Chase or Wait: Dynamic UAV Deployment to Learn and Catch Time-Varying User Activities

Article *in* IEEE Transactions on Mobile Computing · August 2021

CITATIONS
0

READS
11

2 authors:

Zhe Wang
Singapore University of Technology and Design
7 PUBLICATIONS **151** CITATIONS
SEE PROFILE

Lingjie Duan
Singapore University of Technology and Design
173 PUBLICATIONS **3,856** CITATIONS
SEE PROFILE

Chase or Wait: Dynamic UAV Deployment to Learn and Catch Time-Varying User Activities

Zhe Wang, *Member, IEEE*, Lingjie Duan, *Senior Member, IEEE*

Abstract—Unmanned aerial vehicle (UAV) technology is a promising solution for rapidly providing wireless communication services to ground users, where a UAV has limited service coverage and needs to fly through users at different locations for serving them locally. The existing UAV deployment studies largely assume the users' demands do not change during UAV deployment. When the users' demands dynamically change over time, the key challenge is how to adapt the UAV deployment strategy to the partial and even outdated observations on the users' activities given the UAV's flying speed limit. In this paper, we study dynamic UAV deployment to learn and adapt to the time-varying user activities, where the activity pattern of a user (if out of the UAV service coverage) is hidden from the UAV and follows a time-slotted Markov chain that switches between active and idle states. We formulate the learning-and-adaption based UAV deployment problem as a partially observable Markov decision process (POMDP) to maximize the total discounted hit rate of active users, where the UAV decides for itself whether to chase an active user in a distant location (with delayed reward) or to wait for the idle user in the current location to return to the active state (with smaller service probability) over time. We show there is a fundamental delay-reward tradeoff, and prove that the UAV will optimally follow a threshold-based policy by waiting at an idle user for a time threshold before moving to another user. We also show the UAV is more likely to move if the temporal correlation of each user's idling pattern is stronger or the travel distance between users is shorter. Furthermore, we extend to a more general scenario where the UAV does not even know the parameters of each user's temporal activity distribution, and apply Q-learning to develop another threshold-based deployment policy for a multi-user scenario.

Index Terms—Unmanned aerial vehicle, dynamic deployment, partially observable MDP, reinforcement learning.

1 Introduction

Unmanned aerial vehicle (UAV) technology recently emerges as a promising solution to rapidly provide wireless communication and edge computing services to ground users [1]. For example, AT&T and Verizon are now deploying all-weather UAVs as flying stations to enhance the capacity of hot-spot areas of the cellular networks or establish connectivity in the remote areas or disaster scenes that are beyond the reach of ground infrastructure [2]. The UAV-aided wireless communications have been widely studied in various application scenarios such as hot-spot coverage [3], [4], spectrum sharing [5], [6], edge caching [7], data collection [8], computation offloading [9], and communication relaying [10].

Compared with terrestrial base stations, UAV base stations have the advantages of rapid establishment, controlled mobility, and line-of-sight (LoS) air-to-ground channels. With these advantages, the UAV can efficiently enhance the communication quality of the ground user network by jointly optimizing its deployment location, trajectory and resource allocation [3]–[17]. In [3]–[10], the optimal UAV trajectory is designed offline to maximize the network performance (e.g., sum throughput) or minimize the energy consumptions while satisfying certain quality-of-service (QoS) requirements. The existing methods assume the system information (e.g., users' locations, channel states) is static (does not change during the UAV deployment) and the UAV deployment may experience unexpected failure when the environment is dynamic and complex. To deal with the system dynamics, [11]–[16] optimize the fixed UAV locations from an average perspective, where the expected network performance is maximized by taking average over the system dynamics (e.g., users' spatial randomness under certain distributions). In [17], the flight control problem in a dynamic environment is decomposed into multiple sub-problems and solved by convex optimization techniques. However, it usually has high complexity to optimize the UAV trajectory in a dynamic system using the optimization-based methods. Due to the spatial correlation and delayed reward along the UAV trajectory, the myopic UAV deployment policies obtained via decomposition may not guarantee the long-term optimality for the whole path.

The recent works [18]–[31] adopt Markov decision process (MDP) to formulate the sequential UAV deployment decision problems to maximize the expected long-term reward in the complex and dynamic communication systems, where the UAV observes the environment state (e.g., its current location, users' locations [18], channel states [28]), takes an action (e.g., displacement direction and distance), and obtains the corresponding reward. Most of the literature utilizes model-free reinforcement learning (RL) approaches to solve the MDP problems without explicitly requiring the system parameters in advance. The UAV finds its near-optimal path by interacting with the environment in a trial-and-error manner with the aim of maximizing the total accumulated system reward, e.g., total offloaded/collected data [18]–[20], sum rate [21]–[23], wireless coverage [24]–

- *Z. Wang is with School of Computer Science and Engineering, Nanjing University of Science and Technology, China (email: zwang@njust.edu.cn). She was with Pillar of Engineering Systems and Design, Singapore University of Technology and Design.*
- *L. Duan is with Pillar of Engineering Systems and Design, Singapore University of Technology and Design, Singapore (e-mail: lingjie_duan@sutd.edu.sg)*

[28], or minimizing the accumulated system cost, e.g., mission completion time [29], age-of-information [30], energy consumption [31]. Recent advances of both discrete time reinforcement learning algorithms, e.g., Q-learning [24], deep Q network (DQN) [21], [25], double deep Q network (DDQN) [18], [19], [28], [29], and continuous time reinforcement learning algorithms, e.g., deterministic policy gradient (DPG) [22], deep DPG (DDPG) [20], [26], [30], [31], and multi-user DDPG (MA-DDPG) [23], [27], have been applied to solve the trajectory design problems.

The aforementioned literature has overlooked two important issues. First, the existing literature of [18]–[31] assumes that the UAV has full observation about the current states of all users the system in each realization. In practice, due to the limited coverage capability, the UAV has local observations only (e.g., covered users' states) and may not know the exact state of the users outside its coverage. The conventional RL algorithm cannot be directly applied to the scenario with partial observations. Second, the existing literature [18]–[31] assumes that all users always have data to transmit and does not address the temporal dynamics of users' data traffic demand. Due to the limited flying speed, there is a non-negligible delay when the UAV is travelling between different user locations, and plans may fall behind changes. For example, once some user's activity or demand changes (e.g., from active to idle or absent state) in the mean time, it just wastes time for the UAV to reach there without collecting any service reward. Therefore, the key challenge for designing the adaptive UAV deployment in a temporal dynamic environment is how to learn users' activities over time and adapt the UAV location according to the partial and even outdated observations. In this work, we target at dealing with this challenge by analyzing the delay-reward tradeoff: is it worthwhile for the UAV to explore the far/uncovered user's demand that potentially brings higher reward at the cost of flying delay. This tradeoff should widely exist in many UAV deployment problem with time-varying user demand in general but has been overlooked in the existing literature.

We formulate the UAV's learning-and-adaptation based deployment problem as a partially observable Markov decision process (POMDP) by characterizing two key features about user activities: temporal activity correlation per individual user, and hidden user information from the UAV due to the limited coverage. First, a user's data traffic for many IoT and wireless applications is temporally correlated and bursty, which can be well described by Markov modulated models [32]. If a user is not requesting the UAV service now, it is more likely to be idle than active in next time slot; while as time elapses, the belief of active probability for this user increases and the temporal correlation weakens. The UAV is able to serve a user only if it reaches this user right at its active state. Otherwise, the demand of service will be lost. Second, the UAV with limited coverage can only observe the user activity (active or idle) locally and it is difficult to observe the users in far distance. Being aware of the temporal activity correlation, the UAV should dynamically update the belief state over time that characterizes the active probabilities of the uncovered users based on the observation history.

In the broader literature of service provision (not limited to UAV-provided wireless services), POMDP has been studied to solve online decision-making problems, including packet scheduling [33], channel probing [34], spectrum access [35], and route selection [36]. The tractable solution for the optimal policy of POMDP is usually difficult to obtain, even for the system with a small number of states. In our problem, the unique flying delay of the UAV brings in more difficulties to the POMDP analysis. Due to the reduced feasible region of the belief state, our optimal strategy at the boundary points of the feasible region is no longer fixed but varies across different system parameters. This requires new intensive analysis which is different from the literature.

We summarize our key novelty and main contributions as follows.

- *Dynamic UAV deployment approach to learn and adapt to user's activities*: In Section 2, we practically characterize the limited UAV speed and the temporal correlation per user activity, where each user's activity pattern follows an active/idle Markov chain and (if outside the UAV's service coverage) is hidden from the UAV. We formulate the problem as a POMDP, where the UAV observes the exact state of the covered user and updates the belief state of the uncovered user based on the most recent visiting history. Our POMDP model captures the fundamental *delay-reward tradeoff* for deployment: the UAV at a time may either chase the uncovered user with a higher active probability but at the cost of delay; or stay in the current location to wait for the covered user to return to be active immediately in the next slot without any delay in service reward.
- *Optimal threshold-based deployment policy in closed-form*: The analytical results show that the UAV will not leave the covered user as long as it is active. If the covered user is idle instead, we prove in Section 3 that the optimal deployment policy is of a threshold-type on the belief state: the UAV will wait for a time threshold before moving to another location. We derive the waiting time threshold in closed-form in Section 4 by solving the POMDP problem. In Section 4.4, we extend the dynamic UAV deployment solution to the asymmetric scenario with non-identical Markov chains. We manage to prove that the optimal policy is still of a threshold-based type, yet there exists two waiting time thresholds for the two asymmetric users.
- *Reinforcement learning approach with partial observations*: In Section 5.1, we further consider a more challenging scenario where the UAV does not even know the system parameters of user activities in the POMDP model. As the UAV can no longer update the user activity belief based on the transition probabilities, we alternatively propose another threshold-based learning-and-adaptation policy. The memory buffer is purposely truncated to keep system states finite and the complexity low to update a Q-table, and we show that our refined Q-learning algorithm closely approaches the optimal policy (knowing the model parameters ideally) even if we keep a small memory buffer size.

- *Extension to serve multiple users*: In Section 5.2, we extend the dynamic deployment solution from the two-user case to general multi-user cases with line topology and ring toplolgy. In the refined Q-learning algorithm, the UAV not only decides to wait or chase, but also chooses to chase which user in the next step. Our results show that the optimal policy still follows the threshold-based structure, where the UAV is more likely to explore the opportunities at the uncovered users as the user number increases.

2 System Model and Problem Formulation

As illustrated in Fig. 1, we first consider a simplest possible but fundamental scenario where a UAV travels to serve two ground users locally at diverse locations. We model the temporal activity correlation per user in the sense that each user's active/idle activity pattern follows an identical time-slotted Markov chain, since a user's data traffic is temporally correlated and bursty [32]. Later in Section 4.4, we will extend to study the more general case of non-identical Markov activity models. The notations used in this paper are listed in Table 1.

We consider the UAV is operating at a fixed altitude under the air traffic control. Consider the ground users (if active) request to transmit uplink information to the UAV with fixed rate, where the UAV is able to decode the message from the user if the received signal-to-noise ratio (SNR) is above a certain threshold. Since the communication channel between the UAV and the ground user is usually dominated by the line-of-sight (LoS) link (e.g., [2], [8], [12], [37]), the UAV's decoding or service region can be approximated as a disk centered at each user at the UAV's altitude, inside which the received SNR at the UAV is above the threshold and the UAV is able to decode the message from this user. Consider hovering points A and B are at the boundaries of the decoding regions of user A and user B, respectively, where the UAV's received SNR at each hovering point equals to the decoding threshold. When the UAV is right above the hovering point of an active user, we normalize the service time duration of an active request as one time slot. The length of the time slot is determined by the traffic patterns of the user demands. We consider it takes n time slots for the UAV to fly from one hovering point to the other one in a straight line trajectory as the shortest path. To best serve the two users, it is unnecessary for the UAV to enter into the decoding regions. Thus, the UAV chooses its location dynamically between hovering points A and B to cover at most one user at a time. Once the UAV moves outside a decoding region from a hovering point, its received SNR from this user falls below the threshold and we model this as a zero reward for service failure.

