Q-Learning-Based Dynamic Drone Trajectory Planning in Uncertain Environments

Subrahmanya Chandra Bhamidipati*, Adam Maxwell†, Emily Pham‡, Johann Zhang§,
Zack Murry*, Alicia Esquivel Morel*, Chengyi Qu¶, Sharan Srinivas*, Prasad Calyam*
*University of Missouri, USA. †University of Arkansas, USA. ‡University of Minnesota, USA.
§Tufts University, USA. ¶Florida Gulf Coast University, USA.
Email: * {sb5q6, zjmfrr, ace6qv, srinivassh, calyamp}@missouri.edu, †jam144@uark.edu,
‡pham0579@umn.edu, §johann.zhang@tufts.edu, ¶cqu@fgcu.edu

***Abstract*—Drones are being increasingly used in delivery services, often in coordination with delivery trucks through networked communication links. When a drone loses network communication with the truck in dynamic environments with uncertainties (e.g., obstacles and traffic congestion), navigation adaptation and re-routing are necessary to ensure safety. In this paper, we present a novel Drone Trajectory Planning (DTP) model based on Q-Learning, designed to adapt drone delivery missions in the presence of intermittent network connectivity. Our model leverages state representations including the drone's position, proximity to congestion zones, its general direction and truck/drone network status to adapt to uncertainties. We conduct simulations within a realistic grid environment to evaluate the performance of our DTP model. The results demonstrate DTP model's robust performance in optimizing path decisions and ensuring timely deliveries, even when communication between the truck and drone is lost due to uncertainties, achieving approximately 23% better rewards compared to the state-of-the-art A* path-finding algorithm.**

***Index Terms*—Drone Trajectory Planning, Dynamic Decision-making, Reinforcement Learning, Network Uncertainties**

I. Introduction

The rise of autonomous drones in logistics and delivery services has opened new frontiers in efficient and contactless delivery solutions [1]. These drones have shown great potential in mission-critical applications (e.g., urban last-mile parcel delivery, supply delivery in battlefield situations or in disaster management) by providing timely and precise services [2] [3]. However, ensuring the reliability and safety of these drones, particularly when they lose connection with their control stations on the ground, remains a significant challenge [4].

To address such a challenge, there is a need for dynamic decision-making in the ground control station (GCS) in order to adapt the drone navigation and communication protocols suitably [5]. In cases where navigation adaptations need to be done in real-time due to uncertain environments with obstacles (e.g., trees, buildings) and traffic congestion (e.g., network cross-traffic) as shown in Figure 1, there are more complex challenges. Our framework directly addresses these challenges, ensuring energy-efficient and safe drone operations. There is a dearth of methods that adequately address these challenges during drones' network communication loss with ground control stations, leading to inefficient routing and increased risk of collisions or failed deliveries.

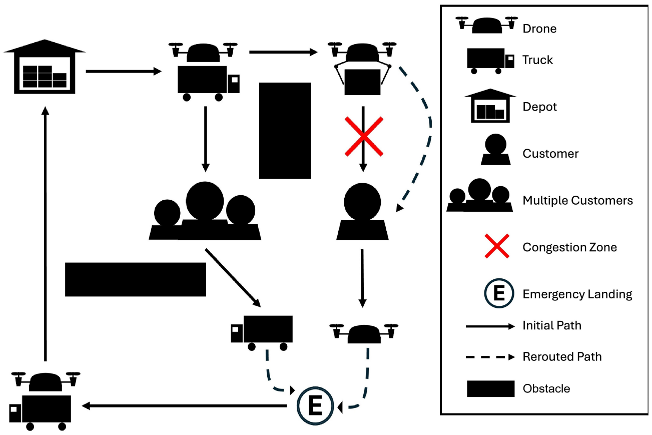

Fig. 1. Illustration of drone trajectory navigation in a dynamic environment with obstacles and congestion zones.

In this paper, we address the problem of drone trajectory planning when it loses network connectivity with the GCS or truck. Specifically, we present a novel Drone Trajectory Planning (DTP) model whose goal is to have the drone determine its next course of action when network communication with the truck is lost. The drone may choose to proceed to the customer's location, divert to an emergency landing spot, or return to the depot as shown in Figure 1. The DTP model uses a Reinforcement Learning (RL) [6]-based approach, with a focus on Q-learning to leverage state representations including the drone's position, proximity to congestion zones, its general direction and truck/drone network status to adapt to uncertainties.

