Piggybacking on UAV Package Delivery Systems to Simultaneously Provide Wireless Coverage: A Deep Reinforcement Learning-Based Trajectory Design

Jeong Min Kong
Department of Electrical and Computer Engineering
University of Toronto
Toronto, Canada
jeong.kong@mail.utoronto.ca

Elvino Sousa
Department of Electrical and Computer Engineering
University of Toronto
Toronto, Canada
es.sousa@utoronto.ca

***Abstract*—Various studies have explored the possibility of utilizing unmanned aerial vehicles (UAVs) as last-mile package delivery agents and aerial base stations in recent years. Despite the tremendous attention in these applications, nearly all of the studies assumed that the UAVs are serving just one purpose, but not both simultaneously; however, for a number of reasons, such as traffic congestion in the sky and energy & resource inefficiency problems, it seems more suitable for the UAVs to be serving several services concurrently. A few papers investigated the case in which the UAV acts as both a package delivery agent and a wireless transceiver, but in all of them, package delivery time constraints were never considered. Stemming from the observation that there are various ongoing industrial projects in UAV-based package delivery, we investigate the possibility of piggybacking on UAV-based package delivery infrastructures to also provide wireless coverage, and consider the problem of designing UAV trajectories that maximize the cumulative downlink sum rate of the ground communication users while simultaneously delivering packages under strict delivery time constraints. We use deep Q-learning (DQL) to solve this optimization problem, and demonstrate the successful formulation & implementation of our algorithm through several simulations.**

***Index Terms*—6G, UAV, Multi-Purpose UAV, UAV-Based Delivery, UAV-Assisted Communications, Deep Reinforcement Learning, Deep Q-Learning**

I. Introduction

Unmanned Aerial Vehicles (UAVs) are expected to disrupt various industries, including environment monitoring, surveillance, filmmaking, and in particular, package delivery and 6th-generation wireless communication systems (6G) [1]. Application of UAVs in package delivery has gained tremendous attraction over the past few years, as it is expected to reduce the delivery cost (including the labor cost), delivery time, and strong reliance on human resources. In recent years, various literature investigated the idea of utilizing UAVs as last-mile package delivery agents, and focused on the problem of time & energy efficient scheduling and routing of UAVs in different package delivery frameworks, including a fully UAV-based delivery system as well as joint ground & UAV-based delivery systems [3-6]. In wireless, numerous investigations have centered around the deployment of UAVs as aerial base stations. UAVs' abilities, such as mobility and capability to reach an altitude of up to 300 m, would enable wider ground wireless coverage and lead to more line-of-sight (LOS) communication links, and many extensively considered the problem of designing flight trajectories that can satisfy the quality of service (QoS) constraints of the ground user terminals/equipments (UEs) while considering factors such as time & energy constraints, space accessibility limitations, and obstacle avoidance [1,2]. Until now, almost all research in these applications have assumed that UAVs are serving just one service, ie. either package delivery or wireless, but not both. However, for a number of reasons, it seems more suitable for UAVs to be serving both purposes simultaneously. As discussed in [7], utilizing different groups of UAVs for different purposes could cause heavy traffic congestion in the sky, and in turn, cause significant performance degradation across both domains. Furthermore, large amounts of energy and resources could be wasted by unnecessarily employing excess number of UAVs. It is also important to highlight that while there are various ongoing large-scale industrial projects in UAV-based package delivery, such as Amazon Prime Air [8], to the best of authors' knowledge, no major telecommunication companies have yet disclosed that they will be developing the infrastructures necessary for enabling UAV-assisted communications. Given that the expected launch date of 6G is now less than 10 years away [9], it seems more feasible for the telecommunication companies to instead partner with UAV package delivery companies that already have the required infrastructures for realizing UAV-assisted communications. Of course, because these infrastructures are developed solely for the purpose of UAV-based package delivery, providing wireless service would be a subordinate, secondary objective. To this end, in this paper, we investigate the possibility of *piggybacking* on UAV package delivery systems to simultaneously provide wireless coverage. Under the *strict* constraint that all of the required deliveries must be completed in a specified time interval, we design UAV trajectories that maximize the cumulative downlink sum rate of the ground UEs with the utilization of deep reinforcement learning (DRL).