In the following, we introduce our system model in details to characterize the temporal correlation per user activity and the hidden user information from the UAV, motivating us to propose POMDP formulation of dynamic UAV deployment.

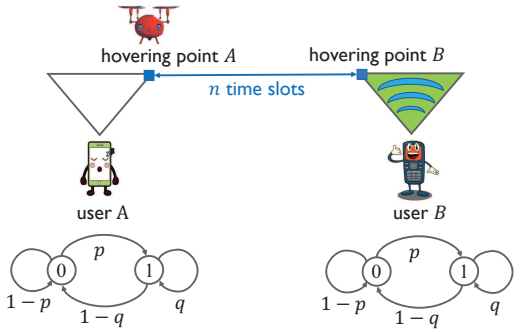

Fig. 1. System model for deploying the UAV to serve two users with n time slots in between for the UAV to fly through. User A's activity state idle/active or equivalently $0/1$ evolves according to two Markov chains over time, where 1 (or 0) tells the user is (not) requesting the UAV service.

2.1 States and Beliefs about Time-varying User Activities

If the UAV is currently at the hovering point for serving user A (or B) in Fig. 1, we call this user as the covered user by the UAV and the other user B (or A) as the uncovered user.[1] Within the UAV coverage, the covered user may or may not be active to request the wireless service, depending on its time-varying activity. Let $x_t \in \{0,1\}$ and $y_t \in \{0,1\}$ denote the states of the covered user and uncovered user by the UAV at time slot t, respectively. x_t and y_t evolves over time independently, and are modeled as two identical two-state Markov chains: state 1 for active mode of requesting UAV service and state 0 for idle/absent mode. We consider the $0/1$ state transition occurs at the beginning of each time slot and the state remains unchanged within this slot. As illustrated in Fig. 1, the transition probabilities of the covered user are given by $\Pr(x_{t+1}=1|x_t=0) = p$ and $\Pr(x_{t+1}=1|x_t=1) = q$, and we also have $\Pr(y_{t+1} = 1|y_t = 0) = p$ and $\Pr(y_{t+1} = 1|y_t = 1) = q$ for the uncovered user as their activities follow identical Markov chains. We practically model the temporal correlation of data traffic per user by using $q > 1/2 > p$, which ensures that the user in state $0/1$ is more likely to keep the same state in next time slot.

The UAV only knows the state x_t of the covered user but does not know y_t of the uncovered user due to its limited wireless coverage. We denote b_t as the UAV's belief about the probability that the uncovered user is active in time slot t given its observation history (at least n time slots ago) on that user, i.e.,

$$b_t = \Pr[y_t = 1|\text{history until time } t]. \quad (1)$$

The state of the system at time slot t is denoted as (x_t, b_t).

Due to the temporal correlation, the UAV can further deduce the user states in the future time slots based on the current observation/belief, which applies to both covered and uncovered users. If the UAV has an initial belief (i.e., active probability) b of uncovered user in (1), the belief

1. When the UAV is travelling between users, it does not cover or serve anyone. As shown in the POMDP model later, we can equivalently skip these n instances in our POMDP model by updating n-step system state. Then it is equivalent to say that the UAV can cover at most one user here.

TABLE 1
Key Notations

Notation	Description
p	transition probability from state 0 to 1
q	transition probability from state 1 to 1
a	action for the UAV
S	action of staying with the covered user
M	action of moving to the uncovered user
x	state of the covered user
y	state of the uncovered user
b	belief of the active probability of the uncovered user
b_L	lower bound of the feasible belief region
b_H	upper bound of the feasible belief region
b_{th}	belief threshold beyond which to move
n	number of time slots to fly between users
k	number of successive time slots a user is unvisited
k_{th}	waiting time threshold beyond which to move
γ	discount factor
$T(\cdot)$	first-step forward evolution operator
$T^k(\cdot)$	k-th step forward evolution operator
$T^{-k}(\cdot)$	k-th step backward evolution operator
$V_a(x,b)$	value function given state (x,b) and action a
$\pi(x,b)$	policy for state (x,b)
ϕ_S	set of beliefs under which it is optimal to stay
ϕ_M	set of beliefs under which it is optimal to move
$\tilde{z}$	index of the covered user
z	index of the uncovered user

that this user is active at the beginning of the next slot is updated according to the following first-step forward evolution operator $T(b)$:

$$T(b) = p(1-b) + qb = p + (q-p)b, \tag{2}$$

where it can turn from idle to active with probability p or keep active with probability q as shown in Fig. 1. Note that $q - p \in [0, 1]$. Similarly, we can also apply this operator to update the future active belief of the covered user given the deterministic current state $x = 0$ (or 1). Using mathematical induction, we obtain $T^k(\cdot)$ and $T^{-k}(\cdot)$, for $k \in \{0, 1, \cdots\}$, as the k-th step forward and backward belief evolution operators, respectively. That is,

$$T^k(b) = T(T^{k-1}(b)) = (q-p)^k(b - b_H) + b_H, \tag{3}$$

$$\begin{aligned} T^{-k}(b) &= T^{-1}(T^{-(k-1)}(b)) \\ &= \frac{b}{(q-p)^k} - \frac{1-(q-p)^k}{1-(q-p)} \frac{p}{(q-p)^k}, \end{aligned} \tag{4}$$

where

$$b_H = \frac{p}{1-(q-p)}. \tag{5}$$

The function $T^k(b)$ represents, with an initial belief of b, the belief of an uncovered user after k time slots unvisited by the UAV. Once this user is visited, the UAV learns its realized state x and the other user becomes uncovered user to update the belief. If the UAV departs from an idle user (who becomes uncovered with initial belief $b = 0$) and this user is unvisited afterwards, $T^k(b)$ in (3) increases with k, telling that the temporal correlation in this user's activity from idle to idle state weakens over time. If this user is unvisited for a sufficiently long time ($k \to \infty$), the belief on active state of this user is $T^\infty(b) = b_H$ in (5), which is $1/2$ if the two-state Markov chain is symmetric (i.e., $q = 1 - p$).

Next, we will formulate the POMDP problem of optimal learning-and-adaption policy for UAV deployment.

2.2 POMDP Problem for Dynamic UAV Deployment

If the UAV is above a user (i.e., at the hovering point of user A or B in Fig. 1) at the beginning of time slot t, it needs to choose an actions a_t out of the action set $\{S, M\}$, where $a_t = S$ tells that the UAV chooses to stay above the covered user and $a_t = M$ tells moving towards the uncovered user. Note that if $a_t = M$, it is not optimal for the UAV to return in the midway before reaching the other hovering point, since it collects no new information in the trip and will not change its moving decision. We denote $r_{a_t}(x_t, b_t)$ as the immediate service reward that the UAV receives for being in state (x_t, b_t) and taking action a_t. Consider the UAV takes the action of $a_t = S$, it receives a positive service reward R if the covered user is active ($x_t = 1$) or zero reward if the covered user is idle ($x_t = 0$). Consider the UAV takes the action of $a_t = M$, it will immediately fly outside the coverage region of the currently covered user and receive zero reward in this time slot. Based on the above case discussions, the reward in the current slot is given by

$$r_{a_t}(x_t, b_t) = \begin{cases} R, \text{ if } x_t = 1 \text{ and } a_t = S & \text{(6a)} \\ 0, \text{ otherwise.} & \text{(6b)} \end{cases}$$

We formulate the POMDP problem as follows. Due to the temporal correlation of the activity states in the Markov chains, the UAV's executed action has effect not only on the reward in the current time slot but also on those in the future time slots. We consider the reward is discounted over time by a discount factor γ, $0 < \gamma < 1$, which means the future reward is relatively less important than the current reward. We define a policy π as a rule for selecting the action for any state, which is a mapping from the state space of (x_t, b_t) to the action space of a_t.

Let $V_\pi(x, b)$ be the expectation of total discounted reward given the initial state (x_0, b_0) and the policy π to be employed, it is represented as

$$V_\pi(x, b) = \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t r_{a_t}(x_t, b_t) \Big| (x_0, b_0) \right]. \tag{7}$$

The expectation is taken over all transition possibilities of the states between any two consecutive time slots. Without much loss of generality, we consider an infinite horizon time span in this paper, resulting in the stationary policies which do not change with time [38]. The infinite-time horizon is reasonable since the adaptation time (e.g., in seconds or minutes) of the UAV is much smaller than its total operation time (e.g., in hours) and we still use the discount factor γ to penalize the future reward. Now we define the value function $V(x, b)$ as

$$V(x, b) = \max_\pi V_\pi(x, b), \tag{8}$$

where from now on we skip the time subscript to fit the stationary policies. According to [38, Theorem 6.3], there exists a unique stationary policy π^* such that $V(x, b) = \max_\pi V_\pi(x, b)$. This value function satisfies the Bellman equation, i.e.,

$$V(x, b) = \max_{a \in \{S, M\}} \{V_a(x, b)\}, \tag{9}$$

where $V_a(x,b)$ is the long-term value obtained by taking action a when the system state is (x,b).

3 Analysis for Structures of Optimal Deployment Policy

In this section, we first derive the value functions $V_a(x,b)$ in the Bellman equation (9), depending on whether the currently covered user is idle or active. Then we will prove the structural results of the optimal policy for deciding the learning-and-adaption based UAV deployment.

3.1 Analysis of Value Functions for POMDP

We discuss the value functions depending on current observation ($x = 0/1$) and the UAV's action $a \in \{S, M\}$. Consider that the UAV is currently covering user A, its beliefs on the active probability of two users are presented in Fig. 2.

3.1.1 The Covered User is Currently Idle ($x = 0$)

When the covered user is idle ($x = 0$), the UAV receives zero reward in the current time slot regardless of its action. The value function is given by

$$V_a(0,b) = \gamma \mathbb{E}_{(x',b')}\left[V(x',b')|(0,b),a\right], \tag{10}$$

where $(0,b)$ is the state of the system in the current time slot, and (x',b') is the state of the system in the next slot. Notice that $V(x',b')$ implies the optimal decision is taken for the subsequent stages. If the UAV chooses to stay ($a = S$) as shown in Fig. 2(i), based on the transition probabilities, we deduce that the active and idle probabilities of the covered user in the next time slot are $T(0) = p$ and $1 - T(0) = 1 - p$, respectively. Given the belief on the uncovered user is b, the belief in the next slot will be $T(b)$ in (2). The value function is given by

$$V_S(0,b) = \gamma\left[(1-p)V(0,T(b)) + pV(1,T(b))\right]. \tag{11}$$

If the UAV chooses to move ($a = M$) as shown in Fig. 2(ii), it will continue to receive zero reward in the next $n-1$ time slots until it reaches the uncovered user at the n-th time slot to observe the exact state. Notice that since the UAV moves from one user to the other one, the original covered user at the current slot will be the uncovered user at the n-th slot. Given the original covered user is idle ($x = 0$), we can learn that its active belief at n-th time slot is $T^n(0)$. Moreover, given the belief of the originally uncovered user is b, its active probability is $T^n(b)$ and idle probability is $1 - T^n(b)$ at the n-th time slot. The value function is rewritten as

$$\begin{aligned} V_M(0,b) = \gamma^n &\left[(1 - T^n(b))V(0,T^n(0))\right. \\ &\left. + T^n(b)V(1,T^n(0))\right]. \end{aligned} \tag{12}$$

In this expression, we use γ^n directly since there is zero reward for $n-1$ consecutive time slots. The Bellman equation is to pick the better action by comparing (11) and (12), i.e.,

$$V(0,b) = \max\{V_S(0,b), V_M(0,b)\}. \tag{13}$$

We are not sure which is greater: (11) or (12). Intuitively, given the covered user is idle, the UAV should decide whether to play safe by waiting for the undelayed reward at the covered user, or to take risk by chasing the delayed reward at the uncovered user. If the UAV chooses to stay, the covered user may return to be active in next time slot only with probability $p < 1/2$. If the UAV chooses to move, it will waste $n-1$ time slots on the way until it reaches the other user who is more likely to be active. We will analyze this delay-reward tradeoff by solving the POMDP problem.