We conduct extensive simulations to evaluate the performance of our DTP model. The simulations feature a realistic grid environment designed to mimic real-world urban settings, complete with various obstacles and congestion zones. The goal of our experiments is to demonstrate the performance of our approach to improve the drone's ability to navigate efficiently (i.e., with low energy consumption) and safely, even when network communication with the delivery truck is lost. Using reward metrics, we illustrate the drone's routing and decision-making process across different scenarios and highlight our DTP model's learning accuracy in comparison with the state-of-the-art A* path-finding algorithm [7].

The remainder of this paper is organized as follows: Section II provides an overview of the related works for drone trajectory planning/adaptation. Section III details the problem formulation relevant to the last-mile parcel delivery involving drone trajectory planning in uncertain environments. Section IV describes the decision-making framework featuring core mechanics of the proposed Q-learning approach to adapt the drone trajectory plan. Section V presents the performance evaluation of our DTP model in extensive simulation scenarios. Finally, Section VI concludes the paper.

II. Related Work

The current state-of-the-art in decision-making for drone trajectory planning primarily involves RL-based approaches [6] due to their ability to optimize decision-making strategies with low human intervention based on long-term rewards in complex environments. In prior works, algorithms such as Q-learning, Deep Q-Networks (DQN), and policy gradient methods have been applied to help drones navigate complex, dynamic environments, focusing on obstacle avoidance, path optimization, and handling real-time environmental changes [8].

Recent research has focused on related problems by considering energy-efficient multi-drone systems and their improved coordination strategies [9]. The study in [6] examines an RL-based solution to balance obstacle avoidance, energy consumption, and coordination between drones. Advances in drone navigation now include deep reinforcement learning (DRL) for real-time terrain mapping and multi-agent coordination [10]. Techniques such as adaptive navigation with interference-aware planning have improved drones' dynamic path optimization by factoring in communication latency and obstacles [11]. Additionally, multi-agent frameworks allow drones to share real-time information for improved decision-making [12]. However, these works assume constant communication of the drone and GCS, and do not consider cases when there is intermittent network connectivity or extended network partitions between the drone and the ground control station due to uncertain network environments.

Industry leaders such as Workhorse as shown in Figure 2 have noted that there is a need for network-aware route management solutions in hybrid truck-drone systems [13]. There have been route management protocols in Mobile Ad-hoc Networks (MANETs) that can maintain connectivity between drones and GCS, but they face challenges with range and delay issues [14]. In contrast, our approach introduces RL-based drone decision-making for situations where network communication between the drone and the truck is lost. Unlike previous models assuming continuous connectivity, our approach enables the drone to make real-time decisions independently on trajectory plans, considering congestion, obstacles, and energy consumption. The integration of network-aware trajectory planning ensures that once communication is restored, drone status updates can be efficiently relayed to the truck and safety can be ensured.

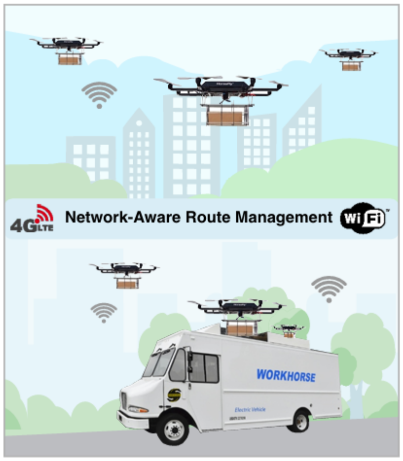

Fig. 2. Network-aware route management between drone and truck in last-mile parcel delivery operations.

III. Problem Formulation

In this section, we present the problem formulation considering a scenario, where drones depend on continuous network communication with the GCS or delivery truck for coordination to achieve last-mile parcel delivery mission success. In the event there is a disruption of network communication due to uncertainties (e.g., obstacles and traffic congestion), we motivate the need for adaptive routing mechanisms that modify the drone trajectory planning. To elaborate, the central problem we address can be broken down into two sub-problems as follows:

A. Identifying Network Uncertainties

The salient research question here is - How can an uncertain network condition manifest to disrupt network communication between a drone and the GCS? Network communication is often disrupted by various factors, leading to uncertainties in the drone's ability to maintain a connection. These uncertainties/disruptions can manifest in the form of:

1) **Physical Obstacles**: As the drone navigates through urban environments, obstacles such as buildings, trees, or other structures may block the communication signals. The Fresnel zone, which defines the region around the direct line-of-sight between the drone and the GCS, is particularly susceptible to signal attenuation from these obstacles and can be calculated with: $F = \sqrt{\frac{\lambda \cdot d_1 \cdot d_2}{d_1+d_2}}$, where F is the Fresnel zone radius, d_1 and d_2 are the distances from the drone to the obstacle and from the obstacle to the GCS, respectively, and λ is the radio signal wavelength. If more than 20% of the Fresnel zone is obstructed, communication interference becomes severe, resulting in failures in data transmission [15]. In such cases, the drone must calculate an alternative route to re-establish communication by avoiding these obstacles.
2) **Network Congestion**: In densely populated urban areas, high network traffic may cause delays and packet loss,

degrading drone-to-GCS communication. The drone's WiFi chipboard detects these issues even within line-of-sight. [16]. Upon detecting these issues, the drone initiates a request to re-establish the packet-forwarding path to maintain the connection with the GCS or truck.

3) **Signal Fading and Distance Attenuation**: As the drone moves further from the GCS or truck, the strength of the communication signal weakens due to distance attenuation. This phenomenon, exacerbated by environmental factors such as adverse weather conditions (rain, fog), may result in intermittent or complete loss of connection. In such cases, the drone dynamically adjusts its route or pauses to wait for reconnection with the GCS or truck.

B. Decision-making for Mission Adaptation

The salient research question here is - How can the drone adapt its mission when it loses connectivity with the GCS or truck? In this case, the drone must decide the best course of action such as proceeding to the customer, returning to the depot or performing an emergency landing to ensure safety. The challenge here for the drone is to balance multiple factors, such as delivery success, operational efficiency, safety, and energy conservation. While mission success is generally defined as reaching the customer and completing delivery, in some scenarios, the primary goal shifts to ensuring the drone's safety and recovering the asset when the original mission plan cannot be executed as intended.

In our framework, we employ Q-learning to guide the drone's decision-making under these network partitioning conditions. The RL algorithm allows the drone to continuously evaluate its environment, including obstacle locations, congestion zones, and remaining energy levels, to decide the best course of action. In cases where reaching the customer would expose the drone to unacceptable risks, such as low energy levels or severe congestion, the RL model prioritizes safety by re-routing the drone to an emergency landing or back to the depot. This approach ensures that mission adaptation may not always result in successful delivery, but it guarantees the drone's safe recovery, protecting the asset. The Q-learning model assigns rewards not only for mission completion but also for making safety-related decisions, such as avoiding obstacles and managing energy efficiently. For example, successfully delivering to the customer results in a high positive reward, whereas entering a congestion zone or running low on battery incurs penalties, reflecting the need for safe operations.

For our experimentation purposes, we assume that the environment can be represented as a grid with static locations for the obstacles, customer, depot, and emergency landing spots; the congestion zones can be dynamically generated to simulate real-world unpredictability. By learning from these dynamic environments, the drone can adapt its navigation to handle complex scenarios, making informed decisions that optimize both delivery success and operational safety when communication is lost, ensuring that drone recovery is prioritized in exceptional circumstances.

Thus, our solution for the problem of RL-based drone navigation is based on addressing the following detailed research questions:

1) How can RL enable drones to adapt dynamically to changing environments in real-time?
2) What RL strategies ensure drones discover optimal or near-optimal navigation policies?
3) How can RL handle large state-action spaces efficiently in complex drone missions?

IV. Decision-Making Framework for Drone Trajectory Adaptation

In this section, we detail the structure and implementation of our Q-learning-based approach for adapting drone navigation in uncertain environments. This framework comprises key components such as the state space, action space, and reward function, which collectively guide the drone's real-time decision-making. Central to this approach is the Drone Trajectory Prediction (DTP) algorithm, built upon Markov Decision Processes (MDP), enabling the drone to predict future states based on current environmental inputs and adapt its trajectory accordingly. Q-learning was selected for its computational efficiency and simplicity, making it ideal for low-power drone hardware where real-time decisions are crucial and deep learning frameworks like DQN may introduce unnecessary overhead. By learning from its environment, the algorithm optimizes the drone's route to ensure safe and efficient navigation, even under conditions of network communication loss. We further discuss the time and space complexity of the DTP algorithm, emphasizing its scalability and potential integration with advanced techniques, such as deep Q-networks (DQN), to enhance performance in large-scale, real-world scenarios.

A. Markov Decision Processes

Applying the concept of MDP enables the drone to predict future states based on real-time inputs, such as obstacles, congestion zones, energy levels, and communication status with the truck. In the event of network communication loss, the drone independently decides whether to proceed with the delivery, return to the depot, or land at an emergency location.