II. Related Work

As discussed, while extensive research has been done in the design of UAV trajectory for UAV-assisted communications and UAV-based package delivery individually, only a few papers explored the possibility of multi-purpose UAVs. The first paper to consider such scenario is [10]. In this paper, the authors propose an algorithm to find the shortest UAV trajectories for completing all of the required package deliveries while simultaneously always providing uniform wireless coverage in the considering region. While their idea of multi-tasking UAVs is visionary, it is important to emphasize that in their problem, the wireless QoS is considered as the constraint instead of the package delivery time, which is exact opposite to the formulation in our problem. A recent paper [7] also studies the possibility of multi-purpose UAVs, by considering a scenario in which a UAV simultaneously delivers a package and collects & delivers internet of things (IoT) data from a ground UE to a ground base station. The authors utilize stochastic geometry and optimization techniques to design trajectories that maximize the amount of collected & delivered IoT data while minimizing the round trip time, subject to a battery constraint. If it is possible for all of the IoT data to be collected & delivered subject to the battery constraint, then among all possible trajectories which can accomplish that, they aim to find the one that has the minimum round trip time. In the case not all of the IoT data can be collected & delivered subject to the battery constraint, they aim to find the trajectory that maximizes the collected & delivered IoT data assuming all of the battery will be used. In their study, the only concern related to package delivery is that it just needs to be completed before the end of the round trip; how fast the task is accomplished is not considered in the trajectory design. While both IoT data collection & delivery and package delivery are considered, in their formulated problem, the package delivery task is secondary to the IoT task. Similar to the first paper, this is fundamentally contrasting to our problem, in which the wireless service task is secondary to the package delivery task. In addition to this, while they only consider single-ground-UE and single-package-delivery scenario, we consider a more realistic multiple-ground-UEs and multiple-package-deliveries scenario in our problem.

III. System Model

Consider a map with dimensions $N \times N$ [m]. Let n_g represent the number of ground UEs in the environment, and n_d represent the number of delivery points in the environment. Let $\mathbf{G}$ be a vector containing the 2-D coordinates of the ground UEs, and $\mathbf{G}_i$, $i \in \{0, 1, ..., n_g - 1\}$, denote the 2-D coordinate of the ith ground UE. Similarly, let $\mathbf{D}$ be a vector containing the 2-D coordinates of the delivery points, and $\mathbf{D}_j$, $j \in \{0, 1, ..., n_d - 1\}$, denote the 2-D coordinate of the jth delivery point. There is one multi-purpose UAV in the environment, positioned initially at a 2-D coordinate denoted as (a, b). The objective of this UAV is to travel in a trajectory that maximizes the cumulative downlink sum rate of the ground UEs in the time duration that all of the package deliveries must be completed. Let T [s] be this maximum time in which all of the deliveries must be made. We denote (x_t, y_t) as the 2-D coordinate of the UAV at time t, and represent $\mathbf{P} \in \mathbb{R}^T$ as a vector containing the sequence of UAV's (x_t, y_t) coordinates at discrete time steps from $t = 0$ to $t = T - 1$ [s], ie. $\{(x_0, y_0),(x_1, y_1),...,(x_{T-1}, y_{T-1})\}$. A delivery is considered completed at delivery point j at time t when UAV's 2-D coordinate, (x_t, y_t), overlaps with the 2-D coordinate of delivery point j, $\mathbf{D}_j$. In this paper, we assume that all of the ground UEs have the same height, and that the UAV is always travelling at a fixed height with a constant, average speed. We denote the height difference between the ground UEs and the UAV as h [m], and the speed of the UAV as v [m/s].

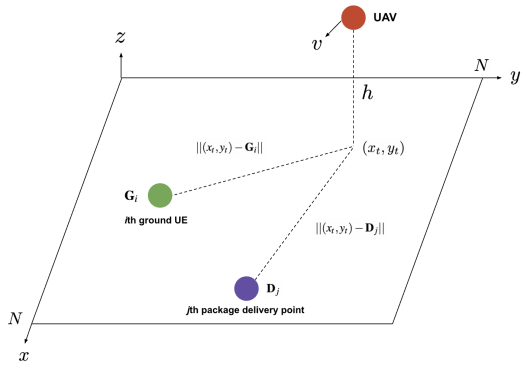


Fig. 1. Illustration of the map described in our System Model.