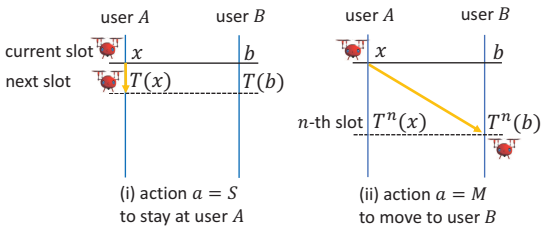

Fig. 2. The UAV's beliefs on the active probabilities of the two users learned at the current time slot. It takes n time slots for the UAV to travel from one user to the other. Consider the UAV is above user A at the current slot and the initial state of the system is (x,b). The UAV needs to decide to stay or move: (i) if the UAV takes the action of $a = S$ to stay at the covered user, the active probability of user A and user B will be $T(x)$ and $T(b)$ in the next slot, where $T(x=0) = p$ and $T(x=1) = q$; (ii) if the UAV takes the action of $a = M$, it will arrive at user B at the n-th slot, where the active probability of user A and user B will be $T^n(x)$ and $T^n(b)$, respectively.

3.1.2 The Covered User is Currently Active ($x = 1$)

If the covered user is active ($x = 1$) and the UAV chooses to stay ($a = S$) as shown in Fig. 2(i), it receives a positive reward $r_S(1,b) = R$ in the current time. We can learn that the active and idle probabilities of the covered user in the next time slot is $T(1) = q$ and $1 - T(1) = 1 - q$, respectively. Similar to (11), the value function is given by

$$V_S(1,b) = R + \gamma\left[(1-q)V(0,T(b)) + qV(1,T(b))\right]. \tag{14}$$

If the UAV chooses to move ($a = M$) as shown in Fig. 2(ii), the initially covered user in active state ($x = 1$) becomes uncovered after the UAV moves to the opposite user. We update the belief of his active probability to $T^n(1)$ at the n-th time slot. Moreover, given the belief of the originally uncovered user is b, its active probability is $T^n(b)$ and idle probability is $1 - T^n(b)$ at the n-th time slot. Similar to (12), the value function is

$$\begin{aligned} V_M(1,b) = \gamma^n &\left[(1 - T^n(b))V(0,T^n(1))\right. \\ &\left. + T^n(b)V(1,T^n(1))\right]. \end{aligned} \tag{15}$$

By comparing (14) and (15), we prove the following lemma.

Lemma 1. *Given the covered user is active now ($x = 1$), the UAV's optimal strategy is to stay with this user as long as it is active (i.e., $\pi^*(x = 1, b) = S$) regardless of the belief of the uncovered user. That is, $V_S(1,b) > V_M(1,b)$ in (9) with $x = 1$.*

It is not worthwhile for the UAV to depart from the active user to the other user (even sure to be active), as it loses the immediate reward R and wastes n time slots during the flight. We therefore simplify the Bellman equation in (9) as

$$V(1,b) = \max\{V_S(1,b), V_M(1,b)\} = V_S(1,b). \tag{16}$$

Before we proceed to analyze the structure of the optimal policy, we first discuss the feasible region of the belief b on

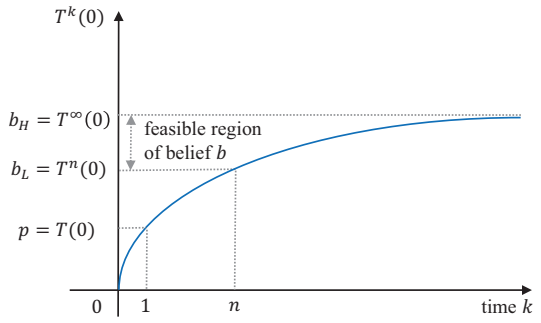

Fig. 3. Illustration of belief evolution over time about an idle user.

the uncovered user in Fig. 3. As discussed in Lemma 1, the UAV will depart from a user only if it is idle. Once the UAV departs from an idle user ($x = 0$), its belief that this user is active at the k-th time slot is $T^k(0)$ which increases as k further increases. We can learn that the lower bound of reasonable belief b for the uncovered user is achieved when the UAV just arrives at the covered user from the other user, i.e., $b_L = T^n(0)$. As given in (5) of Section 2.1, the upper bound of the belief b is obtained if this user has not been visited for sufficiently long time, where a steady state of $b_H = T^\infty(0)$ is approached asymptotically.

Lemma 2. *The active belief b of the uncovered user falls into the region of $[b_L, b_H]$, where $b_L = T^n(0)$ and $b_H = T^\infty(0) = \frac{p}{1-(q-p)}$. And $T^k(b)$ in* (3) *is increasing in unvisited step number k for $k \geq n$.*

3.2 Structure of Optimal Policy

In this subsection, we will show that the optimal policy is of a threshold-based type with respect to the belief state b of the uncovered user. We begin to prove some properties of the value functions in (13) and (16) and the belief sets.

Lemma 3. *Both value functions of $V(0,b)$ in* (13) *and $V(1,b)$ in* (16) *are convex and non-decreasing in active belief b of the uncovered user. Furthermore, $V(1,b) > V(0,b)$.*

Proof. We here generalize our infinite-horizon value function and denote $V(0,b;k)$ as the optimal value when the time spans only k time stages. First, since $T^n(b)$ in (3) is linear in b, $V_M(0,b;k)$ relaxed from (12) is affine in b. Then, we prove the convexity of $V_S(0,b;k)$ and $V(1,b;k)$ using mathematical induction. For $k = 1$, the reward only depends on the current slot, i.e., $V_S(0,b;1) = 0$ and $V(1,b;1) = R$. Then suppose that $V_S(0,b;k-1)$ and $V(1,b;k-1)$ are both convex in b, we can deduce that $V(0,b;k-1)$ is convex in b due to convex $V_M(0,b;k-1)$. Now we want to prove that $V_S(0,b;k)$ and $V(1,b;k)$ are also convex in b. For any believes b_1, b_2, and $\beta \in [0,1]$, based on (11), we have

$$\begin{aligned}
&V_S(0, \beta b_1 + (1-\beta)b_2; k) \\
&= \gamma\left[(1-p)V(0, T(\beta b_1 + (1-\beta)b_2); k-1)\right. \\
&\quad \left. + pV(1, T(\beta b_1 + (1-\beta)b_2); k-1)\right] \\
&\overset{(i)}{=} \gamma\left[(1-p)V(0, \beta T(b_1) + (1-\beta)T(b_2); k-1)\right. \\
&\quad \left. + pV(1, \beta T(b_1; k-1) + (1-\beta)T(b_2, k-1)\right] \\
&\overset{(ii)}{\leq} \gamma\left[\beta(1-p)V(0, T(b_1); k-1)\right. \\
&\quad + (1-\beta)(1-p)V(0, T(b_2); k-1) \\
&\quad \left. + \beta pV(1, T(b_1); k-1) + (1-\beta)pV(1, T(b_2); k-1)\right] \\
&= \beta V_S(0, b_1; k) + (1-\beta)V_S(0, b_2; k), \qquad (17)
\end{aligned}$$

where equality (i) follows the linearity of $T(b)$ in b in (2), and inequality (ii) is due to the convexity assumption of $V(0,b;k-1)$ and $V(1,b;k-1)$. Similar to (17), we can also prove that $V(1,b;k)$ is convex in b. By taking $k \to \infty$, we prove that both $V_S(0,b)$ in (11) (and thus $V(0,b)$ in (13)) and $V(1,b)$ in (16) are convex in b.

Next, we prove the non-decreasing property of the value functions. We can easily prove that $V(1,b) > V(0,b)$, where the detailed proof is omitted here. Since $T^n(b)$ is increasing in $b < b_H$ and $V_S(1, T^n(0)) > V_S(0, T^n(0))$, $V_M(0,b)$ in (12) is non-decreasing in b. Further, we prove that $V_S(0,b;k)$ and $V(1,b;k)$ are non-decreasing in b using mathematical induction. For $k = 1$, we have $V_S(0,b;1) = 0$ and $V(1,b;1) = R$. Then we assume $V(0,b;k-1)$ and $V(1,b;k-1)$ are non-decreasing. Since $T(b)$ in (2) is non-decreasing in b, $V(0, T(b); k-1)$ and $V(1, T(b); k-1)$ are non-decreasing. According to (11), we write $V_S(0,b;k)$ as

$$\begin{aligned}
V_S(0,b;k) = \gamma\left[(1-p)V(0, T(b); k-1)\right. \\
\left. + pV(1, T(b); k-1)\right], \qquad (18)
\end{aligned}$$

which is non-decreasing in b. Similarly, we can also prove that $V(1,b;k)$ is non-decreasing in b. By taking $k \to \infty$, we prove that both $V_S(0,b)$ (and thus $V(0,b)$) and $V(1,b)$ are non-decreasing in b. □

Intuitively, the total discounted reward increases when the uncovered user potentially has a higher active probability to explore. In other words, if the UAV moves to the uncovered user, it has a higher chance to catch an active user than an idle user.

We further define the set of belief states under which it is optimal to take action a as

$$\phi_a = \{b \in [b_L, b_H] | V(0,b) = V_a(0,b)\}, \ a \in \{S, M\}. \qquad (19)$$

Specifically, ϕ_S and ϕ_M denote the sets of beliefs under which it is optimal to take actions of S and M, respectively. We obtain the property of ϕ_M in the following lemma.

Lemma 4. *The set of beliefs ϕ_M for the UAV to move from an idle user is convex in the belief b of the uncovered user.*

Proof. For any beliefs $b_1, b_2 \in \phi_M$ and $\beta \in [0,1]$,

$$\begin{aligned}
&V(0, \beta b_1 + (1-\beta)b_2) \overset{(i)}{\leq} \beta V(0, b_1) + (1-\beta)V(0, b_2) \\
&\overset{(ii)}{=} \beta V_M(0, b_1) + (1-\beta)V_M(0, b_2) \\
&\overset{(iii)}{=} \gamma^n[V(0, T^n(0)) + (\beta T^n(b_1) + (1-\beta)T^n(b_2)) \\
&\quad \times (V(1, T^n(0)) - V(0, T^n(0)))] \\
&\overset{(iv)}{=} \gamma^n[V(0, T^n(0)) + T^n(\beta b_1 + (1-\beta)b_2) \\
&\quad \times (V(1, T^n(0)) - V(0, T^n(0)))] \\
&= V_M(0, \beta b_1 + (1-\beta)b_2) \overset{(v)}{\leq} V(0, \beta b_1 + (1-\beta)b_2), \qquad (20)
\end{aligned}$$

where inequality (i) follows the convexity property of $V(0,b)$ given in Lemma 1, equality (ii) holds due to $b_1, b_2 \in$

ϕ_M, equality (iii) uses $V_M(0,b)$ in (12), equality (iv) comes from the property that $T^n(b)$ in (3) is linear in b, and equality (v) is due to the fact that $V(0,b)$ is the optimal value function. Consequently, the first term $V(0,\beta b_1+(1-\beta)b_2)$ is the same as the last term $V(0,\beta b_1+(1-\beta)b_2)$ in (20), and thus both inequalities (i) and (v) in (20) are tight. That is, $V_M(0,\beta b_1+(1-\beta)b_2)=V(0,\beta b_1+(1-\beta)b_2)$ and belief $\beta b_1+(1-\beta)b_2\in\phi_M$, which proves the convexity of ϕ_M. □

Remark 1. *The proof of the structural policy for the POMDP problem in this work is more challenging than the literature. First, it is difficult to directly prove the convexity of ϕ_S following the similar method as ϕ_M. Moreover, we are not sure about the optimal policy at the boundary points of b_L and b_H due to the delay variable n. In most literature, optimal policy at the boundaries can often be easily deduced, which much simplifies the discussions. In our problem, if there no such delay (i.e., $n=1$), we can know that the optimal policy for the UAV is to move whenever the covered user is idle (i.e., to move at both b_L and b_H). If the delay cost n is very large (e.g., $n\to\infty$), the UAV does not bother to fly even if the covered user has been idle for sufficiently long time (i.e., to stay at both b_L and b_H). For some other medium n, the optimal policy at b_L and b_H cannot be easily deduced. Intuitively, the UAV may be more willing to fly for small n and more reluctant to move for large n.*

To find out the optimal policy structure, we discuss all three possibilities at the boundary points of $[b_L,b_H]$ given convex set ϕ_M:

- $\pi^*(0,b_H)=S$;
- $\pi^*(0,b_L)=\pi^*(0,b_H)=M$;
- $\pi^*(0,b_L)=S$ and $\pi^*(0,b_H)=M$.

