2023 62nd IEEE Conference on Decision and Control (CDC)
December 13-15, 2023. Marina Bay Sands, Singapore

Optimal Dynamic Trajectories for UAVs in Mobility-Enabled Relay Systems

Winston Hurst and Yasamin Mostofi

***Abstract*— We consider a UAV which acts as a mobile relay and must plan a trajectory to enable data transfer between multiple pairs of communication nodes. For each pair, successful data transfer is only possible in non-convex regions (relay regions) where a given quality of service requirement may be satisfied for both nodes. Trajectories consist of 1) locations where the UAV stops to relay (relay positions) and 2) a dynamic transition policy which determines the sequence in which the pairs are serviced. We are interested in minimizing the average time a bit waits at a source before being sent to a destination and first pose a general, non-convex problem that calls for optimization over both the relay positions and the dynamic transition policy. To find approximate solutions, we formulate an average cost semi-Markov decision process and propose a deep-reinforcement-learning-based algorithm to solve it. To validate our approach, we present the results of several simulation experiments, which show our approach significantly outperforms the state-of-the-art.**

I. Introduction

The extraordinary development of robotics seen in recent years has allowed for new paradigms in communication system design. By data muling, relaying, or beam forming, unmanned vehicles can create new communication links or enhance existing networks. For efficient operation, the communication and motion aspects of these systems require careful joint planning, as studied in the field of *communication-aware robotics*. See [1] for a recent review of this area.

This paper considers the operation of a UAV tasked with servicing disparate communication links between several source-destination pairs, as shown in Fig. 1, which models scenarios such as the deployment of a UAV after a natural disaster has taken permanent, terrestrial infrastructure offline. At each destination, data arrives stochastically at a known average rate and must be sent to the corresponding destination, but the distance between each source and destination is too large for direct communication. The UAV's objective is to minimize the time the data waits at a source before being sent to the corresponding destination.

Problems related to autonomous robots facilitating communication systems that include spatially dispersed nodes have received attention across a variety of fields [2]–[6]. Closely related are persistent monitoring problems, in which an unmanned vehicle senses various locations in a workspace and transfers the sensed data to a remote station [7], and communication-aware variations of the vehicle routing problem (VRP) [8], [9]. However, much of existing work either uses a simplified model of the communication channel or has the robot visit sites directly, forgoing any channel modeling. Furthermore, many existing approaches plan static trajectories which do not consider the current state of the system or assume waypoints are provided *a priori*.

Winston Hurst and Yasamin Mostofi are with the Department of Electrical and Computer Engineering, University of California, Santa Barbara, USA (email: {winstonhurst, ymostofi}@ece.ucsb.edu). This work was supported in part by NSF RI award 2008449.

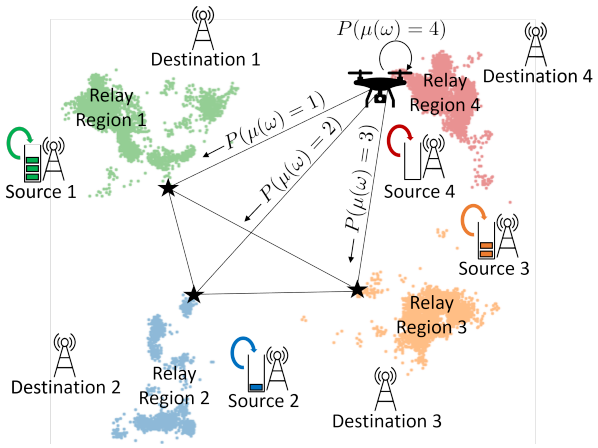

Fig. 1: At each source, bits accumulate in a queue and must be offloaded to the corresponding destination via relay. The data transfer is performed by a UAV, which moves to an optimum position within each relay region to service the corresponding queue and visits the different regions in an optimal sequence, dependent on stochastic arrival process of the data. See the color PDF for better viewing.

In this paper, we use recent advancements in reinforcement learning to find *dynamic* trajectories that account for realistic communication channels, the non-homogeneous data accumulation rate, and the spatial distribution of the pairs. As seen in Fig. 1, for each source-destination pair, there is a non-convex and disjoint region, labeled *Relay Region*, where the link qualities are good enough to allow the UAV to successfully relay data from the source to the corresponding destination. Trajectories then consist of a set of relay positions drawn from the relay regions (one per region), and a dynamic transition policy that determines which relay position to visit next based on the current system state. This planning is to be accomplished without full knowledge of the relay regions (only sparse prior channel samples available) and possibly with limited observability of the system's state.

We next explicitly present our contributions.

1) As the problem of interest is highly non-convex and intractable, we show how to decouple optimization of the relay positions and *dynamic* transition policies (in contrast to the static transition policies found in [5]). We show the problem of finding optimal persistent, dynamic transition policies given fixed relay positions can be formulated as an average cost semi-Markov decision process (SMDP), and to find relay positions, we pose a mixed-integer second-order cone program (MISCOP).
2) To find near-optimal policies, we propose a deep reinforcement learning (DRL) based algorithm which extends

recent work from [10] to handle SMDP's and incorporate relay position optimization. In contrast with recent uses of reinforcement learning in the context of communication-aware robotics which consider finite horizon, cumulative costs (see, *e.g.*, [11], [12]), our DRL approach optimizes a long-run, *i.e.*, infinite horizon, average cost criterion.