A. Communication Channel

The communication channel between the UAV and a ground UE is modeled by log-distance path loss and Rayleigh small-scale fading, and each link is modeled as an orthogonal point-to-point channel.

The downlink information rate of the ith ground UE at time t [bps/Hz] is defined as

$$R_i(t) = \log(1 + \frac{P_T}{P_N} \cdot L_i(t)) \tag{1}$$

where P_T is the transmitted power from the UAV [W], P_N is the noise power [W], and $L_i(t)$ is the channel loss corresponding to the ith ground UE at time t. The channel loss is expressed by

$$L_i(t) = d_i(t)^{-\alpha} \cdot 10^{X_{Rayleigh/10}} \tag{2}$$

with α being the path loss exponent, $X_{Rayleigh}$ being the Rayleigh random variable with scaling factor = 1, and $d_i(t)$ being the distance between the UAV and the ith ground UE at time t [m].

$$d_i(t) = \sqrt{h^2 + \|(x_t, y_t) - \mathbf{G}_i\|} \tag{3}$$

The downlink sum rate at time t [bps/Hz] is given by

$$S(t) = \sum_{i=0}^{n_g-1} R_i(t) \tag{4}$$

and hence, the cumulative downlink sum rate over the time duration T [bits/Hz] can be expressed as

$$C = \int_{t=0}^{T} S(t)dt \tag{5}$$

B. Problem Formulation

We approximate the integral in (5) as a Riemann sum, and define our optimization problem as the following:

$$\begin{aligned} \max_{\mathbf{P}} \quad & C \approx \sum_{t=0}^{T-1} S(t) \\ \text{s.t.} \quad & 0 \leq x_t \leq N \\ & 0 \leq y_t \leq N \\ & \|(x_t, y_t) - (x_{t-1}, y_{t-1})\| = v \\ & (x_t, y_t) = \mathbf{D}_j \; \exists \; t, \forall \; j \end{aligned} \tag{6}$$

IV. Methodology

We approach the non-convex maximization problem in (6) using a DRL algorithm called deep Q-learning (DQL), which is an effective method for solving optimization problems that can be modelled as deterministic or Markov decision processes. In this section, we first provide an overview of reinforcement learning, and then present the formulation of (6) as a DQL problem.

A. Reinforcement Learning Background

RL is an area of machine learning where an agent constantly interacts with its environment in order to learn a sequence of decisions/actions that yield the highest cumulative reward possible [11]. The detailed flowchart diagram of RL is shown in Figure 2. At time t, the agent receives state s_t from its environment, which corresponds to the agent's observations of the environment at t. Based on s_t, the agent decides to take action a_t, and as a consequence, the environment goes to a new state s_{t+1}. After evaluating s_{t+1}, the environment gives reward r_{t+1} to the agent, which is its feedback of agent's decision to take action a_t given s_t. Let $\mathcal{S}$ be a finite state space, and $\mathcal{A}$ be a finite action space. Given a state $s \in \mathcal{S}$, the agent decides which action $a \in \mathcal{A}$ to take based on a probability distribution called policy π, which is expressed as

$$\pi(a|s) = P[a_t = a|s_t = s] \tag{7}$$

A function that indicates the quality of taking action a in state s with policy π is called the Q-function, and is given by

$$Q^{\pi}(s, a) = E_{\pi}\{F_t|s_t = s, a_t = a\} \tag{8}$$

where F_t defined as

$$F_t = \sum_{k=0}^{T-t-2} \gamma^k r_{t+1+k} \tag{9}$$

$\gamma \in [0, 1)$ is called the discount factor, and its significance will be discussed later. From (8) and (9), it can be seen that $Q^{\pi}(s, a)$ is the expected discounted cumulative future reward that the agent will receive if it chooses to proceed with action a from state s at time t, and then take actions based on policy π from $t + 1$ and onward. Thus, the higher the Q-value is for some action a, the better choice it is for the agent to proceed with action a. The discount rate γ plays a vital role in the decision-making process because if γ is small, the Q-value is dominated by short-term future rewards, whereas if γ is large, the Q-value is the culmination of long-term future rewards. This implies that as γ is smaller, the agent's actions will be more influenced by gaining short-term rewards, whereas as γ is larger, the agent's actions will be more influenced by gaining long-term rewards.