Lemma 5. *If the UAV's optimal policy is to stay when the active belief of the uncovered user is $b=b_H$, i.e., $\pi^*(0,b_H)=S$, then it will always stay with the covered user regardless of the belief on the other user, i.e., $\pi^*(0,b)=S$ for any $b\in[b_L,b_H]$.*

Proof. Assume $\pi^*(0,b_H)=S$, we have $V_S(0,b_H)\geq V_M(0,b_H)$ and $V(0,b_H)=\max(V_S(0,b_H),V_M(0,b_H))=V_S(0,b_H)$. Based on (3), we have $T^i(b_H)=b_H$ at steady-state distribution for any $i\geq 0$. Substituting $b=T^n(b_H)=b_H$ into (13) and (16), we have

$$V(0,b_H)=\gamma\left[(1-p)V(0,b_H)+pV(1,b_H)\right], \tag{21}$$

$$V(1,b_H)=R+\gamma\left[(1-q)V(0,b_H)+q_1V(1,b_H)\right]. \tag{22}$$

By jointly solving (21) and (22), we obtain that

$$V(0,b_H)=\frac{p\gamma R}{(1-\gamma)(1-(q-p)\gamma)}, \tag{23}$$

$$V(1,b_H)=\frac{(1-(1-p)\gamma)R}{(1-\gamma)(1-(q-p)\gamma)}. \tag{24}$$

Next, we prove that $V_M(0,T^{-k}(b_H))\leq V_S(0,T^{-k}(b_H))$, $V_S(0,T^{-k}(b_H))=V(0,b_H)$ and $V(1,T^{-k}(b_H))=V(1,b_H)$ hold for all $k\geq 0$. We use backward induction to prove. Assume they hold for $k-1$, i.e., $V_M(0,T^{-(k-1)}(b_H))\leq V_S(0,T^{-(k-1)}(b_H))$, $V_S(0,T^{-(k-1)}(b_H))=V(0,b_H)$ and $V(1,T^{-(k-1)}(b_H))=V(1,b_H)$, we can then replace $V(0,T^{-(k-1)}(b_H))$ and $V(1,T^{-(k-1)}(b_H))$ by (23) and (24) in (25) and (26), i.e.,

$$V_S(0,T^{-k}(b_H))=\gamma\Big[(1-p)V(0,T^{-k+1}(b_H))+pV(1,T^{-k+1}(b_H))\Big] \tag{25}$$

$$V(1,T^{-k}(b_H))=R+\gamma\Big[(1-q_1)V(0,T^{-k+1}(b_H))+qV(1,T^{-k+1}(b_H))\Big]. \tag{26}$$

By further derivation, we obtain that $V_S(0,T^{-k}(b_H))=V_S(0,T^{-(k-1)}(b_H))=V(0,b_H)$ and $V(1,T^{-k}(b_H))=V(1,b_H)$. Based on this result, we will further prove that

$$V_M(0,T^{-k}(b_H))\leq V_S(0,T^{-k}(b_H)) \tag{27}$$

holds given $V_M(0,T^{-(k-1)}(b_H))\leq V_S(0,T^{-(k-1)}(b_H))$ is satisfied. By taking $b=T^{-k}(b_H)$ in (12), we have

$$\begin{aligned}
&V_M(0,T^{-k}(b_H))\\
&=\gamma^n\left[V(0,b_L)+T^{n-k}(b_H)[V(1,b_L)-V(0,b_L)]\right]\\
&\overset{(i)}{\leq}\gamma^n\left[V(0,b_L)+T^{n-(k-1)}(b_H)[V(1,b_L)-V(0,b_L)]\right]\\
&=V_M(0,T^{-(k-1)}(b_H))\leq V_S(0,T^{-(k-1)}(b_H))\\
&=V_S(0,T^{-k}(b_H)).
\end{aligned} \tag{28}$$

The inequality (i) is because $V(1,b)>V(0,b)$ and $T^i(b)$ is increasing in i for $b\in[b_L,b_H]$. Taking $k\to\infty$, we thus can prove that $V_M(0,T^{-k}(b_H))\leq V_S(0,T^{-k}(b_H))$ holds for all $k\geq 0$. In other words, we have $\pi^*(0,b)=S$ for all $b\leq b_H$. □

In Lemma 5, we know that if the optimal policy at b_H is to stay, the UAV should stay for all other feasible belief values and it is not possible for the UAV to move at b_L. Furthermore, assume the optimal policy at b_H is to move, we have the following lemma by using the convex property of the set ϕ_M.

Lemma 6. *If the UAV's optimal policy is to move both at b_L and b_H, i.e., $\pi^*(0,b_L)=\pi^*(0,b_H)=M$, then it will always move to the uncovered user, i.e., $\pi^*(0,b)=M$ for all $b\in[b_L,b_H]$.*

Sketch of proof. According to Lemma 4, we have proved that ϕ_M is a convex set. Assume that the optimal strategies at the belief region's boundary points of $b=b_L$ and $b=b_H$ are to move, i.e., $\pi^*(0,b_L)=\pi^*(0,b_H)=M$. If there exists a belief $b_x\in[b_L,b_H]$ with $\pi^*(0,b_x)=S$, then ϕ_M is no longer convex. We therefore can deduce that $\pi^*(0,b)=M$ holds for any $b\in[b_L,b_H]$. □

Lemma 7. *If $\pi^*(0,b_L)=S$ and $\pi^*(0,b_H)=M$, then there exists a belief threshold $b_{th}\in[b_L,b_H]$ once beyond which the UAV will choose to move, i.e., $\pi^*(0,b)=S$ for $b\in[b_L,b_{th})$ and $\pi^*(0,b)=M$ for $b\in[b_{th},b_H]$.*

Sketch of proof. Assume the optimal strategies at the belief region's boundary points of $b=b_L$ and $b=b_H$ are to stay and move, respectively, i.e., $\pi^*(0,b_L)=S$ and $\pi^*(0,b_H)=M$. Consider the following two subcases. If there does not exist a belief $b\in[b_L,b_H)$ with $\pi^*(0,b)=M$, then $\pi^*(0,b)=S$ holds for any $b\in[b_L,b_H)$ and $\pi^*(0,b)=M$ for $b=b_H$. If there exists a belief $b_{th}\in[b_L,b_H)$ with $\pi^*(0,b_{th})=M$, due

to the convexity of ϕ_M, we can deduce that $\pi^*(0,b) = M$ for any $b \in [b_{th}, b_H]$, and $\pi^*(0,b) = S$ holds for any $b \in [b_L, b_{th}]$. □

In (19), we adopt ϕ_S to represent the belief set to stay and ϕ_M as the belief set to move. Given the conclusion of threshold-based deployment policy with respect to b, in the following section, we will discuss the optimal policies corresponding to Lemmas 5-7 in the following three cases.

- Case I: always stay ($\phi_S = [b_L, b_H]$ and $\phi_M = \emptyset$);
- Case II: always move ($\phi_S = \emptyset$ and $\phi_M = [b_L, b_H]$);
- Case III: wait to move ($\phi_S = [b_L, b_{th})$ and $\phi_M = [b_{th}, b_H]$), where b_{th} is belief threshold to be derived analytically.

In the following section, we will discuss the optimal policies for each case in the adaptive UAV deployment.

4 Optimal Threshold-based Policy in Closed-form

In this section, we will analytically derive the conditions and decision threshold for Cases I-III in closed-form. The basic idea is to first derive the closed-form expressions of $V(0,b)$ and $V(1,b)$ for each case. Then, we derive the condition for each case using these value functions:

- if $V_S(0,b) > V_M(0,b)$ holds for all $b \in [b_L, b_H]$, the optimal policy becomes Case I;
- if $V_S(0,b) < V_M(0,b)$ holds for all $b \in [b_L, b_H]$, the optimal policy becomes Case II;
- otherwise Case III is optimal, and the belief threshold is obtained by solving equation $V_S(0,b) = V_M(0,b)$.

4.1 Case I: Always Stay ($\phi_S = [b_L, b_H]$ and $\phi_M = \emptyset$)

Based on Lemma 5, the UAV's optimal policy is to always stay for all $b \in [b_L, b_H]$ if it is optimal to stay at $b = b_H$. Assume $\pi^*(0, b_H) = S$. By using backward induction as in the proof of Lemma 5, we obtain the expressions of $V(0,b) = V(0,b_H)$ in (23) and $V(1,b) = V(1,b_H)$ in (24), which hold for all $b \in [b_L, b_H]$.

We now derive the condition for Case I to hold. By substituting the expressions of $V(0,b)$ and $V(1,b)$ into (11) and (12) and letting $V_S(0,b) > V_M(0,b)$, the condition for the UAV to stay under each state of $(0,b)$ is thus

$$T^n(b) < \hat{b}_H, \tag{29}$$

where

$$\hat{b}_H = p\frac{1-\gamma^n}{\gamma^{n-1}(1-\gamma)}. \tag{30}$$

From the inequality above, we see that the UAV would choose to stay with the covered user in the current slot if it is too costly to fly to the uncovered user, i.e., the active belief $T^n(b)$ of the uncovered user after n time slots is not large. As $T^n(b)$ is linearly increasing in b, if $T^n(b) < \hat{b}_H$ holds for the upper bound of $b = b_H$ (with $T^n(b_H) = b_H$), it will also hold all other $b < b_H$. Intuitively, the UAV should have the highest incentive to move to the uncovered user if it has not visited to this user for a sufficiently long time. If the UAV chooses to stay with the covered user even under the maximum active belief on the uncovered user, it will stay under any other belief. We therefore present the analytical condition for Case I to be the optimal policy in the following proposition.

Lemma 8. *If $b_H < \hat{b}_H$ (where $\hat{b}_H$ is given in (30)), the optimal policy is to always stay at the currently covered user (i.e., $\pi^*(0,b) = S$) for any active belief $b \in [b_L, b_H]$ about the uncovered user.*

Since $\hat{b}_H$ in (30) is an increasing function of n for any $\gamma \in [0,1]$, we can deduce that Case I is more likely to happen if the travel time n between users is large enough. In addition, we observe that $\hat{b}_H$ increases with p, which means the UAV is more willing to stay if the covered user has a higher probability to turn on in the next time slot.

4.2 Case II: Always Move ($\phi_S = \emptyset$ and $\phi_M = [b_L, b_H]$)

Based on Lemma 6, the UAV's optimal policy is to always move for any $b \in [b_L, b_H]$ if it is optimal to move at both belief boundaries $b = b_L$ and $b = b_H$. In this case, we have $V(0,b) = V_M(0,b)$ and $V_S(0,b) < V_M(0,b)$ for all $b \in [b_L, b_H]$.