3) We empirically demonstrate that dynamic policies can significantly reduce average wait time compared to approaches found in the literature.
4) Finally, we employ concepts from polling systems, in which a single server services multiple queues, and in fact, our robotic relay scenario may be viewed as a generalization of these systems. Many open problems remain in the optimization of polling systems [13], and to our knowledge, this is the first paper to use DRL to optimize the server's operation.

II. System Modeling

The section presents models for our UAV-aided relaying system. While we discuss the modeling from a UAV perspective, it could apply for ground robots as well.

A. Communication and Channel Prediction Model

We first present our channel model. We assume the UAV flies at a fixed height and consider a two-dimensional workspace. At a location, x, in the workspace, the Signal-to-Noise Ratio (SNR) is given by $SNR(x) = \Gamma_{\mathrm{T}}\Upsilon(x)$ where Γ_{T} is the transmit power and $\Upsilon(x)$ is the Channel-to-Noise Ratio (CNR), which varies over x due to path loss, shadowing, and multipath fading effects. For a given minimum Bit Error Rate (BER) or other Quality of Service (QoS) requirement, reliable communication is only achieved if the SNR exceeds a threshold, *i.e.*, $SNR(x) \geq SNR_{\mathrm{th}}$. We assume that the transmission power is fixed, inducing a minimum required CNR, Υ_{th}, so the spatially varying channel determines where communication can occur.

The UAV must assess the channel quality to successfully relay data between a source-destination pair. In general, the UAV will not know the CNR at every point in the workspace but rather must rely on a few *a priori* readings to predict the channel at unvisited locations. We predict the channel quality using the stochastic approach developed in [14], which uses a spatial stochastic process model to account for the effects of path loss, shadowing, and multipath fading. The channel is characterized by path loss intercept K_0, path loss exponent, n_{PL}, shadowing power α^2, shadowing decorrelation distance β, and multipath fading power σ^2. Given a few prior channel measurements, the CNR (in dB) at an unvisited location x, $\Upsilon_{\mathrm{dB}}(x)$, is modeled by a Gaussian random variable with mean $\mathbb{E}[\Upsilon_{\mathrm{dB}}(x)]$ and variance $\Sigma(x)$. For brevity, we omit the details of these calculations and refer the reader to [14].

With this Gaussian process model, we can find the probability that the CNR at a point exceeds the CNR threshold imposed by the QoS. Specifically, at point x: $\mathrm{P}(\Upsilon_{\mathrm{dB}}(x) \geq \Upsilon_{\mathrm{th,\,dB}}) = \mathbb{Q}((\Upsilon_{\mathrm{th,\,dB}} - \mathbb{E}[\Upsilon_{\mathrm{dB}}(x)])/\sqrt{\Sigma(x)})$, where $\mathbb{Q}(\cdot)$ represents the complementary cumulative distribution function of the standard normal distribution.

For successful end-to-end communication, the CNR for both the source-to-UAV and UAV-to-destination channels must exceed the threshold $\Upsilon_{\mathrm{th,dB}}$. We call the set of points where this holds the *true relay region*: $R_i^{\mathrm{true}} = \{x \in \mathbb{R}^2 | \Upsilon^{\mathrm{true}}_{i,s,\mathrm{dB}}(x) \geq \Upsilon_{\mathrm{th,dB}}, \Upsilon^{\mathrm{true}}_{i,d,\mathrm{dB}}(x) \geq \Upsilon_{\mathrm{th,dB}}\}$, where $\Upsilon^{\mathrm{true}}_{i,s,\mathrm{dB}}(x)$ and $\Upsilon^{\mathrm{true}}_{i,d,\mathrm{dB}}(x)$ are the true CNR in dB at location x of, respectively, the source-to-UAV and UAV-to-destination channels for the i^{th} source-destination pair.

As R_i^{true} is unknown, the UAV predicts the relay regions using the channel prediction framework discussed above. For independent channels, the probability of successful communication between the i^{th} source-destination pair via the mobile UAV at position x is

$$p_{\mathrm{i,sd}}(x) = \mathrm{P}(\Upsilon_{i,s,\mathrm{dB}}(x) \geq \Upsilon_{\mathrm{th,dB}}) \cdot \mathrm{P}(\Upsilon_{i,d,\mathrm{dB}}(x) \geq \Upsilon_{\mathrm{th,dB}}), \quad (1)$$

where $\Upsilon_{i,s,\mathrm{dB}}(x)$ and $\Upsilon_{i,d,\mathrm{dB}}$ are the predicted CNR in dB at location x of, respectively, the source-to-UAV and UAV-to-destination channels for the i^{th} source-destination pair. Then given a threshold probability, p_{th}, we may estimate R_i^{true} with the set $R_i = \{x \in \mathbb{R}^2 | p_{i,\mathrm{sd}}(x) \geq p_{\mathrm{th}}\}$, which we call the *predicted relay region*.