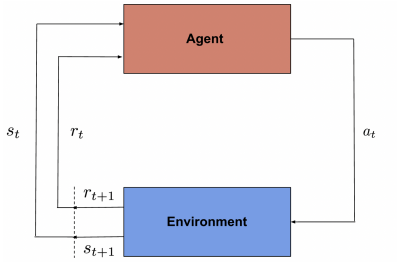


Fig. 2. Reinforcement learning flowchart.

The objective in Q-learning is to find policy π^* that maximizes $Q^{\pi}(s, a)$. The iterative update rule for approaching $Q^{\pi^*}(s, a)$ is described by the Bellman equation

$$Q^{\pi}(s_t, a_t) \leftarrow r_{t+1} + \gamma \max_a Q^{\pi}(s_{t+1}, a) \tag{10}$$

In traditional Q-learning, the Q-function is represented by a table. However, for scenarios with large state and action spaces, this is not an ideal approach due to the exponentially-growing table size. A good alternative is approximating the Q-function using a deep neural network with parameters θ, ie. $Q^{\pi}(s, a, \theta)$ [12]. This neural network-based Q-function is also known as a Q-network, and the process of updating θ of the Q-network to achieve $Q^{\pi^*}(s, a)$ is known as deep Q-learning. To tune the Q-network, at every t, θ is first updated by minimizing the squared error between (10) and $Q^{\pi}(s_t, a_t, \theta)$, expressed as

$$L(\theta) = ((r_{t+1} + \gamma \max_a Q^{\pi}(s_{t+1}, a, \theta)) - Q^{\pi}(s_t, a_t, \theta))^2 \tag{11}$$

After, θ is also similarly updated using a mini-batch of B memories $\{s_{t'}, a_{t'}, r_{t'+1}\}$ from $t' < t$.

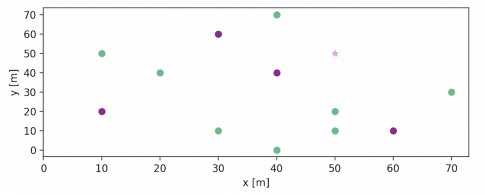
B. DQL-Based Multi-Purpose UAV Trajectory Design

In our DQL problem, the UAV is the *agent*, and the map containing the UAV, the delivery points, and the ground UEs is the *environment*. Our objective is to find a Q-network that solves the optimization problem in (6). We describe the formulation of our state, action set, and reward function next.

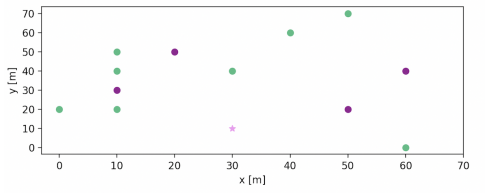
We define the state at time t as

$$s_t = \{t, x_t, y_t, \mathbf{W}(t), \mathbf{C}(t)\} \tag{12}$$

$\mathbf{W}(t) \in \mathbb{R}^{n_d}$ is a vector containing the 2-D distances between the UAV and all delivery points at t, ie. $\mathbf{W}(t)_j = \|(x_t, y_t) - \mathbf{D}_j\|$. $\mathbf{C}(t) \in \mathbb{R}^{n_d}$ is a vector containing the delivery completion status of all delivery points at t, with $\mathbf{C}(t)_j$ = 10 if



(a)



(b)

Fig. 3. Two simulation maps considered in this paper. Pink star is UAV's initial position, purple circles are the delivery points, and green circles are the ground UEs.