To derive the condition for Case II, we first need to derive the expressions of $V(0,b)$ and $V(1,b)$. Given $\pi^*(0,b_L) = \pi^*(0,b_H) = M$. For any $b \in [b_L, b_H]$, we can deduce that $T^{k+1}(b) \in [b_L, b_H]$ and thus have $\pi^*(0, T^{k+1}(b)) = M$ and $V(0, T^{k+1}(b)) = V_M(0, T^{k+1}(b))$ for any $k \geq 0$. Using forward induction, $V(1,b)$ in (16) can be expressed as

$$\begin{aligned} V(1,b) = R\sum_{i=0}^{k} q^i\gamma^i + (1-q)\Big\{ & V(0,b_L)\times \\ & \sum_{i=1}^{k} q^{i-1}\gamma^{n+i}(1-T^{n+i}(b)) + V(1,b_L)\times \\ & \sum_{i=1}^{k} q^{i-1}\gamma^{n+i}T^{n+i}(b)\Big\} + q^k\gamma^{k+1}\Big[(1-q)V(0,T^{k+1}(b)) \\ & +qV(1,T^{k+1}(b))\Big]. \end{aligned} \tag{31}$$

By taking $k \to \infty$ in the infinite time horizon, we have $T^\infty(b) = b_H$ and remove the term of $q^\infty\gamma^\infty = 0$ from (31). As a result, we obtain $V(1,b)$ as a function $V(0,b_L)$ and $V(1,b_L)$, i.e.,

$$V(1,b) = \frac{R}{1-q\gamma} + (1-q)\left[V(0,b_L)f(b) + V(1,b_L)g(b)\right], \tag{32}$$

where the functions of $f(b)$ and $g(b)$ are defined as

$$f(b) = \gamma^{n+1}\left[\frac{1-b_H}{1-q\gamma} + \frac{(q-p)^{n+1}(b_H-b)}{1-(q-p)q\gamma}\right] \tag{33}$$

and

$$g(b) = \gamma^{n+1}\left[\frac{b_H}{1-q\gamma} - \frac{(q-p)^{n+1}(b_H-b)}{1-(q-p)q\gamma}\right]. \tag{34}$$

Since $V(0,b) = V_M(0,b)$ and $b_L = T^n(0)$, we rewrite (13) as

$$V(0,b) = \gamma^n\left[(1-T^n(b))V(0,b_L) + T^n(b)V(1,b_L)\right], \tag{35}$$

which is also a function of $V(0, b_L)$ and $V(1, b_L)$. By jointly solving (32) and (35) at $b = b_L$, we obtain the closed-form expressions of $V(0, b_L)$ and $V(1, b_L)$.

$$V(0, b_L) = \frac{R}{(1 - q\gamma)\left[\rho - (f(b_L) + g(b_L)\rho)(1 - q)\right]} \quad (36)$$

and

$$V(1, b_L) = \rho V(0, b_L), \quad (37)$$

where $\rho = 1 + \frac{1-\gamma^n}{b_H \gamma^n (1-(q-p)^{2n})}$.

Now we are ready to solve the condition for Case II. Substituting the expressions of $V(0, b)$ in (35) and $V(1, b)$ in (32) back into $V_S(0, b)$ in (11) and $V_M(0, b)$ in (12). Let $V_S(0, b) > V_M(0, b)$, we have $b_L > \hat{b}_L$, where $\hat{b}_L$ is given in (38). The results in this case is summarized in the following proposition.

Lemma 9. *If $b_L > \hat{b}_L$ (where $\hat{b}_L$ is given in* (38)*), the optimal policy is to always move from the idle user (i.e., $\pi^*(0, b) = M$) for any active belief $b \in [b_L, b_H]$.*

From this inequality, we see that the UAV will choose to move in the current slot if the belief state of the uncovered user is sufficiently large. As $T^n(b)$ is linearly increasing with b, Intuitively, if the UAV has just arrived at the covered user from the other idle user, the UAV has the smallest active belief on the uncovered user, i.e., $b = b_L$ at $k = n$ in Fig. 3. As time k goes by, the active belief b of the uncovered user increases and the UAV has more incentive to probe this user. If the UAV chooses to move even at $b = b_L$, it will always move for any other beliefs. Moreover, Case II is more likely to happen if the travel time n between users is small or the covered user's temporal correlation is strong to stay idle.

4.3 Case III: Wait to Move ($\phi_S = [b_L, b_{th})$, $\phi_M = [b_{th}, b_H]$)

Based on Lemma 7, if the UAV's optimal policy is to stay at $b = b_L$ but move at $b = b_H$, there exists a belief threshold $b_{th} \in (b_L, b_H)$ beyond which the UAV should move. In this case, we have $V_S(0, b) > V_M(0, b)$ for any $b \in [b_L, b_{th})$, and $V_S(0, b) \leq V_M(0, b)$ for all $b \in [b_{th}, b_H]$.[2] The closed-form belief threshold b_{th} can be obtained by solving

$$V_S(0, b_{th}) = V_M(0, b_{th}). \quad (39)$$

In the following proposition, we present the optimal UAV deployment policy for Case III.

Lemma 10. *If $b_H \geq \hat{b}_H$ with $\hat{b}_H$ in* (30) *and $b_L \leq \hat{b}_L$ with $\hat{b}_L$ in* (38)*, there exists a unique closed-form belief threshold b_{th} in* (40) *such that the optimal policy to follow. The UAV will stay at the idle covered user ($x = 0$) if the active belief of the uncovered user is less than b_{th} and otherwise move to the uncovered user, i.e.,*

$$\pi^*(0, b) = \begin{cases} S, \text{ if } b \in [b_L, b_{th}) & (41a) \\ M, \text{ if } b \in [b_{th}, b_H]. & (41b) \end{cases}$$

Proof. To solve (39), we need to derive the expressions of $V(0, b)$ and $V(1, b)$ for any $b \in [b_L, b_H]$. For $b \in [b_{th}, b_H]$, we have $V(0, b) = V_M(0, b)$. In this region, $V(0, b)$ and $V(1, b)$ can be similarly derived as in Case II.

Then we only need to derive $V(0, b) = V_S(0, b)$ and $V(1, b)$ for the other region of $b \in [b_L, b_{th}]$. Denote the range $\theta_k = (T^{-k}(b_{th}), T^{-(k-1)}(b_{th}))$. For $b \in \theta_k$, we have $T^{k-1}(b) \leq b_{th} \leq T^k(b)$. Since $T^i(b)$ is an increasing function of i for $b \in [b_L, b_H]$, we can deduce that $T^i(b) \leq b_{th}$ and $V(0, T^i(b)) = V_S(0, T^i(b))$ for $i \leq k-1$, and similarly $T^i(b) \geq b_{th}$ and $V(0, T^i(b)) = V_M(0, T^i(b))$ for $i \geq k$.

For any $b \in [b_L, b_{th}]$, we can use the forward induction to derive $V(0, b)$ and $V(1, b)$ as functions of $V(0, T^k(b))$ and $V(1, T^k(b))$, i.e,

$$V(0, b) = R \sum_{i=1}^{k-1} \left[\gamma^i T^i(0)\right] + \gamma^k \left[1 - T^k(0)\right] V(0, T^k(b)) + \gamma^k T^k(0) V(1, T^k(b)), \quad (42)$$

$$V(1, b) = R \left[1 + \sum_{i=1}^{k-1} \gamma^i T^i(1)\right] + \gamma^k [1 - T^k(1)] V(0, T^k(b)) + \gamma^k T^k(1) V(1, T^k(b)). \quad (43)$$

Since $T^k(b) \in [b_{th}, b_H]$, we have $\pi^*(T^k(b)) = M$ and the expressions of $V(0, T^k(b))$ and $V(1, T^k(b))$ can be obtained as functions of $V(0, b_L)$ and $V(1, b_L)$ similar to Case II. Finally, we can rewrite $V(0, b)$ in (42) and $V(1, b)$ in (43) as functions of $V(0, b_L)$ and $V(1, b_L)$. By jointly solving $V(0, b)$ and $V(1, b)$ at $b = b_L$, we cannot directly obtain the expression of $V(0, b_L)$ as we did in Case II. This is because different b may take different rounds of k to go across b_{th} by using the operator $T^k(b)$. In other words, if we start with $b \in \theta_k$, we know $T^k(b) > b_{th}$; but if we start with b_L, we do not know if $T^k(b_L)$ is greater or smaller than b_{th}. To solve this problem, we use a new index j for b_L by assuming $T^{j-1}(b_L) \leq b_{th} \leq T^j(b_L)$. Then, we solve the optimal j by one-dimensional line search. Finally, we substitute $V(0, b_L)$ and $V(1, b_L)$ into $V(0, b)$ and $V(1, b)$ and obtain the final expressions.

The threshold b_{th} can be obtained by solving (39) which is rewritten as

$$\gamma \left[(1 - p) V(0, T(b_{th})) + p V(1, T(b_{th}))\right] = \gamma^n \left[(1 - T^n(b_{th})) V(0, b_L) + T^n(b_{th}) V(1, b_L)\right]. \quad (44)$$

Since $T(b_{th}) > b_{th}$, we have $V(0, T(b_{th})) = V_M(0, T(b_{th}))$ to further simplify (44). We finally obtain b_{th} in (40) by solving (44). □

Based on Lemmas 8-10, we summarize the key results for the optimal UAV deployment policy in the following theorem.

Theorem 1. *The optimal UAV deployment policy is given by*

$$\pi^*(0, b) = \begin{cases} S, \text{ if } b_H < \hat{b}_H, b \in [b_L, b_H] & (45a) \\ S, \text{ if } b_L \leq \hat{b}_L, b_H \geq \hat{b}_H, b \in [b_L, b_{th}] & (45b) \\ M, \text{if } b_L \leq \hat{b}_L, b_H \geq \hat{b}_H, b \in [b_{th}, b_H] & (45c) \\ M, \text{ if } b_L > \hat{b}_L, b \in [b_L, b_H]. & (45d) \end{cases}$$

4.4 Empirical Studies

We now conduct some empirical studies and present numerical results for showing the optimal policy of UAV in

2. The value of b_{th} may be lower than b_L or greater than b_H in general, and (if so) we can simply replace it by $\max(b_L, \min(b_{th}, b_H))$ without hurting the optimality.

$$\hat{b}_L = b_H - \frac{[\gamma^{n-1} - (1-p)\gamma^n]\left[1 + \frac{1-\gamma^n}{b_H\gamma^n(1-(q-p)^{2n})}b_H\right] - p\left(1 + \frac{1-\gamma^n}{b_H\gamma^n(1-(q-p)^{2n})}\right)}{\frac{1-\gamma^n}{b_H\gamma^n(1-(q-p)^{2n})}\gamma^{n-1}(q-p)^n\left[1-(1-p)\gamma(q-p) + \frac{p\gamma^2(q-p)(1-q)(1-(q-p))}{1-(q-p)q_1\gamma}\right]}. \tag{38}$$

$$\begin{aligned} b_{th} = &-\frac{[b_H(1-\gamma(1-p)-(q-p)^n+(q-p)^{1+n}\gamma(1-p)) + \gamma^{1-n}p(1-q)\frac{\gamma^{n+1}(q-p)^{n+1}(b_H-p)}{1-(q-p)q\gamma}]}{[(q-p)^n-(q-p)^{1+n}\gamma(1-p) - \frac{(q-p)^{n+2}}{1-(q-p)q\gamma}\gamma^2p(1-q)]} \\ &+\frac{\frac{\gamma^{1-n}Rp}{1-q\gamma} - [1-\gamma(1-p)]V(0,b_L) + \gamma^{1-n}p(1-q)\frac{\gamma^{n+1}}{1-q\gamma}\left(b_HV(1,b_L) + (1-b_H)V(0,b_L)\right)}{[V(1,b_L)-V(0,b_L)][(q-p)^n-(q-p)^{1+n}\gamma(1-p) - \frac{(q-p)^{n+2}}{1-(q-p)q\gamma}\gamma^2p(1-q)]}. \end{aligned} \tag{40}$$

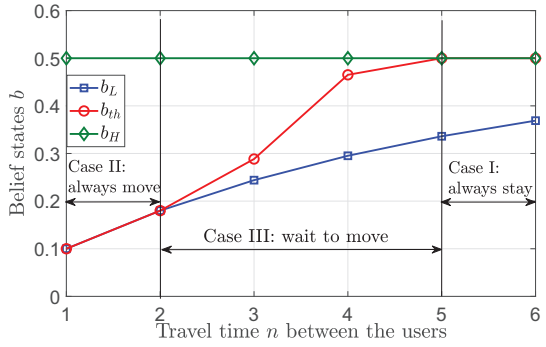

Fig. 4. The effect of travel time n between the users on the optimal policy's belief threshold b_{th} in the three cases.