The state space represents all possible drone states, including position, heading, energy levels, and proximity to obstacles. The action space defines the set of actions the drone can take, such as adjusting speed, changing direction, or avoiding obstacles. Transition probabilities model environmental uncertainties, and the reward function assigns values to actions based on safety, efficiency, and delivery success.

By thus applying MDP, we can optimize drone routes in real-time, maintaining high performance despite disconnections or environmental challenges. The simulations in grid environments that we present in Section V utilize MDP and RL to demonstrate how the drone can adapt and make reliable decisions, ensuring efficient operation in dynamic environments.

As part of modeling our dynamic environment, herein we detail our state space, action space, reward function and the policy specific to our drone's decision-making.

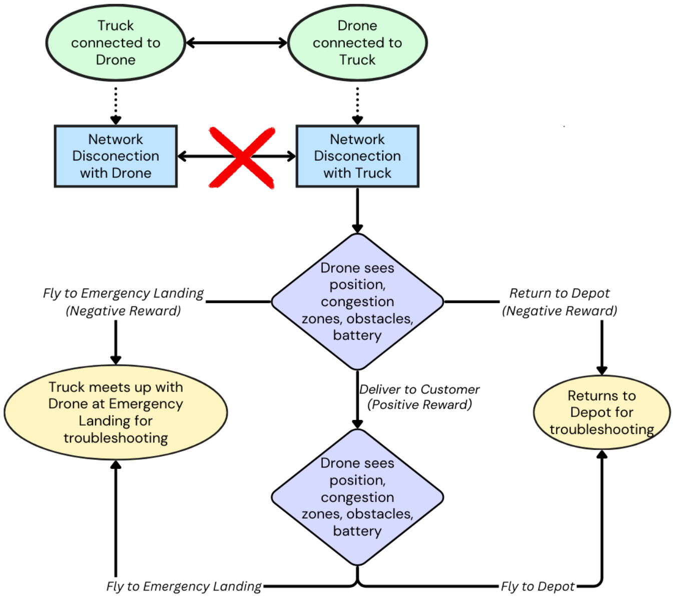
Truck connected to Drone
Drone connected to Truck
Network Disconection with Drone
Network Disconection with Truck
Drone sees position, congestion zones, obstacles, battery
Fly to Emergency Landing (Negative Reward)
Return to Depot (Negative Reward)
Deliver to Customer (Positive Reward)
Truck meets up with Drone at Emergency Landing for troubleshooting
Returns to Depot for troubleshooting
Drone sees position, congestion zones, obstacles, battery
Fly to Emergency Landing
Fly to Depot

Fig. 3. Drone decision making framework in response to network disconnection with truck

B. State (S) and Action (A) Space

The state space S represents the set of all possible states that the drone can occupy during its operation, encapsulating the necessary information about both the environment and the operational parameters. At any given time t, the state s_t is expressed as:

$$s_t = \{P_D(n),\, \phi_t,\, E_t^n,\, L_d,\, C_z,\, D_{Cz}\}$$

Here, $P_D(n)$ refers to the drone's current position, given by its coordinates in a two-dimensional grid. The variable ϕ_t denotes the heading of the drone, representing the direction in which the drone is moving at time t. The term E_t^n indicates the total operational time remaining for the drone, which is a measure of its battery capacity since the last recharge or reset. Additionally, L_d corresponds to the coordinates of the drone's intended delivery location, representing the target point in its mission.

The state also includes a boolean variable C_z, which indicates whether the drone is currently within a congestion zone, a region that may impose restrictions on its movement or increase operational difficulty. Finally, D_{Cz} represents the direction of the nearest congestion zone within a 100-pixel radius, providing a guide for the drone to avoid these areas and optimize its trajectory.

For example, at a particular time t', the state could be represented as:

$$s_t' = \{P_D(n) = (10, 20),\, \phi_t = 45^\circ,\, E_t^n = 80\,\text{minutes}, \\ L_d = (50, 60),\, C_z = \text{false},\, D_{Cz} = \text{East}\}$$

In this instance, the drone is located at coordinates $(10, 20)$ and is heading in a direction of 45°, with 80 minutes of operational time remaining based on its battery capacity. The delivery location is specified at coordinates $(50, 60)$. The drone is not currently in a congestion zone, and the nearest congestion zone is located to the east of its current position.