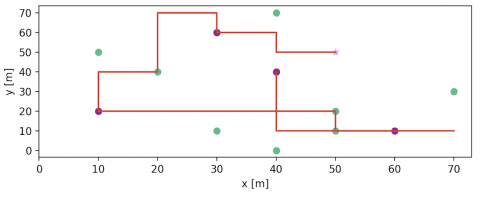
the delivery has been completed at delivery point j by t, and $\mathbf{C}(t)_j = 0$ if the delivery has not yet been completed at delivery point j by t. The action set of the UAV is defined by $\mathbf{A}$ = {'up','down','left','right'}, with 'up' indicating +ve change in y by v, 'down' indicating -ve change in y by v, 'left' indicating -ve change in x by v, and 'right' indicating +ve change in x by v. As discussed before, because the height of the UAV is fixed in our problem, change in height is not part of our action set. Lastly, we define the reward at time t as $r_t = r_t^a + r_t^b + r_t^c$. r_t^a is 400 $\times$ sum SNR at t, ie. $400 \cdot \sum_{i=0}^{n_g-1} \frac{P_T}{P_N} \cdot L_i(t)$, r_t^b is 2000 $\times$ # of deliveries completed at t, and r_t^c is 4000 if the final delivery has been completed at t, and 0 otherwise. As outlined, a substantial reward is provided to the UAV upon successful completion of a delivery. To ensure that the UAV fulfills all deliveries, an additional large reward is granted if the UAV finishes its last delivery. Furthermore, a reward proportional to the sum SNR is given at every t to encourage the UAV to strive for the highest cumulative sum rate possible, given the delivery constraints.

Our Q-network has two hidden layers of 256 neurons with ReLU activation, and uses Adam with a learning rate of 0.001 as the optimizer. The discount rate γ is set to 0.99, and the mini-batch size B is set to 1000. The number of episodes (or rounds) in which the UAV is trained is 20000. Furthermore, we make the UAV take a random action (ie. not based on the output of the Q-network) with a probability of 60% in the first episode, and then let this randomness linearly decline at a rate of -0.0025% per increase in episode; the purpose of this mechanism is to allow the UAV to explore its environment more in the earlier episodes.

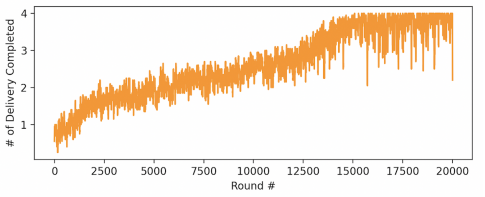
V. Simulations

A. Simulation Set-Ups

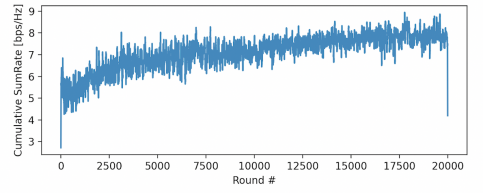
To evaluate our DQL algorithm, we consider $T = 35$ and $T = 20$ for each of the two maps shown in Figure 3. The coordinates of the delivery points, $\mathbf{D}$, the ground UEs, $\mathbf{G}$, and UAV's initial position, (a, b), are vastly different for each



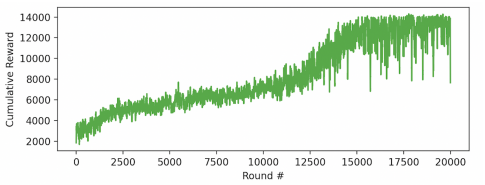
(a)



(b)



(c)



(d)

Fig. 4. Training curves of our algorithm for the map A, $T = 35$ scenario. Note: The curves have been smoothed using convolution, ie. the moving average algorithm, which is why it "appears" like there is a sudden decline at the end of each curve. a) shows the UAV's trajectory, b) shows the number of deliveries completed by the UAV vs round number, c) shows the cumulative sum rate vs round number, and d) shows the cumulative reward attained by the UAV vs round number.

map, while $N = 70$, $h = 10$, $v = 10$, $P_T = 10$, $P_N = 1$, and $\alpha = 2$ for both maps. Moving forward, we will refer to the map shown in Figure 3 a) as map A, and Figure 3 b) as map B.