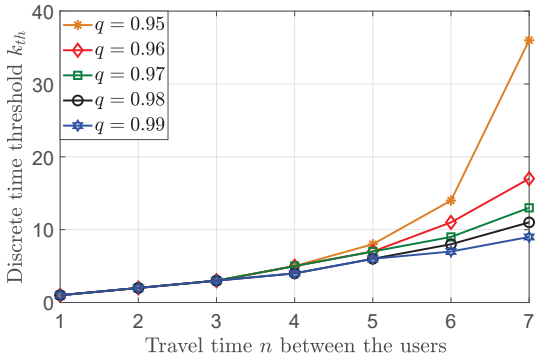

Fig. 5. Effects of travel time n and symmetric self-transition probability $q = 1-p$ on the threshold k_{th} in (46).

the three cases in Sections 4.1-4.3. We normalize the service reward as $R = 1$ and set the discount factor as $\gamma = 0.9$. We set $q = 1-p = 0.9$ for symmetric state-transition probabilities in the Markov chain in Fig. 1. In Fig. 4, we show the effect of travel time n between the two users on the optimal policy in the three cases. As the delay cost n for serving the other user increases, the UAV's optimal policy changes from "always move" in Case II to "wait-to-move" in Case III and finally to "always stay" in Case I. When $n \leq 2$, once the covered user is idle, the UAV will move to the uncovered user that has higher chance in active state without worrying the small delay. For the other extreme case of $n \geq 5$, b_{th} overlaps with b_H and the delay cost to reach the uncovered user is high, motivating the UAV to stay with the covered user. In the medium n regime, the threshold b_{th} increases with the travel time, as the UAV is more reluctant to moves across a greater n between users.

The optimal policy in (41a) (or (41b)) requires to learn and update belief b about the uncovered user. To further simplify this, the UAV only needs to count a threshold number k_{th} of time slots for waiting at the idle covered before moving, by solving the equation of

$$b_{th} = T^{k_{th}}(0) \tag{46}$$

and (if the solution k_{th} is decimal) rounding it as the greater integer. Then the UAV's optimal strategy is equivalently

$$\pi^*(0,k) = \begin{cases} S, \text{ if } k \in [k_L, k_{th}) & \text{(47a)} \\ M, \text{ if } k \in [k_{th}, k_H]. & \text{(47b)} \end{cases}$$

If the covered user keeps idle for k_{th} number of consecutive slots, the UAV will move to the uncovered user.

Remark 2. *Note that our threshold-based policy still holds by considering limited energy storage or operation time of the UAV. We can extend our POMDP problem in Section 2.2 to a finite time horizon, and show that where the decision threshold b_{th} (though no longer in closed-form) increases with the operation time and the UAV is less likely to move given less remaining time/energy is left in the storage. Another way to interpret the effect of limited operation time in our current results is through the discount factor γ in (7) of our infinite time horizon model. The smaller γ tells that the UAV cares less about the future time slots or there are less operation time/energy left. We can show that the threshold b_{th} in (40) increases (i.e., the UAV is more likely to stay) as γ decreases.*

In Fig. 5, we further examine the effects of the travel time n and the self-transition probability $q = 1-p$ in Fig. 1 on the discrete waiting time threshold k_{th} (as the solution to (46)) for Case III (wait-to-move). Similar to the belief threshold b_{th} in Fig. 4, here we see that k_{th} also increases with n since the UAV is less likely to move as the longer travel time n negates the moving reward. Moreover, we see that k_{th} decreases with the increase of the self-transition probability q. For large $q = 1-p$ close to 1, the covered user is very likely to be idle in the next slot given it is idle now, and the UAV has little chance at the covered user and it would rather explore the active opportunity at the uncovered user by waiting a shorter time k_{th}.

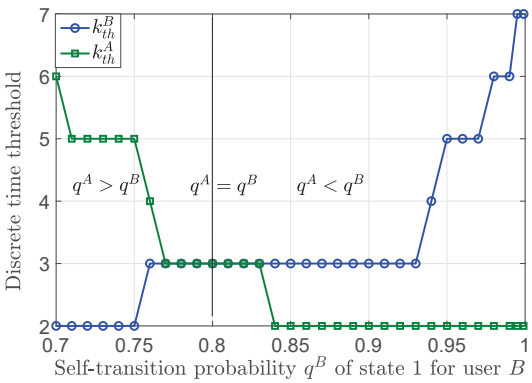

Fig. 6. The effect of asymmetry on the time thresholds for non-identically distributed two-user scenario ($\gamma = 0.9$, $\epsilon = 0.5$, $R = 1$, $q^A = 0.8$, $p^A = 0.2$, $p^B = 1 - q^B$, $n = 2$, $K = 12$).

4.5 Extension to Asymmetric User Activity Patterns

In this subsection, we consider a more general case of non-identical Markov models for the two users in the POMDP modelling and analysis. We denote the states of user A and user B as s^A and s^B, respectively, which are modeled as two independent two-state Markov chains. The transition probabilities in Fig. 1 are extended as p^A and q^A for user A, and p^B and q^B for user B, where q^A and q^B denote the self-transition probabilities of state 1 for user A and user B, and p^A and p^B denote the transition probabilities from state 0 to state 1 for user A and user B, respectively. We also have $q^A > 1/2 > p^A$ and $q^B > 1/2 > p^B$ to ensure the positive correlation of the data traffic per user.

Due to user heterogeneity, we generally denote the index of the covered user by $\bar{z} \in \{A, B\}$ and that of the uncovered user by $z \in \{A, B\} \backslash \bar{z}$. The state of the system is denoted as $(\bar{z}, x, b)$. By including $\bar{z}$ as part of the state, now the value functions in (11), (12), (14) and (15) are extended to eight functions $V_a(\bar{z}, x, b)$. We can similarly prove the structure of the optimal policy follows a threshold-type, where Lemmas 1-7 still hold. Different from the identical-user case, we now have two feasible regions for the beliefs on the two users, i.e., $[b_L^z, b_H^z]$ for user $z \in \{A, B\}$. Given the covered user $\bar{z}$ is idle, one can obtain the optimal deployment policy similar to Propositions 1-3 by deriving the belief thresholds of the uncovered user z. Similar to (30), (38) and (40), we can obtain $\hat{b}_H^z$, $\hat{b}_L^z$, and b_{th}^z for $z \in \{A, B\}$. If $b_H^z < \hat{b}_H^z$, the optimal policy is $\pi^*(\bar{z}, 0, b) = S$ for any $b \in [b_L^z, b_H^z]$. If $b_L^z > \hat{b}_L^z$, the optimal policy is $\pi^*(\bar{z}, 0, b) = M$ for any $b \in [b_L^z, b_H^z]$. Otherwise, the optimal policy is $\pi^*(\bar{z}, 0, b) = S$ for $b \in [b_L^z, b_{th}^z)$, and $\pi^*(\bar{z}, 0, b) = M$ for $b \in [b_{th}^z, b_H^z]$. Similarly, the belief threshold can be translated to two discrete time thresholds k_{th}^z for $z \in \{A, B\}$ (similar to (46) in Section 4.3), where we denote k_{th}^z as the waiting time threshold to stay with the covered user $\bar{z}$ before moving to the uncovered user z. The UAV should spend a threshold number of slots on the idle covered user before it moves to the uncovered user on the other side. The difference here is that we have two waiting time thresholds due to the non-identical property of the two users.

In Fig. 6, we numerically study the time threshold for the non-identical two-user scenario. For the ease of exposition, we still set symmetric Markov chain for each user, i.e., $q^A = 1 - p^A$ and $q^B = 1 - p^B$. For the special case of two symmetric users with $q^A = q^B = 0.8$, we see from Fig. 6 that $k_{th}^A = k_{th}^B = 3$. As we unilaterally increase the self-transition probability q^B of state 1 for user B by keeping $q^A = 0.8$, we equivalently decrease the probability for user B to recover from idle to active state. We see k_{th}^B increases with q^B (user B's probability to keep idle): given the UAV now covers idle user A, it prefers to patiently stay longer with user A to wait until user B recovers to the active state. We also see from Fig. 6 that k_{th}^A decreases with q^B: given the UAV now covers idle user B, if user B will not likely to return to the active state, the UAV prefers to depart earlier to explore user A. To sum up, we can see that the UAV visits the user more often if it has a higher probability to turn active.

5 Dynamic UAV Deployment via Refined Reinforcement Learning

In this section, we consider a more challenging scenario where the UAV does not even know the system parameters of the users' time-varying user activities in the POMDP problem including the state-transition probabilities. We will analyze the adaptive UAV deployment policy in such environment refining prior design of Q-learning algorithm in [39].[3] In the following, we will first present the new system state and then discuss how to obtain the efficient Q-learning policy by updating the Q-table. Later we will extend to multi-user scenario.

5.1 Refined Reinforcement Learning for Two-User System

We first design the UAV deployment via Q-learning for the two-user system in Fig. 1. More generally, here the two users' activity probability distributions are not necessarily the same and they can follow two different Markov chains independently. As the UAV still observes the state of the covered user, it just needs to learn from the other (uncovered) user. The difficulty here is that the UAV can no longer learn the active belief b due to the unknown state-transition probabilities. By exploiting the memoryless property in the Markov chains, we propose that the UAV just learns the state of the other (uncovered) user based on its memory of the last visit on the uncovered user and the time elapsed since the last visit. For each time episode, we can characterize the system state of the two users by a vector of $\nu = [\bar{z}\ x\ u\ k]$, where $\bar{z} \in \{A, B\}$ represents which user (A or B) is currently covered by the UAV as shown in Fig. 1, $x \in \{0, 1\}$ is the idle/active state of the covered user, $u \in \{0, 1\}$ is the last observation (idle/active state) of the uncovered user, and k is the number of time slots elapsed since the last visit. As it takes n time slots to travel back from the other user, $k \geq n$ holds. Yet k may go to infinity in the case that the UAV always stays with the currently covered user, resulting in formidably high complexity for us to learn and update the Q-table for infinitely many possibilities of ν. To reduce the complexity of Q-learning, we truncate the value of k once beyond the allowed buffer size of K, i.e., $k = \min(k, K)$. Then our state size is only $8K$ and note

3. Besides Q-learning, some other RL algorithms like Sarsa takes longer time to converge.

that the convergence time of Q-learning grows greatly with the state size. One may imagine that this truncation to K may cause efficiency loss in learning. Later in Fig. 7, we will analyze the effect of K and show the learning efficiency loss is mild even for a small K value (e.g., $K = 12$).

For each system state ν, we output one of two actions in Q-learning policy: stay ($a = S$) or move ($a = M$). If action $a = S$ is chosen, the UAV receives the reward of $r_S(\nu)$ in the current slot and will update a new state of $\nu' = [\bar{z}'\ x'\ u'\ k']$ in the next slot, where we can deduce that $\bar{z}' \leftarrow \bar{z}$ for staying at the same user, $u' \leftarrow u$, $k' \leftarrow \min(k+1, K)$, and x' is the newly observed state at the existing covered user. If action $a = M$ is chosen and the UAV will not make any observation for $n-1$ time slots during travelling in midway, the UAV receives zero reward of $r_M(\nu)$ in the current slot and will update the state of $\nu'' = [\bar{z}''\ x''\ u''\ k'']$ only after n slot. Then we have $\bar{z}'' \leftarrow \{A, B\}\backslash\bar{z}$ for changing the covered user's identity, $u'' \leftarrow x$, $k'' \leftarrow n$, and x'' is the observed state of the newly covered user. The new states of the covered user x' and x'' above as well as those at the uncovered user are realized by the system with hidden state-transition probabilities from the UAV. We then gradually establish a Q-table over time to help the UAV learn and decide which action a to take for each observed state ν.