The action space A represents the set of all possible actions that the drone can take while operating in a given state. These actions allow the drone to interact with its environment and modify its trajectory and velocity. The available actions include adjusting the drone's movement direction and speed.

Initially, the drone can move in any of the eight cardinal and intercardinal directions, allowing for precise adjustments to its position in the 2D grid. In addition to directional movement, the drone can modify its velocity by either accelerating or decelerating. Speed adjustments are critical for optimizing energy consumption and ensuring safe navigation through various environmental conditions. Specifically, the drone has the option to increase its speed, denoted as a "speed up" action, or decrease its speed, described as a "speed down" action.

Thus, the action set A consists of movement in eight directions, increasing speed, and decreasing speed, which together provide a comprehensive range of options for the drone to efficiently navigate its environment.

C. Reward Function (R) and Derived Policy (P)

The reward function R quantifies the benefit or penalty associated with each action, guiding the drone to maximize cumulative rewards. A significant positive reward α is granted for reaching the customer, while a smaller reward β is given for reaching an emergency landing site or depot. Negative rewards are applied for undesirable actions: the drone incurs a penalty γ for traversing congestion zones, and a larger penalty δ for collisions or battery depletion. Movement incurs a small penalty proportional to velocity, $-0.0005 \times (|v_x| + |v_y|)$.

This reward structure balances objectives, with greater weight on avoiding collisions and conserving energy. Positive rewards for key locations ($\alpha \gg \beta$) prioritize deliveries, while larger penalties for collisions and depletion ($\delta \gg \gamma$) promote safe and efficient navigation. The exact values for these rewards and other hyperparameters are listed in Table I.

The policy P derived from our Q-learning algorithm equips the drone with the ability to navigate and make decisions, even in scenarios where communication with the delivery truck is disrupted. This policy is designed to optimize cumulative rewards by guiding the drone's actions based on environmental inputs, ensuring efficient and safe operation.

A key component of the policy is its focus on *safe navigation*, where the drone actively avoids congested areas and obstacles when network connectivity is lost. The policy also incorporates *energy-aware decision-making*, directing the drone to return to a depot or emergency landing site if its remaining battery life becomes critical. Furthermore, the policy emphasizes *operational efficiency*, enabling the drone to continue making optimized decisions despite communication disruptions, ensuring both safety and mission success. This learned policy allows the drone to operate, maintaining its objectives even in unpredictable or uncertain conditions, thus ensuring reliable performance across various intermittent network connectivity scenarios with the GCS or truck.

D. Time and Space Complexity

In analyzing our RL-based algorithm for drone trajectory prediction, it is essential to evaluate its time and space complexity to ensure scalability and efficiency in real-world last-mile delivery scenarios. Understanding these complexities helps identify performance bottlenecks and optimize the algorithm for large-scale applications. To fully assess the performance of our algorithm, it is essential to evaluate its efficiency across different scales of complexity, particularly in large-scale environments. Approaches such as DQN offer promising solutions to balance computational demands and memory requirements, allowing the algorithm to scale while maintaining robust performance in complex last-mile delivery scenarios.

Specifically, the time complexity of Q-learning is $O(|S| \times |A| \times N)$, where $|S|$ and $|A|$ represent the state and action spaces, and N is the number of episodes required for convergence. In our scenario, the state space includes the drone's position, battery, obstacles, congestion zones, and network status, while the action space covers movement decisions and handling network issues. As the environment complexity increases, so does the computational demand. Others, the space complexity is $O(|S| \times |A|)$, as Q-values are stored for each state-action pair. The memory requirements grow with more complex environments, but discretization and methods such as deep Q-networks (DQN) can help reduce memory usage by approximating Q-values rather than storing them explicitly.

V. Performance Evaluation

In this section, we evaluate the performance of our proposed Q-learning-based approach for drone navigation. We conduct detailed simulations in a dynamic grid environment to assess the drone's ability to make real-time decisions under varying conditions, including obstacle avoidance, congestion zones, and network communication loss. We compare the performance of the Q-learning algorithm with the best path-finding A* algorithm focusing on reward accumulation and decision-making efficiency. Through scenario-based analyses, we demonstrate how the Q-learning model optimizes drone routing and ensures mission completion even in unpredictable and complex environments.

A. Grid Environment Setup

The environment is a static 800x800 pixel grid representing the drone's operating area. It includes predefined obstacles, congestion zones, a start position, goal position, emergency spots, and a depot. The maze generation algorithm ensures a consistent layout with both obstacles and clear paths, simulating a controlled environment.