B. Results and Discussion

The training curves of our algorithm for the map A, T = 35 scenario and map A, T = 20 scenario are shown in Figure 4 and 5, respectively. For both of these scenarios, we can observe that as the episode number increases, the UAV learns to design a trajectory that simultaneously finishes more package deliveries & achieves higher cumulative sum rate, on average. Another observation that can be made, through the final UAV trajectory plots, is that the area covered by the UAV is proportional to T. This result is expected, because as the time requirement to complete all of the deliveries is smaller, the UAV also has less flexibility in its trajectory. Through the final UAV trajectory plots & their corresponding $\mathbf{P}$, we were also able to observe that the UAV spends a great portion of

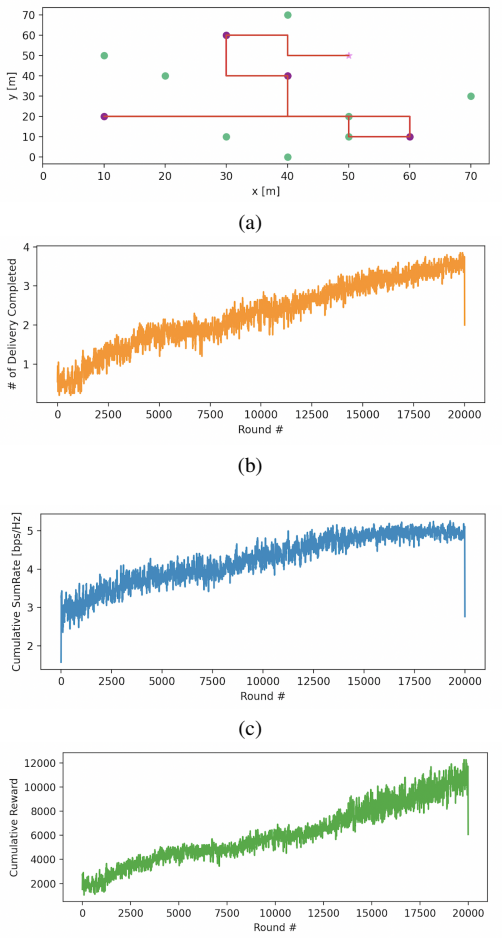


Fig. 5. The smoothed training curves of our algorithm for the map A, $T = 20$ scenario. a) shows the UAV's trajectory, b) shows the number of deliveries completed by the UAV vs round number, c) shows the cumulative sum rate vs round number, and d) shows the cumulative reward attained by the UAV vs round number.

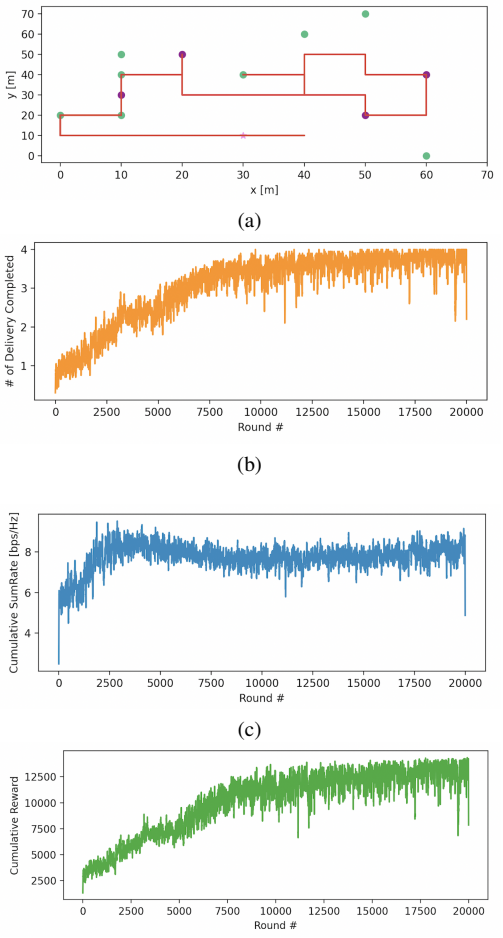
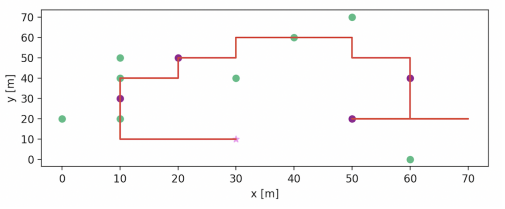