Given each state-action pair, there is a unique Q-function $Q_a(\nu)$ that quantifies the expected reward by taking action $a \in \{S, M\}$ at state ν. Here $Q_a(\nu)$ is used instead of the value function of $V_a(\bar{z}, x, b)$ discussed in Section 4.4. Since the UAV does not know the system parameters about time-varying users' activities, it is not able to explicitly derive $V_a(\bar{z}, x, b)$. The size of Q-table is $16K$ for the two possible actions as we cut down the state size to $8K$. Our basic idea of refined Q-learning design to iteratively update and improve the Q-table by employing the temporal difference between the predicted and existing Q-values. Formally, we present the refined Q-learning algorithm in Algorithm 1.

In Algorithm 1, we adopt ϵ-greedy method to balance between exploitation and exploration by choosing actions, where the UAV takes the currently optimal action suggested by the Q-table with probability $1-\epsilon$ and selects a random action with probability ϵ. We also adapt a dynamic learning rate η to adjust the effect of the new information on the existing Q-value. We dynamically choose the learning rate according to current state-action tuple (ν, a) for improving learning efficiency over time. As suggested by [40], we choose $\eta = \frac{1}{\sqrt{1+N(\nu,a)}}$. The convergence of this algorithm can be ensured once each state-action tuple in the Q-table has been visited for enough times [39]. Taking Fig. 7 as an example, Algorithm 1 takes around 10 seconds to converge and we can further reduce the complexity by setting a smaller buffer size K. Besides Q-learning, some other RL algorithms like Sarsa takes longer time to converge. In practice, one can also train the Q-table offline using sufficient number of users' activity data, and then further modify the policy via the online iterations.

In Fig. 7, we compare between the expected total discounted rewards achieved by the optimal POMDP policy in Section 4 (as the performance upper-bound) and model-free Q-learning based Algorithm 1. Both of them are increasing in the initial active belief on the uncovered user, which is consistent with Lemma 3. As Algorithm 1 does not know or use the system parameters, inevitably it has efficiency loss. Further, we truncate the memory buffer size to finite K for keeping a finite Q-table and this also results in possible efficiency loss. We see from Fig. 7 that the performance of Q-learning improves as K increases since the UAV's belief update is more accurate based on previous visits. In other words, a higher K value allows for a better estimation of the belief on the uncovered user. Still, we find that even if K is small (e.g., $K = 12$), our refined Q-learning algorithm approaches well to the optimal policy, and we manage to only mildly sacrifice efficiency for saving huge complexity.

We present the convergence performance of the proposed algorithm for various buffer size of K in Fig. 8. The simulation platform is set up via MATLAB 2019b on a desktop with 2.60 GHz Intel core, 16 GB RAM, and Win-

Algorithm 1 Refined Q-learning Algorithm for UAV Deployment

Initialize Q-table, learning rate η and discount factor γ.
Observe current state $\nu = [\bar{z}\ x\ u\ k]$.
repeat
  Choose $a \in \{S, M\}$ to yield $\max\{Q_S(\nu), Q_M(\nu)\}$ with probability ϵ and a random action with $1-\epsilon$.
  Update learning rate $\eta = \frac{1}{\sqrt{1+N(\nu,a)}}$, where $N(\nu, a)$ is the number of times to observe (ν, a) tuple till now.
  if $a = S$ **then**
    Observe current reward $r_S(\nu)$ and existing covered user's state x'.
    Update system state $\nu' = [\bar{z}\ x'\ u\ \min(k+1, K)]$.
    Update the Q-value $Q_S(\nu) \leftarrow Q_S(\nu) + \eta[r_S(\nu) + \gamma \max_{a'} Q_{a'}(\nu') - Q_S(\nu)]$ in the Q-table.
    Replace $\nu \leftarrow \nu'$.
  else
    Observe current reward $r_M(\nu)$ and newly covered user's state x''.
    Update system state $\nu'' = [\{A, B\}\backslash\bar{z}\ x''\ x\ n]$.
    Update Q-value $Q_M(\nu) \leftarrow Q_M(\nu) + \eta[r_M(\nu) + \gamma^n \max_{a''} Q_{a''}(\nu'') - Q_M(\nu)]$ in the Q-table.
    Replace $\nu \leftarrow \nu''$.
  end if
until end

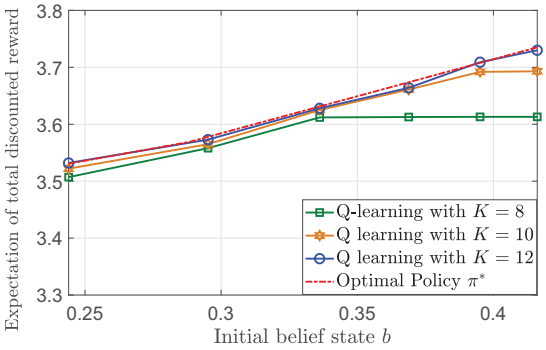

Fig. 7. Comparison between the expectations of the total discounted reward achieved by our optimal policy in Section 4 and our Q-learning policy returned by Algorithm 1 versus initial active belief b of the uncovered user ($n = 3$, $\epsilon = 0.5$, $R = 1$, $\gamma = 0.9$, $q = 0.9$, $p = 0.1$).

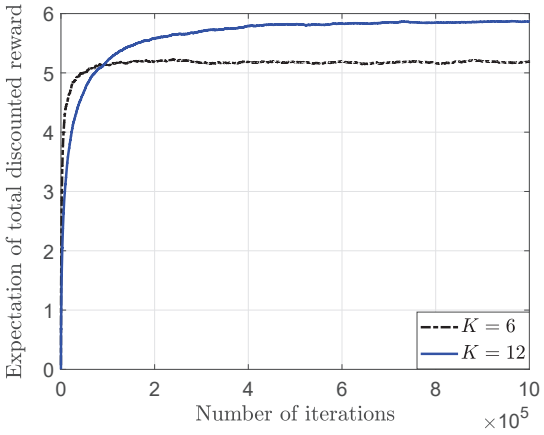

Fig. 8. Convergence performance of the proposed algorithm under different buffer size K ($n=2$, $R=1$, $\gamma=0.9$, $q=0.9$, $p=1-q$).

dows x86 professor. First, we observe that the expectation of total discounted reward converges after the 4×10^5 iterations (around 8 s) and 8×10^5 iterations (around 22 s) for $K=8$ and $K=12$, respectively. It shows that the convergence speed is slower for larger buffer size K, which is due to the higher complexity in searching through a larger size of Q-table. Moreover, the average reward increase with K since the UAV has more accurate information about the environment. According to Figs. 7 and 8, we see that there is a tradeoff between the optimality and complexity, where adopting a greater K value improves the average reward at the cost of slower convergence. The training is processed offline. By observing the system states/beliefs, the UAV find the corresponding policy in the converged Q-table for online deployment decisions.

5.2 Extension to Multi-User Service

In this subsection, we extend Algorithm 1 to multi-user cases. We will show that the threshold-based policy also holds for the multi-user scenario.

5.2.1 Line Topology

Without much loss of generality, we first consider a typical structure of three users $A-B-C$ who are connected one by one in a linear line with the middle user B and two side users A and C, where user B is located at a distance of n flying time slots from either user A and user C, respectively. Compared with the two-user case, there are mainly two differences for the refined Q-learning algorithm design. First, when defining the system state ν, we have two instead of one uncovered users now (e.g., users B and C if the UAV covers user A now), and need to update the elapsed time steps k^B and k^C since last visits on them. Similarly, when defining the Q-table, we truncate both k^B and k^C if beyond K to keep the Q-table size finite. Second, the UAV now has more than two choices if hovering above user B. It may move to user A, stay at middle user B, or move to user C. Accordingly, we add more actions to the Q-table and the rest of Algorithm 1 remains the same.

Here we use extensive simulations to show that the dynamic deployment policy still follows a threshold-based

TABLE 2
Waiting time thresholds for three-user and two-user cases given the UAV is currently above idle user A ($n=2$, $\gamma=0.9$, $p=1-q$, $K=12$).

Self-transition probability q	0.78	0.79	0.8	0.81
$\{k_{th}^B, k_{th}^C\}$ in 3-user case	$\{3,9\}$	$\{3,5\}$	$\{2,8\}$	$\{2,4\}$
k_{th} in 2-user case	4	4	3	3

TABLE 3
Waiting time thresholds for three-user and two-user cases given the UAV is currently above idle user B ($n=2$, $\gamma=0.9$, $p=1-q$, $K=12$).

Self-transition probability	0.78	0.79	0.8	0.81
k_{th}^C in 3-user case	6	6	6	6
k_{th} in 2-user case	4	4	3	3

structure under various self-transition probability q values. For illustration purpose, we consider i.i.d. Markov chains at the three users with $p=1-q$. In the second row of Table 2, we consider the UAV is now above idle user A, and both users B and C are on its right-hand side. The UAV deployment policy depends on not only the consecutive unvisited time slots k^B of user B but also the unvisited time slots k^C of user C. For any given q, we show that it will move to user B if both k^B and k^C exceed the corresponding thresholds of k_{th}^B and k_{th}^C at the same time, respectively. We compare the results with a previous case of two users $A-B$ by simply removing user C, where the optimal policy has a single threshold of k_{th} as shown in the third row of Table 2. Given the UAV is above the idle user A, our first observation is that both the thresholds k_{th}^B (for three-user case) and k_{th} (for two user case) decrease with q, as user A is less likely to recover from idle to active. Moreover, we have $k_{th}^B < k_{th}$, which means the UAV in the three-user case is more willing to move to the middle user B to explore the potential opportunity at both users B and C. Furthermore, given the same k_{th}^B value (e.g., $k_{th}^B=3$), the threshold k_{th}^C also decreases with q (e.g., $k_{th}^C=9$ for $q=0.78$ versus $k_{th}^C=5$ for $q=0.79$). This is because user C (since last UAV visit) returns from idle to active faster as q increases.

Next, we consider the case that the UAV is now above the idle user B in the middle, and suppose user C was visited a longer time ago than user A. We can show that the UAV (if moves) prefers to move to the user C that was visited a longer time ago. We can also see that k_{th}^C in Table 3 is greater than k_{th} and k_{th}^B in Table 2, telling that the UAV is less likely to move to the two sides, and is more likely to move from sides to the center.

5.2.2 Ring Topology

Consider a multi-user ring network, where the mutual distance between every pair of neighbouring users is n. Assume the UAV can only cover at most one user at a time. Without loss of generality, we consider the UAV may either stay above the covered user to exploit the local demand, or fly clockwise/counter-clockwise to the nearest uncovered user to explore new opportunities. By applying similar Q-learning algorithm to this network, the results show that the optimal deployment policy is still a threshold-based type. Denote that the number of time slots that the UAV has been

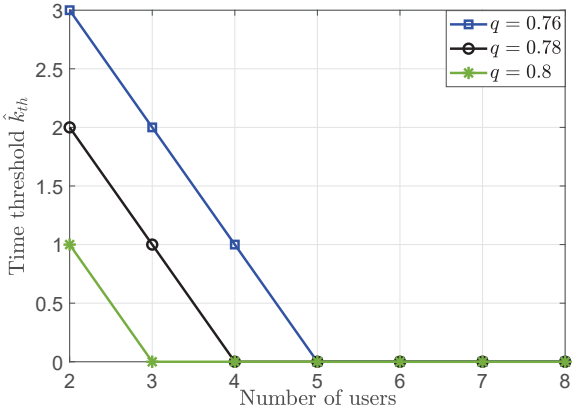

Fig. 9. The effect of user number on the waiting time threshold ($n = 2$, $K = 12$, $R = 1$, $\gamma = 0.9$, $p = 1 - q$).

waiting above an idle user by $\hat{k}$. Given the covered user turns idle, the optimal policy for the UAV is to stay with the covered user if $\hat{k} < \hat{k}_{th}$ and move to another user if $\hat{k} \geq \hat{k}_{th}$. We observe that the UAV prefers visiting the neighbour user that was visited a longer time ago since this user has higher active probability than the other neighbour. Thus, the UAV's moving trajectory follows a round robin pattern among the users. Whether the trajectory follows clockwise or counter-clockwise depends on the initial flying direction. In Fig. 9, we see that time threshold $\hat{k}_{th}$ increases with user number, where the UAV is more likely to move if there are more users to explore. As the user number goes large (e.g., greater than 5 users), we see that the time threshold $\hat{k}_{th}$ approaches zero $\hat{k}_{th} = 0$, where the UAV moves immediately to another user once the covered user is idle. Furthermore, to alleviate the curse of dimensions of Q-table for large user number and user distance, one can adopt deep learning techniques to approximate the Q-values. Intuitively, the time threshold $\hat{k}_{th}$ increases with the user distance n since it is more costly to explore.