Grid Description: The grid, with 20x20 pixel units, simplifies the action space and provides challenges to the drone through a fixed configuration of obstacles.

Trajectory Layout: The maze-like static layout ensures designated paths between the start, customer location, emergency landing spots, and the depot, replicating structured routes in drone delivery and navigation scenarios.

Congestion Zones: These zones are initially generated randomly and dynamically updated during the simulation to simulate areas with increased navigation difficulty, such as high network traffic or environmental hazards.

B. Simulation and Learning

The simulation runs for 100,000 episodes with optimized learning parameters (alpha, gamma, epsilon decay) to facilitate effective learning. Pygame is used to visualize the drone's movements, rewards, and Q-values in real time.

Simulation Parameters: The large number of episodes ensures sufficient exploration, with the learning rate, discount factor, and epsilon decay guiding the learning process.

TABLE I
HYPERPARAMETERS FOR Q-LEARNING SIMULATIONS

Hyperparameter	**Value**
Environment Dimensions	800x800
Grid Dimensions	20x20
Learning Rate (α)	0.2
Discount Factor (γ)	0.95
Episodes	100,000
Speed	1 to 20 (variable)
Congestion Zone Size	100x100
Congestion Timer Range	50 to 200 steps
Obstacle Sizes	400x100, 100x200, 100x100
Reward for Reaching Customer	700
Reward for Reaching Depot	50
Reward for Reaching Emergency Landing Spot	50
Penalty for Obstacle collision	-100
Penalty for Congestion Zone	-30
Penalty Proportional to Velocity	-0.0005 * (abs(vx) + abs(vy))

Learning Process: During each episode, the drone selects an action based on the epsilon-greedy strategy, moves to a new state, receives a reward, and updates the Q-table. This process continues until the drone reaches the goal, an emergency spot, or the depot, or until a maximum number of steps is reached.

C. Epsilon Decay

The epsilon decay graph illustrates how the exploration rate decreases over time. As ϵ decreases, the drone relies more on its learned policy rather than exploring new actions. The epsilon decay used in Q-learning is:

$$\epsilon = \max\left(0.15, 1 - \frac{\textit{episode}}{0.75 \times \textit{total\_episodes}}\right)$$

where ϵ is the exploration rate, *episode* is the current episode, and *total_episodes* is the total training duration. This decay helps the drone balance exploration and exploitation, ensuring it transitions to the learned optimal policy while maintaining some exploration.

D. Experimental Results: Scenario-Based Decision Making

In this section, we highlight different scenarios where the drone makes real-time decisions based on its environment, showcasing the robustness of the Q-learning algorithm. Each scenario demonstrates the drone's ability to adapt dynamically. Specifically, in *Scenario 1*, we explore situations where, after delivering to the customer, the drone must decide between two post-delivery options: proceeding to the depot (Scenario 1a) or to an emergency landing spot (Scenario 1b). These scenarios emphasize the drone's capacity for decision-making, ensuring efficient mission completion even under varying conditions.

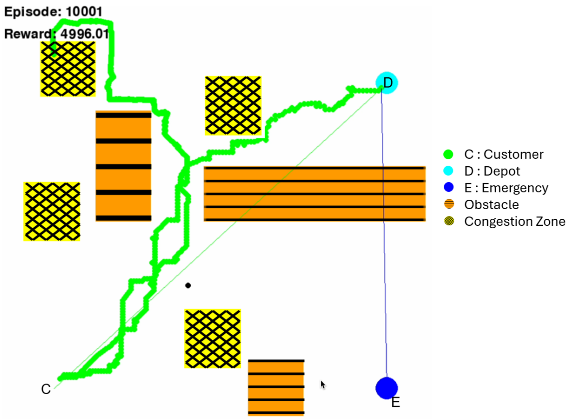

Fig. 4. Real-time decision-making by the drone, returning to the depot

1) Scenario 1a: Drone Returns to Depot after Delivery: Following the delivery, the drone makes the strategic decision to return to the depot, as depicted in Figure 4. This decision highlights the drone's ability to independently choose the most efficient post-mission course of action. By returning to the depot, the drone ensures it is ready for future missions, maintaining operational readiness.