All links are of bandwidth BHz, and both the source and mobile relay transmit with a fixed spectral efficiency ξ b/s/Hz. The source and the UAV may transmit simultaneously, so the service rate when the UAV relays is ξBbps. The relay can only service a single link at a time.

B. Motion Model

We consider trajectories where, for each pair, the UAV chooses a single *relay position* $r_i \in R_i$, as a waypoint at which the UAV remains while relaying. The set of these point is denoted as $X_r = \{r_1, ..., r_n\}$. We assume that when in motion, the UAV moves at a constant velocity, v, and the *switching time* $s_{i,j} = ||r_i - r_j||/v$ is the duration of time between the moment the UAV completes service for queue i and begins service at queue j. If the UAV remains at the same queue it has just serviced, it waits a short period, $s_{i,i}$, and if no new data has arrived, it queries the transition policy again.

C. Data Arrival and Servicing

We model data arrival at each source as independent Poisson processes with intensity λ_i bps. The *service time*, $\zeta = 1/(\xi B)$, is the time required to remove a single bit of data from a queue. The *traffic* for a single queue is denoted by $\rho_i = \lambda_i \zeta$. System-level values are denoted with a subscript "s", so that $\lambda_s = \sum_{i=1}^n \lambda_i$ and $\rho_s = \sum_{i=1}^n \rho_i$. We assume the UAV will continue to service a queue until it is empty, as this results in the smallest average wait time [15].

D. Transition Policies and Markov Chain Model

The *transition policy* determines the sequence in which the UAV services source-destination pairs. Generally, a policy may be deterministic (*e.g.*, a cyclic policy) or stochastic (*e.g.*, the policies studied in [5]). We consider a broad class of dynamic, stochastic policies, $\mu : \Omega \to \Delta(N)$, where Ω is the observation space, $N = \{1, ..., n\}$ is the set of indices of the source-destination pairs, and $\Delta(N)$ is the probability simplex over N. The choice of Ω is discussed next.

We assume the UAV knows the current queue it is servicing, q_k, and the queue lengths at each destination are either

fully observed (FO) or *partially observed* (PO). In the FO case, $\Omega^{\text{FO}} = N \times Z_{\geq 0}^{n}$, where $Z_{\geq 0}$ is the set of non-negative integers. The PO case assumes that the UAV knows only the time since it last serviced each pair, with observation space $\Omega^{PO} = N \times \mathbb{R}_{\geq 0}^{n}$.

The policy is queried at each *service completion instant*, *i.e.*, upon emptying the current queue, and the duration of the UAV's operation may be decomposed into a sequence of *stages*, with the k^{th} stage beginning and ending at the k^{th} and $(k+1)^{\text{th}}$ service completion instants, respectively. Let $L_k = [L_{1,k}, ..., L_{n,k}]$, $T_k = [T_{1,k}, ..., T_{n,k}]$, and $q_k \in \{q_1, ..., q_n\}$ denote the queue lengths, times since last visit, and the source at which service has just been completed, respectively, at the k^{th} service completion instant. Under a specific transition policy, μ, the sequence of state variables $\mathscr{L}_\mu := \{\omega_k\}_{k \geq 0}$ forms a Markov chain, with the state given by (q_k, L_k) and (q_k, T_k) for the FO and PO observation spaces, respectively. The policy is said to be *stable* if $\mathscr{L}_\mu$ is positive recurrent.

The *average wait time* of the system is the average time between the moment a bit enters the system and the moment it begins to be relayed from the source to the destination. As the system of multiple queues may alternatively be viewed as a single queue (albeit not first-come-first-serve), Little's Law [16] applies to the overall system, and average wait time $\bar{W}$ may be expressed as

$$\bar{W} = \lim_{t \to \infty} \int_0^t L(t)dt / (\lambda_s t) \quad , \tag{2}$$

where $L(t)$ is the total number of bits in the system at time t. This limit is guaranteed to exist for all stable policies.

Relationship with Polling systems: The considered scenario can be viewed as a generalization of a polling system, in which a single server services multiple queues. In these systems, optimization is performed over the transition policy while switching times are considered fixed. In our problem, we additionally optimize over switching times, though these are constrained by the robot's fixed velocity and the geometry of the relay regions.

Relationship between Traffic and Energy Consumption: We note that for any stable policy, the proportion of operation time spent servicing and traveling is well approximated by ρ_s and $(1-\rho_s)$, respectively. A surprising consequence of this fact is that energy consumption is approximately constant for all stable policies when motion power and communication power are constant [5]. Thus, we do not explicitly consider energy consumption when optimizing trajectories.

III. Problem Formulation

We are interested in finding the dynamic UAV relay policy, consisting of relay locations X_r and transition policy μ, which together minimize the average wait time:

$$\min_{\mu,\, X_r \in \prod_{i=1}^{n} R_i} \bar{W} \tag{3}$$

While a closed form expression for $\bar{W}$ in terms of system parameters exists for certain transition policies (*e.g.*, under a cyclic policy), no analytic expression is available for general policies of the kind considered here. Furthermore, the non-convexity of the relay regions and the interplay of X_r with the dynamic transition policy μ make solving the problem directly intractable. Therefore, we decouple X_r and μ by iteratively optimizing each independently while keeping the other fixed. The rest of this section poses each decoupled optimization problem separately, and in Section IV we present our approach for joint optimization.

