega_{t+1} | \omega_t) - \theta p \rho).$$

(15)
Take the expectation over the weights $\omega_t$ and the estimation error $Q_t$ on both sides of (15):

$$E[L_{t+1} - L_t + f_t | H_t] \leq (\theta + E[\omega_{t+1}])\sigma^2 + \frac{1}{2} V + ((1 - pp)E[\omega_{t+1}]E(Q_t^2) - \theta pp E(Q_t^2)).$$

(16)

Since the first two terms in the right hand side of (16) are already constants, therefore we only need to focus on the last term by choosing a proper $\theta$. Thus we can choose $\theta$ as follow to set the last term in (16) as zero:

$$\theta = \frac{1 - pp}{pp} E(\omega_{t+1}).$$

(17)

Therefore, the drift-plus-penalty is bounded by:

$$E[L_{t+1} - L_t + f_t] \leq \frac{1}{pp} E(\omega_{t+1})\sigma^2 + \frac{1}{2} V.$$ (18)

Then the sum of expected drift-plus-penalty from time slot 0 to time slot $T - 1$ will be:

$$\sum_{t=0}^{T-1} E[L_{t+1} - L_t + f_t] = E[L_T] - E[L_0] + \sum_{t=0}^{T-1} E[f_t] \leq (\frac{1}{pp} E(\omega_{t+1})\sigma^2 + \frac{1}{2} V)T.$$ (19)

With the fact that $E[L_t] = E[\frac{1}{2} V H_t^2 + \theta Q_t^2] \geq 0$, we can divide the constant $T$ at both sides of (19):

$$\frac{1}{T}(E[L_T] - E[L_0] + \sum_{t=0}^{T-1} E[f_t]) \leq \frac{1}{pp} E(\omega_{t+1})\sigma^2 + \frac{1}{2} V.$$ (20)

Furthermore, when $T \to \infty$:

$$\limsup_{T \to \infty} \frac{1}{T} E[\sum_{t=0}^{T-1} \omega_t Q_t^2] \leq \frac{1}{pp} E(\omega_{t+1})\sigma^2 + \frac{1}{2} V.$$ (21)

B. Context-Aware Timely Status Update

With the chosen $\theta$ in (17), we propose the dynamic threshold-based scheduling policy, shown in Algorithm 1.

In the policy, the inputs are: the probability of successful transmission $\rho$, the maximum average update frequency $\rho$, the coefficient of Lyapunov function $V$, the initial length of virtual queue $H_0$, the state transferring probability matrix of the context-aware weights $P$, and the variance of $A_t$: $\sigma^2$.

At the beginning of each time slot, we only have to calculate $J_t = (\theta + E[\omega_{t+1} | \omega_t])pQ_t^2$ and compare $J_t$ with $V H_t$, if $J_t > V H_t$, then the vehicle should be scheduled to update.

IV. SCHEDULING IN AN UNKNOWN CONTEXT

Although the timely status update scheduling presented in Sections III is simple, the policy still needs the conditional expectation of the context-aware weights $E(\omega_{t+1} | \omega_t)$ to make decisions. Therefore the policy requires the completed transition probability matrix of the context-aware weights in advance. However, in most practical systems, the transition probability matrix of the weight may not be known in advance.

To this end, we will assume the vehicle does not have the information of the transition probabilities of $\omega_t$ in advance and have to learn them in this section. We use the online reinforcement learning algorithm Sarsa to learn the transition probability of the weight over time.

Therefore, in our formulation, Sarsa algorithm, defined by the 5-tuple $(S, A, r, S', A')$: The set of states is $(\omega_t, Q_t) \in S$. Besides, to make the states space countable, we have to first do the discretization of the estimation error $Q_t$. The finite action set $A = \{U_t = 0, U_t = 1\}$ represents the only two optional actions in each time slot are update or not. For the ‘reward’ $r$, we define the reward in time slot $t$ as: $r_t = \omega_{t+1} Q_{t+1}^2 + \eta U_t$, where $\eta \geq 0$ is the Lagrange multiplier. However, we do not want to maximize $r$, but minimize this reward $r$ according to the minimization objective in (6). Finally, $S, A$ and $S', A'$ represents the state-action pair in the current slot and next slot, respectively.

To further explain the proposed Sarsa-based scheduling policy, we show the detailed policy in Algorithm 2. For example, in time slot $t$, the learning algorithm will first select and perform the action $a_t \in A$ according to the observation of the current state $s_t \in S$ with a policy derived by Q-table. This Q-table stores $Q_{Q_t}(s, a)$, the estimate of action-value function in respect of different state-action pair.