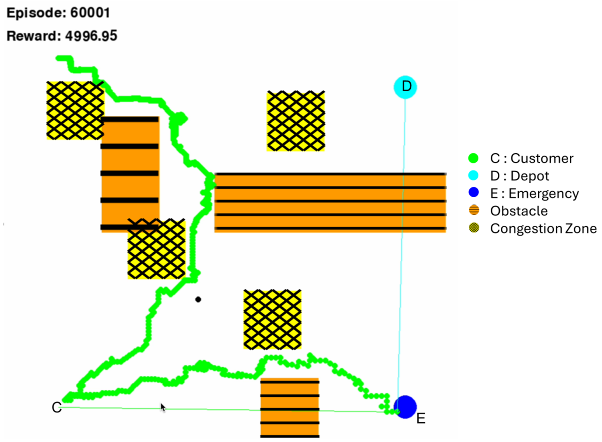

Fig. 5. Real-time decision-making by the drone, reaching the emergency landing spot

2) Scenario 1b: Drone Proceeds to Emergency Landing after Delivery: After successfully delivering to the customer, the drone demonstrates its flexibility by quickly adapting to an emergency situation. As shown in Figure 5, the drone shifts its focus to a safe landing procedure, prioritizing safety after mission completion. This decision illustrates the drone's ability to respond to unforeseen events, such as low battery or emergency airspace restrictions.

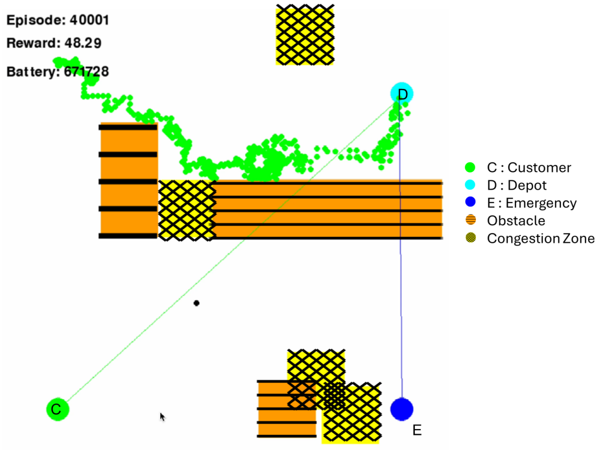

Fig. 6. Drone dynamically avoiding congestion zones, by not delivering to customer and returning to depot

3) Scenario 2: Drone Returns to Depot without Delivering: In some instances, the drone determines that it is more efficient to return directly to the depot without completing the delivery. Figure 6 illustrates the drone's decision to avoid a heavily congested area. Rather than attempting a delivery in suboptimal conditions, the drone re-routes and heads back to the depot, shown in cyan. This scenario highlights the Q-learning algorithm's ability to prioritize drone safety over task completion by ensuring the drone does not operate in dangerous or inefficient conditions, thereby conserving its energy and minimizing the risk of collision or other hazards in highly congested environments.

E. Discussion

The above experiment scenarios illustrate the adaptability of the Q-learning algorithm in helping with real-time decision-making in the drone. In *Scenario 1*, the drone successfully completes its delivery objective while handling potential post-delivery options such as emergency landings and returning to the depot. In *Scenario 2*, the drone demonstrates the flexibility to abandon a customer delivery in favor of returning to the depot to avoid hazards caused by congestion.

In addition to the adaptability demonstrated by the Q-learning algorithm in these scenarios, a comparative analysis with the state-of-the-art A* algorithm [7] reveals further advantages. When comparing the rewards accumulated by the drone using Q-learning versus A* shown in Figure 7, the Q-learning approach outperformed A* by approximately 23%. This improvement highlights the strength of RL in environments where unpredictability and real-time decision-making are key factors. While A* is effective in static, well-defined path-finding tasks, it struggles with dynamic elements such as congestion and dynamic obstacles. In such cases, the

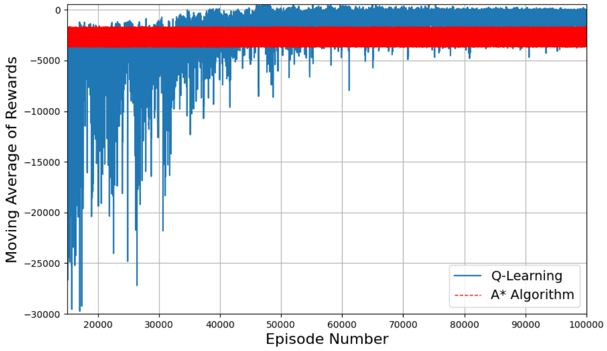

Fig. 7. Comparison of Rewards between Q Learning and A* Algorithms

Q-learning excels by continuously adapting to new environmental states and optimizing decision-making based on past experiences. This performance in reward accumulation demonstrates the potency of Q-learning in complex and dynamic environments, making it a promising solution for real-world drone navigation challenges in the presence of intermittent network connectivity conditions.