A. Optimal Dynamic Routing Policies

Assuming fixed relay positions, we focus on finding a dynamic transition policy which minimizes the average wait time. We first show how the problem may be posed as a staged decision problem before formulating a specific SMDP for the FO and PO cases. As discussed in Section II, the UAV's operation may be decomposed into a sequence of stages beginning and ending at the service completion moments. Thus, using Eq. (2) and results from [17], our optimization problem may be expressed as:

$$\min_{\mu} \quad \lim_{K \to \infty} \frac{1}{\lambda_S \mathbb{E}[t_K]} \mathbb{E}\left[\sum_{k=1}^{K} \int_{t_{k-1}}^{t_k} L(t)dt \right] \quad . \tag{4}$$

where t_k is the k^{th} service completion instant.

We next introduce average cost semi-Markov decision processes and show how to formulate our problem of interest as such. This allows us to use recent results from the literature to solve our problem, as discussed in Section IV.

1) Average Cost Semi-Markov Decision Process: The average cost semi-Markov decision process is an infinite horizon SMDP specified by a state space Ω, an action set U, state transition probabilities $P(\omega'|\omega,u)$ with $u \in U$, a stage level cost function $c(\omega,u)$, and an average stage duration $d(\omega,u)$. The distinguishing characteristic of these problems is that the objective is to find a policy which minimizes the long term *average* cost rather than cumulative cost, *i.e.*,

$$\min_{\mu} \quad \lim_{K \to \infty} \frac{1}{\mathbb{E}[t_K]} \mathbb{E}\left[\textstyle\sum_{k=1}^{K} c(\omega_k, \mu(\omega_k)) \right] \quad . \tag{5}$$

Under certain technical assumptions, the Bellman equations for these problems can be modified to account for this difference, as shown below:

$$h(\omega) = \min_{u \in U(\omega)} \left[c(\omega,u) - \psi^* d(\omega,u) + \sum_{\omega' \in \Omega} P(\omega'|\omega,u) h(\omega') \right] \tag{6}$$

where ψ^* is the optimal average cost. The solution to these problems depends on the properties of the underlying Markov chain induced by μ, *e.g.*, the number and size of recurrent classes. Further details on average cost SMDP's are found in [18]. We next show that our problem is an SMDP.

2) The Fully Observed Case - SMDP Formulation: For the FO system, states consist of the queue which has just been serviced and all queue lengths, *i.e.*, $\omega_k = (q_k, L_k) \in \Omega^{FO}$. For a policy μ and state ω_k, the probability of ω_{k+1} taking on some value (q,L) given all previous observations is

$$\begin{aligned} &P(\omega_{k+1} = (q,L) | \omega_k, ..., \omega_0, \mu) \\ &\quad = P(\mu(\omega_k) = q) P(L_{k+1} = L | \omega_k, q) \quad . \end{aligned} \tag{7}$$

The probability $P(L_{k+1} = L | \omega_k, q)$ can be found using the properties of Poisson processes, but we omit the derivation for brevity. Thus, the state transition process is Markovian. The action set consists of N, the choice of which source-

destination pair to visit next, though we reiterate that our more generic policies give a probability distribution over N.

From Eq. (4), the stage cost is given by

$$c(\omega_k, q_{k+1}) = \mathbb{E}\left[\int_{t_{k-1}}^{t_k} L(t)dt \big| q_k, L_k, q_{k+1}\right] \ , \tag{8}$$

and the average stage duration is given by

$$d(\omega_k, q_{k+1}) = \mathbb{E}[t_k - t_{k-1} | q_k, L_k, q_{k+1}] \ . \tag{9}$$

Closed form expressions for (8) and (9) can be derived but are omitted for brevity.

Although the state space is countably infinite, it can be shown that all stable policies induce a positive recurrent Markov chain so that the limit in (5) exists and, consequently, the average-cost Bellman equations in (6) hold.

3) The Partially Observed Case: We now consider the PO case, in which the UAV knows only the time since each queue was last visited. A SMDP may be derived for this case, but in practice, a PO policy may be constructed from a FO policy[1], μ^{FO}, by first finding the *expected* FO state given the PO state, $\mathbb{E}[\omega^{\text{FO}}|\omega^{\text{PO}}] = (q, [\lambda_1 T_1, ..., \lambda_n T_n])$, then passing the expected state to μ^{FO}. In practice, these policies result in wait times comparable to those achieved using FO policies and avoid directly optimizing the partially observable SMDP, which requires much more computation time than solving the fully observed SMDP.