6 Conclusion

This paper presented a novel UAV deployment to learn and chase the time-varying activities of the ground users at diverse locations. There are mainly two challenges: first, the UAV only has local observations about the user activities due to the limited coverage; second, the users' demands may have changed when the UAV arrives at the scene due to the delay caused by the limited flying speed. We formulated the problem as a POMDP to address both the temporal correlation and partial observability about the user activities, where the UAV can update the belief about the active probability of the uncovered users based on the visiting history. Given the covered users is idle, there exists a fundamental delay-reward tradeoff in the deployment process: the UAV may either chase a higher but delayed reward at the uncovered user, or wait for a smaller reward at the covered user. We proved the optimal deployment policy follows a threshold-based type and derived the thresholds in closed-form. The results showed that the UAV would stay with an idle user for a threshold number of time slots before moving to the uncovered user, where the threshold can be zero (always move), a positive value (wait-before-chase), or infinity (always wait) depending on the system parameters. We also showed that the UAV has greater incentive to move if the distance to fly between users is shorter or the temporal correlation of each user's idling pattern is stronger. We extended the discussions to the two-user scenario with non-identical Markov chains and showed that there exists two waiting time thresholds for the two asymmetric users, where the UAV is more willing to visit the user that has a higher recovery probability from the idle state to active state. Furthermore, we extended to a more challenging scenario without knowing each user's temporal activity distribution parameters, and we applied Q-learning to develop the efficient deployment policy which also suggests a threshold-type. Finally, we extended the threshold-based policy to a multi-user scenario. Apart from the time-varying user demand, we will further investigate the effect of spatial mobility on the UAV deployment policy in the future work.

References

[1] S. Hayat, E. Yanmaz, and R. Muzaffar, "Survey on unmanned aerial vehicle networks for civil applications: A communications viewpoint," *IEEE Commun. Surveys & Tut.*, vol. 18, no. 4, pp. 2624-2660, Fourth Quarter, 2016.

[2] Y. Zeng, Q. Wu, and R. Zhang, "Accessing from the sky: A tutorial on UAV communications for 5G and beyond,"*Proc. IEEE*, vol. 107, no. 12, pp. 2327-2375, Dec. 2019.

[3] F. Jiang and A. L. Swindlehurst, "Optimization of UAV heading for the ground-to-air uplink," *IEEE J. Sel. Areas Commun.*, vol. 30, no. 5, pp. 993-1005, Jun. 2012.

[4] A. Al-Hourani, S. Kandeepan, and S. Lardner, "Optimal LAP Altitude for Maximum Coverage," *IEEE Wireless Commun. Lett.*, vol. 3, no. 6, pp. 569-572, Dec. 2014.

[5] C. Zhang and W. Zhang, "Spectrum sharing for drone networks," *IEEE J. Sel. areas Commun.*, vol. 35, no. 1, pp. 136-144, Jan. 2017.

[6] Y. L. Che, S. Luo, and K. Wu, "Spectrum sharing based cognitive UAV networks via optimal beamwidth allocation," in Proc. *IEEE Int. Conf. Commun. (ICC)*, Shanghai, China, May 20-24, 2019.

[7] M. Chen, M. Mozaffari, W. Saad, C. Yin, M. Debbah, and C. S. Hong, "Caching in the sky: Proactive deployment of cache-enabled unmanned aerial vehicles for optimized quality-of-experience," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 5, pp. 1046C1061, May 2017.

[8] J. Gong, T. H. Chang, C. Shen, and X. Chen, "Flight time minimization of UAV for data collection over wireless sensor networks," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 9, pp. 1942-1954, Sep. 2018.

[9] S. Jeong, O. Simeone, and J. Kang, "Mobile edge computing via a UAV-mounted cloudlet: Optimization of bit allocation and path planning," *IEEE Trans. Veh. Technol.*, vol. 67, no. 3, pp. 2049-2063, Mar. 2018.

[10] H. Wang, J. Wang, G. Ding, J. Chen, Y. Li, and Z. Han, "Spectrum sharing planning for full-duplex UAV relaying systems with underlaid D2D communications," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 9, pp. 1986-1999, Sep. 2018.

[11] J. Chen, U. Yatnalli, and D. Gesbert, "Learning radio maps for UAV-aided wireless networks: A segmented regression approach," in Proc. *IEEE Int. Conf. Commun. (ICC)*, Paris, France, May 21-25, 2017.

[12] V. V. Chetlur and H. S. Dhillon, "Downlink coverage analysis for a finite 3-D wireless network of unmanned aerial vehicles," *IEEE Trans. Commun.*, vol. 65, no. 10, pp. 4543-4558, Oct. 2017.

[13] X. Wang and L. Duan, "Dynamic pricing and capacity allocation of UAV-provided mobile services," in *Proc. IEEE Int. Conf. Comput. Commun. (INFOCOM)*, Paris, France, May 2019.

[14] Z. Wang, L. Duan, and R. Zhang, "Adaptive deployment for UAV-aided communication networks," *IEEE Trans. Wireless Commun.*, vol. 18, no. 9, pp. 4531-4543, Jul. 2019.

[15] X. Zhang and L. Duan, "Fast deployment of UAV networks for optimal wireless coverage," *IEEE Trans. Mobile Comput*., vol. 18, no. 3, pp. 588-601, Mar. 2019.

[16] Q. Zhang, M. Jiang, Z. Feng, W. Li, W. Zhang, and M. Pan, "IoT enabled UAV: Network architecture and routing algorithm," *IEEE Internet Things J.,* vol. 6 , no. 2 , Apr. 2019.

[17] L. Bertizzolo, S. D'Oro, L. Ferranti, L. Bonati, and E. Demirors, "SwarmControl: An automated distributed control framework for self-optimizing drone networks," in *Proc. IEEE Int. Conf. Comput. Commun. (INFOCOM),* Toronto, Canada, May 2020.

[18] Q. Liu, L. Shi, L. Sun, J. Li, M. Ding, and F. Shu, "Path planning for UAV-mounted mobile edge computing with deep reinforcement learning," *IEEE Trans. Veh. Technol.,* vol. 69, no. 5, pp. 5723-5728, May 2020.

[19] H. Bayerlein, M. Theile, M. Caccamo, and D. Gesbert, "UAV path planning for wireless data harvesting: A deep reinforcement learning approach," in *Proc. IEEE Global Commun. Conf. (Globecom),* Taipei, Taiwan, Dec. 8-10, 2020.

[20] R. Ding, F. Gao, and X. S. Shen, "3D UAV trajectory design and frequency band allocation for energy-efficient and fair communication," *IEEE Trans. Wireless Commun.,* vol. 19, no. 12, pp. 7796-7809, Dec. 2020.

[21] X. Zhong, Y. Huo, X. Dong, and Z. Liang, "Deep Q-network based dynamic movement strategy in a UAV-assisted network," in *Proc. IEEE 92nd Vehicular Technology Conference (VTC2020-Fall)*, Nov. 18-Dec. 16 2020.

[22] S. Yin, S. Zhao, Y. Zhao, and F. R. Yu, "Intelligent trajectory design in UAV-aided communications with reinforcement learning," *IEEE Trans. Veh. Technol.,* vol. 68, no. 8, pp. 8227-8231, Jun. 2019.

[23] Y. Zhang, Z. Mou, F. Gao, J. Jiang, R. Ding, and Z. Han, "UAV-Enabled secure communications by multi-agent deep reinforcement learning," *IEEE Trans. Veh. Technol.,* vol. 69, no. 10, pp. 11599-11611, Aug. 2020.

[24] J. Hu, H. Zhang, L. Song, Z. Han, and H. V. Poor, "Reinforcement learning for a cellular Internet of UAVs," *IEEE Wireless Commun.,* vol. 27, no. 1, pp. 116-123, Feb. 2020.

[25] H. Huang, Y. Yang, H. Wang, Z. Ding, H. Sari, and F. Adachi, "Deep reinforcement learning for UAV navigation through massive MIMO technique," *IEEE Trans. Veh. Technol.,* vol. 69, no. 1, pp. 1117-1121, Jun. 2020.

[26] N. Zhao, Y. Cheng, Y. Pei, Y.-C. Liang, and D. Niyato, "Deep reinforcement learning for trajectory design and power allocation in UAV networks," in *Proc. IEEE Int. Conf. Commun. (ICC),* Jun. 7-11, 2020.

[27] C. H. Liu, Z. Chen, J. Tang, J. Xu, and C. Piao, "Energy-efficient UAV control for effective and fair communication coverage: A deep reinforcement learning approach," *IEEE J. Sel. Area. Commun.,* vol. 36, no. 9, pp. 2059-2070, Sep. 2018.

[28] J. Qiu, J. Lyu, and L. Fu, "Placement optimization of aerial base stations with deep reinforcement learning," in *Proc. IEEE Int. Conf. Commun. (ICC),* Jun. 7-11, 2020.

[29] Y. Zeng, X. Xu, S. Jin, and R. Zhang, "Simultaneous navigation and radio mapping for cellular-connected UAV with deep reinforcement learning," *IEEE Trans. Wireless Commun.,* to appear, 2021.

[30] M. Samir, C. Assi, S. Sharafeddine, D. Ebrahimi, and A. Ghrayeb, "Age of information aware trajectory planning of UAVs in intelligent transportation systems: A deep learning approach," *IEEE Trans. Veh. Technol.,* vol. 69, no. 11, pp. 12382-12395, Nov. 2020.

[31] L. Wang, K. Wang, C. Pan, W. Xu, N. Aslam, and A. Nallanathan, "Deep reinforcement learning based dynamic trajectory control for UAV-assisted mobile edge computing," *IEEE Trans. Mobile Comput.,* to appear, 2021.

[32] N. Ansari, H. Liu, Y. Q. Shi, and H. Zhao, "On modeling MPEG video traffics," *IEEE Trans. Broadcasting,* vol. 48, no. 4, pp. 337-347, Dec. 2002.

[33] M. Johnston, I. Keslassy, and E. Modiano, "Channel probing in opportunisitc Communication Systems," *IEEE Trans. Inf. Theory,* vol. 63, no. 11, pp. 7535-7552, Nov. 2017.

[34] A. Laourine and L. Tong, "Betting on Gilbert-Elliot channels," *IEEE Trans. Wireless Commun.,* vol. 9, no. 2, pp. 723-733, Feb. 2010.

[35] Q. Zhao, L. Tong, A. Swami, and Y. Chen, "Decentralized cognitive MAC for opportunistic spectrum access in ad hoc networks: A POMDP framework," *IEEE J. Sel. Areas Commun.,* vol. 25, no. 3, pp. 589-600, Apr. 2007.

[36] Y. Li, C. Courcoubetis, and L. Duan, "Recommending paths: Follow or not follow," in *Proc. IEEE Int. Conf. Comput. Commun. (INFOCOM),* Paris, France, May 2019.

[37] D. W. Matolak, and R. Sun, "Air-ground channel characterization for unmanned aircraft systems-part III: The suburban and near-urban environments," *IEEE Trans. Veh. Technol.,* vol. 66, no. 8, 6607-6618, Aug. 2017.

[38] S. M. Ross, *Applied Probability Models with Optimization Applications,* San Francisco: Holden-Day, 1970.

[39] C. J. C. H. Watkins and P. Dayan, "Q learning," *Machine Learning,* vol. 8, no.3-4, pp. 279-292, 1992.

[40] E. Even-Dar and Y. Mansour, "Learning rates for q-learning," *Journal of Machine Learning Research,* vol. 5, pp. 1-25, Dec. 2003.