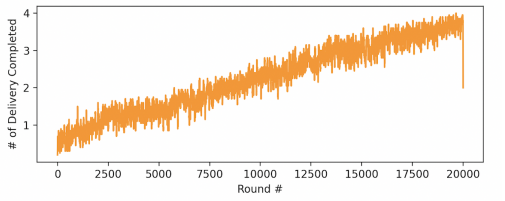
Fig. 6. The smoothed training curves of our algorithm for the map B, $T = 35$ scenario. a) shows the UAV's trajectory, b) shows the number of deliveries completed by the UAV vs round number, c) shows the cumulative sum rate vs round number, and d) shows the cumulative reward attained by the UAV vs round number.

its time looping in the regions with the highest concentration of ground UEs. More specifically, the UAV spent a staggering 40% of its total flight time staying bounded in the $x = [30, 60]$, $y = [0, 30]$ region for the T = 35 scenario, and 55% for the T = 20 scenario. From these results, we can deduce that the UAV optimizes its trajectory to spend as maximal time possible in the regions with the highest concentration of ground UEs, given the delivery requirement constraint.

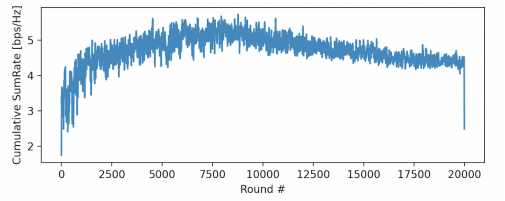
The training curves of our algorithm for the map B, T = 35 scenario and map B, T = 20 scenario are shown in Figure 6 and 7, respectively. Similar to map A scenarios, as the episode number increases, the number of completed deliveries also increases on average. However, the cumulative sum rate graphs show a very interesting pattern that is contrasting to those of map A scenarios. The cumulative sum rate first increases, and peaks at approximately the same episode number in which the deliveries completed is 2; then, it begins to decline until the deliveries completed is 3, and remains relatively constant after. When the UAV only has to complete 2 deliveries, it completes the two deliveries in the left region of the map, $(10, 30)$ and $(20, 50)$, and loops around there for the whole time duration, because that is the area where the ground UEs are most densely populated; this is the reason why the cumulative sum rate is maximum when the deliveries completed is 2. Once the UAV has to complete 3 deliveries, it now has to *leave* the densely populated left region & travel all the way to the right region, where the remaining two delivery points exist. Due to this, as the UAV learns to complete 3 deliveries more frequently, the average cumulative sum rate first declines. However, in this process of learning to complete 3 deliveries, the UAV simultaneously constantly optimizes its trajectory to ultimately determine a path that *maximizes* the cumulative sum rate while doing 3 deliveries; once it is able to find this optimal path, the cumulative sum rate then saturates. The crucial question is, how could the maximum cumulative sum rate for 4 deliveries be the same as for 3 deliveries? Because the third and fourth delivery locations in the right region are very close to each other, it is likely that the optimal path for 3 deliveries is



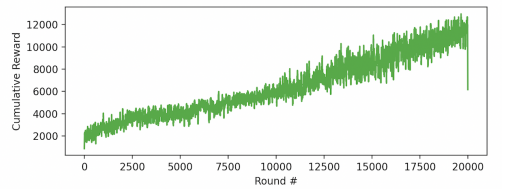
(a)



(b)



(c)



(d)

Fig. 7. The smoothed training curves of our algorithm for the map B, $T = 20$ scenario. a) shows the UAV's trajectory, b) shows the number of deliveries completed by the UAV vs round number, c) shows the cumulative sum rate vs round number, and d) shows the cumulative reward attained by the UAV vs round number.

nearly identical to that for 4 deliveries, and there is thus no discernible difference in the maximum cumulative sum rate between the two.