B. Relay Positions

The relationship between relay positions and average wait time given a fixed transition policy is non-convex and does not lend itself to existing algorithmic tools, so we propose a heuristic method for selecting relay positions. Considering again the entire system as a single queue, the switching times can be seen as idling periods. We then choose X_r to minimize average idling period, *i.e.*, the average switching time:

$$\min_{X_r \in \prod_{i=1}^n R_i} \quad \sum_{i=1}^n \sum_{j \neq i} P_{ij} ||r_i - r_j||_2 \tag{10}$$

where P_{ij} is the percentage of transitions that occur from pair i to pair j, which may be calculated by keeping track of transitions while simulating the policy for a long period.

As shown in [5], this problem can be reformulated as a MISOCP by decomposing each relay region R_i into a set of convex, polygonal regions and introducing a number of binary indicator variables. State-of-the-art solvers can provide solutions to this problem with guarantees of optimality and finite time convergence. For brevity, we omit the full formulation and refer the reader to [5] for details.

IV. Finding Optimal Dynamic Policies

In this section, we describe in detail our approach to jointly find relay positions, X_r, and policies, μ, with particular focus on solving the SMDP presented in Section III-A.

Average cost MDP's with countably-infinite state spaces are difficult to solve directly. A common approach approximates the average cost problem by reformulating it as a discounted cumulative cost problem. However, recent work by Zhang and Ross [10] proposes a DRL method for solving average costs MDP's directly, and in the context of robotics, they show that this can lead to significant improvement over the discounted cumulative cost approximation. Our method, presented in Algorithm 1, extends their method to (1) handle SMDP's and (2) account for relay position optimization.

[1]While the robot may not know real-time queue lengths, it is reasonable that data would be available when training the routing policy

Algorithm 1: A DRL-based algorithm for minimizing average cost in semi-Markov decision process, based on [10]. Subscript ϕ^p indicates the value is calculated under the policy parameterized by ϕ^p, while V_{ϕ^c} is the value function parameterized by ϕ^c.

Input: Number of trajectories M, Trajectory length K, Policy parameters ϕ^{p}, Critic parameters ϕ^{c}, Relay position update frequency J.

for $i \leftarrow 1$ **to** M **do**

- **Step 1:** Generate K-stage trajectory $\{\omega_k, q_{k+1}, \hat{c}(\omega_k, q_{k+1}), \Delta t_k, \omega_{k+1}\}$ with policy $\mu_{\phi^{\text{p}}}$.
- **Step 2:** Estimate average cost as $\psi_{\phi^{\text{p}}} = \sum_{k=1}^N \hat{c}(\omega_k, q_{k+1}) / \sum_{k=1}^N \Delta t_k$.
- **Step 3:** Estimate value function as $V_k^{\text{target}} = \hat{c}(\omega_k, q_{k+1}) - \psi_{\phi^{\text{p}}} \Delta t_k + V_{\phi^{\text{c}}}(\omega_{k+1})$.
- **Step 4:** Estimate advantage as $A(\omega_k, a_k) = V_k^{\text{target}} - V_{\phi^{\text{c}}}(\omega_k)$.
- **Step 5:** Use value and advantage estimates to update policy and critic parameters using Proximal Policy Optimization (PPO) [19].
- **Step 6:** Every J trajectories, re-estimate transition probabilities P given current policy and solve (10) to update $S(X_r)$.

end

The DRL method approximates the policy and a value function using artificial neural networks (ANN's) parameterized by a set of weights which we denote ϕ^p and ϕ^c, respectively. The input of both the policy and value function networks is the stacked state vector, with q_k one-hot encoded. The output layer of the policy network is an n-dimensional softmax, while the output of the value approximator, the *critic*, is a scalar value. For convenience, we define the value of a state $V_\mu(\omega)$ and both the action bias $\bar{Q}_\mu(\omega, q)$ and advantage $A_\mu(\omega, n)$ of each state-action pair, which play important roles in the algorithm. First, the average-cost action bias function is given by

$$\bar{Q}_\mu(\omega, q) = \mathop{\mathbb{E}}_{\mu, \mathscr{A}}\left[\sum_{k=0}^\infty c_k - \psi_\mu d_k \big| \omega_0 = \omega, q_1 = q\right] \tag{11}$$

where $\mathscr{A}$ is the Poisson arrival process over all queues, ψ_μ is the average cost under policy μ, and c_k and d_k are the cost and duration of stage k, respectively. The value and advantage functions can then be respectively expressed as

$$\begin{aligned} V_\mu(\omega) &= \sum_{q=1}^n P(\mu(\omega) = q) \bar{Q}_\mu(\omega, q) \\ A_\mu(\omega, q) &= \bar{Q}_\mu(\omega, q) - V_\mu(\omega) \quad . \end{aligned} \tag{12}$$

The algorithm proceeds by generating a number, M, of K-stage trajectories, with K sufficiently large so that the number of visits to each state approximates the proportions indicated by the stationary distribution on $\mathscr{L}_\mu$ (Step 1). For each trajectory, the samples and the critic network are used