To balance exploitation and exploration, the policy used in our Sarsa is the $\epsilon$-greedy policy, namely the policy will exploit the trained Q-table and select the action $a_t$ with minimum $Q_t(s_t, a_t)$ with probability $\epsilon$ and randomly select an action with probability $1 - \epsilon$ to explore other more valuable actions, namely:

$$a_t = \begin{cases} \operatorname{arg\ min} Q_t(s_t, a_t), & \text{with probability } \epsilon; \\ a_t \text{ randomly select } a_t \in A, & \text{with probability } 1 - \epsilon. \end{cases}$$

(22)

Therefore, in our formulation, the Q-table is a matrix with $|S|$ rows and $|A|$ columns. $|S|$ is the size of the state space of the vehicle while $|A|$ represents the two optional actions: update or not. Further, $|S|$ should be the product of $|S_n|$, the size of state space of the context-aware weights $\omega_t$.

Algorithm 1 Dynamic threshold-based scheduling Policy

1. **Input:** $\rho, V, H_0, P, \sigma^2$
2. $t \leftarrow 0$
3. $H_t \leftarrow H_0$
4. **repeat**
5. $J_t = (\theta + E[\omega_{t+1} | \omega_t])pQ_t^2$
6. $U_t \leftarrow 0$
7. **if** $J_t > V H_t$, then:
8. $U_t \leftarrow 1$
9. **end if**
10. $H_t = (H_{t-1} - \rho + U_t)^+$
11. $t \leftarrow t + 1$
12. **until** end

<table>
<thead>
<tr>
<th>$A$</th>
<th>$R$</th>
<th>$S'$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$U$</td>
<td>$V$</td>
<td>$U$</td>
</tr>
</tbody>
</table>

93
Algorithm 2 SARS-based Scheduling Policy with a given $\eta$

1. Input: $p, \eta, \rho, \sigma^2, \alpha_t, \gamma, \epsilon$
2. $t \leftarrow 0$
3. $Q_{t}^{S,S_{1}\times 2} \leftarrow 0$
4. repeat
5. observe the current state $s_t$
6. select action $a_t$ under epsilon-greedy policy
7. perform action $a_t$ and observe the reward $r_t$ and next state $s_{t+1}$
8. select action $a_{t+1}$ under epsilon-greedy policy
9. Update the Q-table with the 5-tuple of $(s_{t},a_{t},r_{t},s_{t+1},a_{t+1})$ by (23)
10. $s_t \leftarrow s_{t+1}$
11. $a_t \leftarrow a_{t+1}$
12. $t \leftarrow t + 1$
13. until end

and $|S_{Q_t}|$, the size of state space of the estimation error $Q_t$. At the beginning of the algorithm, the Q-table is initialized to an all zero matrix.

Secondly, observe the next state $s_{t+1}$ and the reward $r_t$. Then, the algorithm will select and perform the action $a_{t+1}$ based on the next state $s_{t+1}$. Based on the 5-tuple of $(s_{t},a_{t},r_{t},s_{t+1},a_{t+1})$, the Q-table will be updated by:

$$Q_{t}(s_t,a_t) \leftarrow Q_{t}(s_t,a_t) + \alpha_t[\omega_{t+1}Q_{t+1} + \eta U_t$$

$$+ \gamma Q_{t}(s_{t+1},a_{t+1}) - Q_{t}(s_t,a_t)],$$

(23)

where $\alpha_t$ is the learning rate for time slot $t$ and $\gamma$ is the discount-rate parameter.

In this policy, the inputs are: the probability of successful transmission $p$, the Lagrange coefficient $\eta$, the maximum average update frequency $\rho$, the variance of $A_t$: $\sigma^2$, the probability of exploitation $\epsilon$, the learning rate and the discount-rate parameter.

In this paper, with a given update frequency constraint $\rho$, we will first choose a proper Lagrange multiplier $\eta$ which will reach an average update frequency close to and less than $\rho$. With the chosen $\eta$, we will run Algorithm 2 to obtain the SARSA-based policy. Intuitively, the smaller $\eta$, the closer $\rho$ to 1.

V. SIMULATION RESULTS

A. Simulation Setup

In order to facilitate the simulation and conform to the assumptions about $\omega_t$ used in Section III.A, we assume that the context-aware weight $\omega_t$ has the first-order Markov property. Also, we only consider the simplest situation where the context-aware weights of the vehicle have two different states: one 'low' state and one 'high' state. The state transition diagram of $\omega_t$ is shown in Fig. 2.

When $\omega_t$ is in the 'low' state, then we set $\omega_t$ as 1 to represent that the vehicle is in an ordinary state while the significance of the accuracy of the status information is relatively low. When $\omega_t$ is in the 'high' state, then $\omega_t$ is set as a much larger than 1 constant $\omega_c$, and the vehicle is in emergency with a relatively high significance of the accuracy of the status information. Also, $p_1$ is the probability of the transition from ordinary state to emergency, while $p_2$ is the probability of the transition from emergency to ordinary states.