The proportional relationship between the area covered by the UAV and T from map A scenarios can also be seen in map B scenarios. Furthermore, similar to map A scenarios, we observe that the UAV spends a large amount of its time in regions with the highest concentration of ground UEs. We were able to calculate that the UAV spent 57% of its total flight time staying in $x = [0, 10]$, $y = [20, 50]$ and $x = [30, 50]$, $y = [40, 70]$ regions for the $T = 35$ scenario, and 45% for the $T = 20$ scenario.

VI. Conclusion and Future Work

This paper considered the possibility of piggybacking on UAV package delivery infrastructures to simultaneously provide wireless coverage, and presented a novel DQL algorithm for finding UAV trajectories that maximize the cumulative downlink sum rate of the ground UEs under package delivery time constraints. In the future, in addition to further extending this paper by considering 3-D trajectories and a comprehensive battery consumption model for the UAV, we will also investigate many unique problems arising in this new aerial communications framework. For example, we will account for the fact that unlike communication-only UAVs, package delivery UAVs will have a dynamic weight throughout the course of their flight, that alters after the release of each package they carry. In the subsequent studies, we will address varying package weights for different delivery tasks, and investigate how the sequence in which the UAV dispenses packages influences its battery usage and, consequently, the optimal trajectories. In the wireless communications side, we will consider multiple access channels, and techniques such as time division multiple access (TDMA) and non-orthogonal multiple access (NOMA). To this end, our future work will aim to not solely optimize UAV trajectories, but rather jointly optimize UAV trajectories and user scheduling strategies. Other extensions will include carefully accounting for the safety regulations posed on UAV operations, such as prohibited aerial spaces and maximum weight limitations, and considering the possibility of piggybacking on different UAV-based services, such as joint ground & UAV-based package delivery systems and aerial taxis.

References

[1] M. Mozaffari, W. Saad, M. Bennis, Y. -H. Nam, and M. Debbah, "A tutorial on UAVs for wireless networks: Applications, challenges, and open problems," in *IEEE Communications Surveys & Tutorials*, vol. 21, no. 3, pp. 2334-2360, thirdquarter 2019.

[2] 3GPP TR 36.777: "Technical specification group radio access network: Study on enhanced LTE support for aerial vehicles," V15.0.0, Dec. 2017.

[3] K. Dorling, J. Heinrichs, G. G. Messier, and S. Magierowski, "Vehicle routing problems for drone delivery," in *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47, no. 1, pp. 70-85, Jan. 2017.

[4] N. Agatz, P. Bouman, and M. Schmidt, "Optimization approaches for the traveling salesman problem with drone," in *Transportation Science*, vol. 52, no. 4, pp. 965–981, 2018.

[5] D. Wang, P. Hu, J. Du, P. Zhou, T. Deng, and M. Hu, "Routing and scheduling for hybrid truck-drone collaborative parcel delivery with independent and truck-carried drones," in *IEEE Internet of Things Journal*, vol. 6, no. 6, pp. 10483–10495, 2019.

[6] S. Choudhury, K. Solovey, M. J. Kochenderfer, and M. Pavone, "Efficient large-scale multidrone delivery using transit networks," *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 4543–4550.

[7] Y. Qin, M. A. Kishk, and M. -S. Alouini, "Stochastic-geometry-based analysis of multipurpose UAVs for package and data delivery," in *IEEE Internet of Things Journal*, vol. 10, no. 5, pp. 4664-4676, March. 2023.

[8] S. R. R. Singireddy and T. U. Daim, "Technology roadmap: Drone delivery-amazon prime air," in *Infrastructure and Technology Management*. New York, NY, USA: Springer, 2018, pp. 387–412.

[9] G. Wikstrom, J. Peisa, P. Rugeland, N. Johansson, S. Parkvall, M. Girnyk, G. Mildh, and I. L. Da Silva, "Challenges and technologies for 6G," in *Proc. 2nd 6G Wireless Summit (6G SUMMIT)*, Mar. 2020, pp. 1–5.

[10] M. Khosravi and H. Pishro-Nik, "Unmanned aerial vehicles for package delivery and network coverage," *2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*, Antwerp, Belgium, 2020, pp. 1-5.

[11] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[12] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, et al., "Human-level control through deep reinforcement learning," in *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.