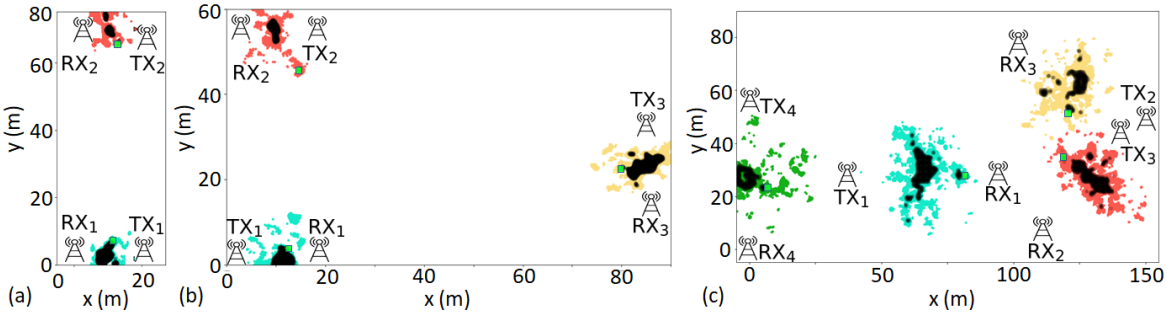

Fig. 2: Relay regions for three example systems with (a) $n = 2$, (b) $n = 3$, and (c) $n = 4$. Sources and destinations are labeled TX and RX, respectively. The green squares indicate relay positions. The colored regions indicate the true relay regions, and the darker portions correspond to the predicted relay regions with $p_{\text{th}} = 0.7$. For $n = 2$, $\lambda_1 = 1.44\,\text{Mb/s}$, $\lambda_2 = 0.16\,\text{Mb/s}$. For $n = 3$, $\lambda_1 = 0.72\,\text{Mb/s}$, $\lambda_2 = 0.72\,\text{Mb/s}$, $\lambda_3 = 0.16\,\text{Mb/s}$. For $n = 4$, $\lambda_1 = 0.72\,\text{Mb/s}$, $\lambda_2 = 0.72\,\text{Mb/s}$, $\lambda_3 = 0.24\,\text{Mb/}$, $\lambda_4 = 0.72\,\text{Mb/s}$. See color PDF for better viewing.

to estimate the average cost ψ_μ (Step 2), the value of each state (Step 3), and the advantage of each state-action pair encountered during the trajectory (Step 4). The estimated values and and advantages are then used to update the policy and critic parameters as described in [19] (Step 5). During training, we keep track of the transitions between regions and use these to estimate the transition probabilities P_{ij} needed to periodically update the relay positions by solving (10) (Step 6). We have modified steps 1 through 5 from [10] to account for the *semi*-Markov nature of our problem, and we also add Step 6 to handle our particular problem of interest. While updates to the relay positions will shift the underlying cost structure the DRL algorithm tries to learn, the high-level structure, *i.e.*, the arrival rates and relative positioning of the relay regions, remains unchanged, and in practice, we find that Step 6 does not adversely affect the algorithm's stability.

V. Numerical Results

In this section, we present numerical results for systems simulated using real-world channel parameters. We first illustrate policies found with Algorithm 1, then compare their performance against other approaches found in the literature.

Our simulations consider the two-, three-, and four-pair systems shown in Fig. 2. In all simulations, the data rate is $\xi = 8\,\text{b/s/Hz}$ and the channel bandwidth is $1\,\text{Mhz}$, so that $\zeta = 0.125\,\text{s/Mb}$. The UAV and sources transmit with a power of $\Gamma_t = 100\,\text{mW}$, the receiver noise power is $-80\,\text{dBm}$, and the acceptable SNR threshold for all the channels is set to $33\,\text{dB}$. Given the transmit power, the SNR threshold translates to a CNR threshold of $\Upsilon_{\text{th, dB}} = 13\,\text{dBm}$. We fix the UAV velocity at $v = 1\,\text{m/s}$. The value of the channel parameters introduced in Section II-A are chosen to be consistent with empirical studies of air-to-ground channels as reported in [20]: $K_0 = -15\,\text{dB}$, $n_{\text{PL}} = 5.2$, $\alpha^2 = 16$, $\beta = 2.09\,\text{m}$, and $\sigma^2 = 1.5$. For the four-pair system, the path loss parameters are modified to $K_0 = -5\,\text{dB}$, $n_{\text{PL}} = 4.5$ with all other parameters the same. The threshold probability of connectivity used for relay region prediction is $p_{\text{th}} = 0.7$. For each system, we find relay positions and transition policies using Algorithm 1. The structure of the policy and critic networks for each system is given in Table I.

A. Near-Optimal Dynamic Policies

In this section, we find the relay positions and transition policy using Algorithm 1 for all systems shown in Fig. 2

	# Layers	Layer Dim.
$n = 2$	4	8
$n = 3$	5	16
$n = 4$	8	32

TABLE I: Structure of the the policy and critic ANN's. Each layer consists of a fully connected linear layer followed by an Exponential Linear Unit (ELU) activation. See Section IV for further details on network architecture.

and examine the resulting policies. Fig. 3 depicts the last 10 minutes of a two-hour operation period under these policies. For the system shown in Fig. 2 (a), the UAV services data at q_1 until sufficient data (around $50\,\text{Mb}$) accumulate at q_2, as illustrated in Fig. 3 (top). After servicing q_2, the UAV always immediately moves back to q_1. Similarly, for the system shown in Fig. 2 (b), the UAV moves between q_2 and q_1 until the queue length at q_3 exceeds about $50\,\text{Mb}$. This is shown in Fig. 3 (middle).