VI. Conclusion

‘

Future work can involve exploration of advanced RL algorithms, such as Deep Q-Networks (DQN) or Proximal Policy Optimization (PPO), which could better handle complex environments and uncertainties. Testing the framework in real-world testbeds can further validate the effectiveness of our solution approach.

Acknowledgments

This research was supported by the National Science Foundation (NSF) under Award Nos. CNS-2313887 and CNS-1647182, and by the National Security Agency (NSA) Award No. H98230-23-1-0238. The authors express their sincere gratitude for the funding provided through the NSF PFI-TT (Partnerships for Innovation Technology Translation) program. The views presented are those of the authors and do not necessarily reflect the views of the NSF, or the NSA.

References

[1] G. Brunner, B. Szebedy, S. Tanner, and R. Wattenhofer, “The urban last mile problem: Autonomous drone delivery to your balcony,” in *2019 international conference on unmanned aircraft systems (icuas)*. IEEE, 2019, pp. 1005–1012.

[2] H. D. Yoo and S. M. Chankov, “Drone-delivery using autonomous mobility: An innovative approach to future last-mile delivery problems,” in *2018 ieee international conference on industrial engineering and engineering management (ieem)*. IEEE, 2018, pp. 1216–1220.

[3] C. Qu, J. Boubin, D. Gafurov, J. Zhou, N. Aloysius, H. Nguyen, and P. Calyam, “Uav swarms in smart agriculture: Experiences and opportunities,” in *2022 IEEE 18th International Conference on e-Science (e-Science)*, 2022, pp. 148–158.

[4] K. Dorling, J. Heinrichs, G. G. Messier, and S. Magierowski, “Vehicle routing problems for drone delivery,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47, no. 1, pp. 70–85, 2016.

[5] C. Qu, P. Calyam, J. Yu, A. Vandanapu, O. Opeoluwa, K. Gao, S. Wang, R. Chastain, and K. Palaniappan, “Dronecoconet: Learning-based edge computation offloading and control networking for drone video analytics,” *Future Generation Computer Systems*, vol. 125, pp. 247–262, 2021.

[6] C. Qu, R. Singh, A. E. Morel, F. B. Sorbelli, P. Calyam, and S. K. Das, “Obstacle-aware and energy-efficient multi-drone coordination and networking for disaster response,” pp. 446–454, 2021.

[7] P. E. Hart, N. J. Nilsson, and B. Raphael, “A formal basis for the heuristic determination of minimum cost paths,” *IEEE transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, 1968.

[8] J. Silva *et al.*, “Reinforcement learning framework for autonomous drone navigation,” *Journal of Autonomous Systems*, 2018.

[9] A. Grote, E. Lyons, K. Thareja, G. Papadimitriou, E. Deelman, A. Mandal, P. Calyam, and M. Zink, “Flypaw: Optimized route planning for scientific uavmissions,” in *2023 IEEE 19th International Conference on e-Science (e-Science)*. IEEE, 2023, pp. 1–10.

[10] L. Yang *et al.*, “Optimizing drone delivery routes using q-learning,” *Urban Systems Journal*, 2019.

[11] S. Lee and J. Kim, “Safe landing strategies for drones using reinforcement learning,” *Safety Systems Review*, 2020.

[12] J. Westheider, J. Rückin, and M. Popović, “Multi-uav adaptive path planning using deep reinforcement learning,” in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2023, pp. 649–656.

[13] Workhorse Group Inc., “Workhorse - innovation driving change,” 2024, accessed: 2024-09-15. [Online]. Available: https://workhorse.com/

[14] C. Qu, F. B. Sorbelli, R. Singh, P. Calyam, and S. K. Das, “Environmentally-aware and energy-efficient multi-drone coordination and networking for disaster response,” *IEEE transactions on network and service management*, vol. 20, no. 2, pp. 1093–1109, 2023.

[15] C. Qu, “Intelligent orchestration of computation and networking for drone swarm applications,” Ph.D. dissertation, University of Missouri-Columbia, 2023.

[16] R. R. Ramisetty, C. Qu, R. Aktar, S. Wang, P. Calyam, and K. Palaniappan, “Dynamic computation off-loading and control based on occlusion detection in drone video analytics,” in *Proceedings of the 21st International Conference on Distributed Computing and Networking*, 2020, pp. 1–10.