The benchmark algorithm is the optimal policy obtained by CMDP. In CMDP, the state space and the optional actions of the vehicle are the same as SARSA. However, CMDP knows the completed transition matrix of all the states in advance which makes it possible for CMDP to get the UoI-optimal policy.

B. Numerical Results

Fig. 3 and Fig. 4 show the comparison of the average UoI and AoI of the policy derived by CMDP, the dynamic threshold-based scheduling policy and the SARSA-based scheduling policy when $p_1 = 0.001$, $p_2 = 0.01$, $\rho = 0.9$, $\sigma^2 = 1$, $\omega_c = 100$, respectively. In the dynamic threshold-based scheduling policy, $V$ is set as 1, while in the SARSA-based scheduling policy, the learning rate $\alpha_t = 0.1$ and the discount-rate is 0.5.

In Fig. 3, CMDP performs best while the other two only achieve near-optimal when the update resources are relatively rich, namely when $\rho$ is in a high level. SARSA-based scheduling policy performs better than the dynamic threshold-based scheduling policy when the resources are relatively scarce. This is caused by the defects of the virtual queue used in the policy, a smaller $\rho$ will result in a faster growth of the
virtual queue when $\omega_k$ is in the 'high' state. This phenomenon will cause a severe outage of updates when the weight suddenly drops to a low level. However, in Fig. 4, the dynamic threshold-based scheduling policy performs best, which shows the significant difference between the results of optimizing age and UoI.

Fig. 5 shows the comparison among the convergence rate of CMDP and the SARSA based scheduling policy with different $\epsilon$. It shows that the convergence rate of CMDP is higher than the other three. Also, when $\epsilon$ is 0.99, our SARSA preforms better than the policy with $\epsilon = 0.95$ or 0.9.

Fig. 6 shows the UoI curves of the dynamic threshold-based scheduling policy and a smaller $V$ will result in a lower update threshold and a lower average UoI. However, the smallest $V$ is not always best. Fig. 7 shows the convergence of the actual average update frequency of the dynamic threshold-based policy over 5 millions slots. We can conclude that a smaller $V$ will also result in a lower convergence rate and ruin the fairness of scheduling over time before a convergence.

The final question may be how to choose a proper $\omega_k$, the value of the weight when the vehicle is in emergency. Fig. 6 shows that a larger $\omega_k$ will result in a lower average $Q_T^k$ when the vehicle is in the 'high' state or in emergency, because a larger $\omega_k$ improves the probability of updates in emergency. However, a large number of updates in emergency will also lead to the rapid growth of virtual queue length, which will cause a lack of update resources in the 'low' state. Fig. 9 shows that when the weight in the 'high' state is relatively large, such as when $\omega_k = 1000$, the average age increased because there will always be cases where there is no update for a long time in the 'low' state.

We did not use UoI here as the basis for choosing $\omega_k$, for UoI is the product of the weight and the cost function of the estimation error. Therefore there may be a situation where average UoI is big but the average estimation error is small in emergency, and the estimation error can already stratifies the system requirement when the average UoI may fail to meet the standard of the system.

Thus in this case, we turn to AoI and the estimation error to
help us find an optimal $\omega_e$. We can conclude that if we only focus on the estimation error in emergency, then a larger $\omega_e$ and a smaller $V$ will be good choices. If we focus on the fairness of updates over the two states, then a moderate $\omega_e$ and $V$ are needed.

VI. CONCLUSIONS

In this work, we study how to minimize UoI, which is a new metric proposed in [13] to characterize information freshness. Unlike [13], the proposed dynamic threshold-based scheduling policy only requires the conditional expectation of the context-aware weight of the next time slot to make decision in the current slot. To solve this problem, we further use SARSA algorithm to obtain a scheduling policy in an unknown environment where the conditional expectation of the weight is unavailable. Simulations show that both policies achieve a near-optimal performance when the update resources are relatively rich. Meanwhile, the SARSA-based policy achieves a lower average UoI than the dynamic threshold-based scheduling policy when the update resources are scarce. We further study how to choose the proper context-aware weights and the parameters used in the proposed policies. According to the results, if we only focus on the estimation error in emergency states, then a larger $\omega_e$ and a smaller $V$ will be good choices. If our objective is the fairness of updates over all the states, then a moderate $\omega_e$ and $V$ are needed.

ACKNOWLEDGMENT

This work is sponsored in part by the Nature Science Foundation of China (No. 61871254, No. 91638204, No. 61861136003), and Intel Collaborative Research Institute for Intelligent and Automated Connected Vehicles.

REFERENCES