For the four-queue system shown in Fig. 2 (c), Fig. 3 (bottom) shows that every time the UAV visits q_2, it also visits q_3, since these two are close together, so that even though λ_2 is significantly smaller than the other arrival rates, q_2 is visited as frequently as q_3 and q_4. Furthermore, after servicing q_4, the UAV will always move to q_1 as it lies in the general direction of the other relay positions.

B. Comparison of Dynamic Policies to State-of-the-Art

We compare the fully observed (FO) dynamic UAV relay policies found using Algorithm 1 to existing strategies in the literature. We specifically consider (1) cyclic policies based on the traveling salesperson with neighborhoods problem, (2) stochastic policies, in which $P(\mu(\omega) = i) = \pi_i, \forall \omega \in \Omega$, and (3) policies described by a deterministic sequence derived from the stochastic policies using the golden ratio. Details on the latter can be found in [5]. We also compare again the partially observed (PO) policies described in Section III-A.3. For each policy, we simulate twenty four-hour operation periods, and average the wait time.

The FO dynamic policies significantly reduce average wait time. For example, with $n = 4$, it gives an average wait time of $141\,\text{s}$, but for the cyclic, stochastic, and golden ratio policies, the average wait times are $161\,\text{s}$, $224\,\text{s}$, and $176\,\text{s}$, respectively. Fig. 4 shows the percent increase in average wait time under the cyclic, stochastic, and golden ratio policies compared to the FO dynamic policy. As can be seen, static policies can result in an over 50% increase in average wait time, and while one method may perform

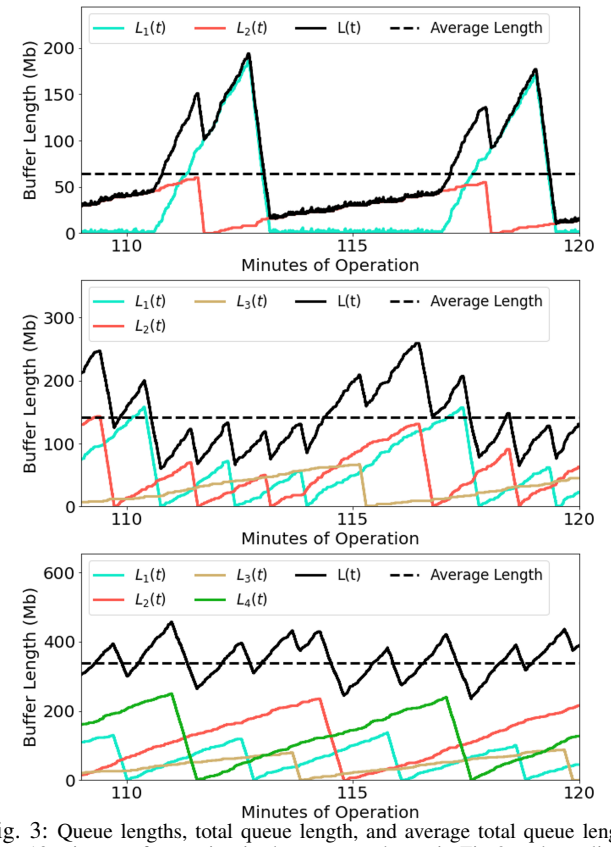

Fig. 3: Queue lengths, total queue length, and average total queue length over 10 minutes of operation in the systems shown in Fig 2 under policies found with Algorithm 1. $L_i(t)$ is the length of queue i at time t. See color PDF for better viewing.

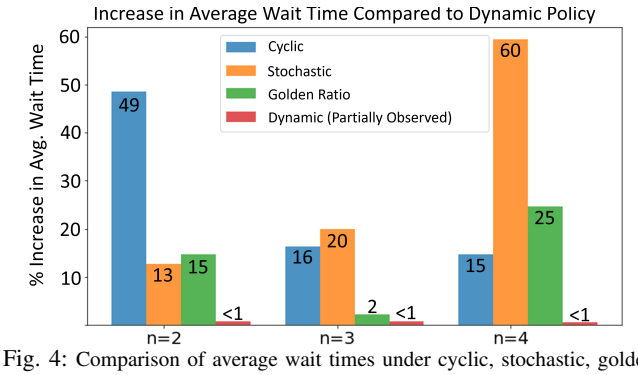

Fig. 4: Comparison of average wait times under cyclic, stochastic, golden ratio, and partially observed dynamic policies as a percentage of wait time under the fully observed dynamic policy. See color PDF for better viewing.

close to the dynamic policy in one scenario, in a different scenario, it performs much worse. We also note that the PO dynamic policies perform on par with the FO dynamic policies, increasing average wait time by less than 1%.

While the cyclic, stochastic, and golden ratio policies may be found quickly, our DRL-based approach requires the simulation of many extended trajectories. To reduce training time, we use transfer learning to intelligently initialize the weights of the policy and critic networks. Further reducing the computational cost of our DRL-based approach by optimizing transfer learning is an important area for future work.

VI. Conclusions

This paper considered a UAV acting as a relay between pairs of distant source and destination communication nodes. We posed the problem of finding the optimal dynamic UAV relay policy, consisting of relay positions and a dynamic transition policy. We found approximate solutions by decoupling the choice of relay locations and policy and showing that for fixed relay positions, the problem of finding an optimal transition policy can be formulated as a average cost SMDP. Using a DRL-based approach, we proposed an algorithm to find approximate solutions to the SMDP. Through simulation in realistic channel environments, we empirically showed that dynamic policies can significantly reduce average wait times.

References

[1] A. Muralidharan and Y. Mostofi, "Communication-Aware Robotics: Exploiting Motion for Communication," *Annu. Rev. of Control, Robot., and Auton. Syst.*, 2020.

[2] S. Evmorfos, D. Kalogerias, and A. Petropulu, "Adaptive Discrete Motion Control for Mobile Relay Networks," *Frontiers in Signal Processing*, vol. 2, 2022.

[3] A. Pogue, S. Hanna, A. Nichols, X. Chen, D. Cabric, and A. Mehta, "Path Planning Under MIMO Network Constraints for Throughput Enhancement in Multi-robot Data Aggregation Tasks," in *IEEE/RSJ Int. Conf. on Intell. Robots and Syst.*, 2020, pp. 11 824–11 830.

[4] J. George, C. T. Yilmaz, A. Parayil, and A. Chakrabortty, "A Model-Free Approach to Distributed Transmit Beamforming," in *IEEE Int. Conf. on Acoust., Speech and Signal Process.*, 2020, pp. 5170–5174.

[5] W. Hurst and Y. Mostofi, "Optimization of Mobile Robotic Relay Operation for Minimal Average Wait Time," *IEEE Trans. on Wireless Commun.*, 2022.

[6] M. Bliss and N. Michelusi, "Power-Constrained Trajectory optimization for Wireless UAV Relays with Random Requests," in *IEEE Int. Conf. on Commun.*, 2020, pp. 1–6.

[7] S. C. Pinto, S. B. Andersson, J. M. Hendrickx, and C. G. Cassandras, "Multi-Agent Persistent Monitoring of Targets with Uncertain States," *IEEE Trans. on Autom. Control*, 2022.

[8] G. D. Çelik and E. H. Modiano, "Dynamic vehicle routing for data gathering in wireless networks," *IEEE Conf. on Decis. and Control*, pp. 2372–2377, 2010.

[9] L. E. F. Luyo, A. Agra, R. Figueiredo, and E. O. Anaya, "Mixed integer formulations for a routing problem with information collection in wireless networks," *Eur. J. Oper. Res.*, vol. 280, pp. 621–638, 2020.

[10] Y. Zhang and K. W. Ross, "On-Policy Deep Reinforcement Learning for the Average-Reward Criterion," *ArXiv*, vol. abs/2106.07329, 2021.

[11] U. Challita, W. Saad, and C. Bettstetter, "Interference Management for Cellular-Connected UAVs: A Deep Reinforcement Learning Approach," *IEEE Trans. on Wireless Commun.*, vol. 18, no. 4, pp. 2125–2140, 2019.

[12] B. Galkin, E. Fonseca, R. Amer, L. A. DaSilva, and I. Dusparic, "REQIBA: Regression and Deep Q-Learning for Intelligent UAV Cellular User to Base Station Association," *IEEE Trans. on Veh. Technol.*, vol. 71, no. 1, pp. 5–20, 2022.

[13] V. Vishnevsky and A. V. Gorbunova, "Application of machine learning methods to solving problems of queuing theory," in *Information Technologies and Mathematical Modelling. Queueing Theory and Applications*, A. Dudin, A. Nazarov, and A. Moiseev, Eds. Cham: Springer International Publishing, 2022, pp. 304–316.

[14] M. Malmirchegini and Y. Mostofi, "On the Spatial Predictability of Communication Channels," *IEEE Trans. on Wireless Commun.*, 2012.

[15] Z. Liu, P. Nain, and D. Towsley, "On optimal polling policies," *Queueing Syst*, vol. 11, pp. 59–83, 1992.

[16] J. D. C. Little, "A Proof for the Queuing Formula: L = λW," *Operations Research*, vol. 9, pp. 383–387, 1961.

[17] S. M. Ross, *Applied Probability Models with Optimization Applications*. Holden-Day, 1970.

[18] D. P. Bertsekas, *Dynamic Programming and Optimal Control 4 th Edition , Volume II*. Athena Scientific, 2015.

[19] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal Policy Optimization Algorithms," *ArXiv*, 2017.

[20] W. A. G. Khawaja, I. Guvenc, D. W. Matolak, U.-C. Fiebig, and N. Schneckenburger, "A Survey of Air-to-Ground Propagation Channel Modeling for Unmanned Aerial Vehicles," *IEEE Commun. Surv. & Tut.*, vol. 21, pp. 2361–2391, 2019.