©2026

ERIC J. KNOBLOCK

ALL RIGHTS RESERVED

SPECTRUM OPTIMIZATION FOR ADVANCED AIR MOBILITY USING MACHINE LEARNING METHODS

A Dissertation

Presented to

The Graduate Faculty of The University of Akron

In Partial Fulfillment

of the Requirements for the Degree

Doctor of Philosophy

Eric J. Knoblock

May, 2026

SPECTRUM OPTIMIZATION FOR ADVANCED AIR MOBILITY USING MACHINE LEARNING METHODS

Eric J. Knoblock

Dissertation

Approved:

Advisor
Dr. Nghi H. Tran

Committee Member
Dr. Rafael D. Apaza

Committee Member
Dr. Yan (Mia) Zhang

Committee Member
Dr. Chen Ling

Committee Member
Dr. Shengyong Wang

Accepted:

Department Chair
Dr. Yilmaz Sozer

Dean of the College of Engineering and Polymer Science
Dr. Craig Menzemer

Senior Director, School of Graduate Studies
Dr. Joseph Wilder

Date

ABSTRACT

The emergence of new aerial vehicles into the airspace and the continued growth of aviation operations will result in an increased demand for voice and data communications throughout the National Airspace System (NAS). However, the continued use of existing aviation spectrum allocations will not meet the anticipated needs of future aerospace applications; as a result, a new approach to aviation spectrum management is needed to support the forecasted needs of new airspace users. The National Aeronautics and Space Administration (NASA) is investigating the use of recent advancements in artificial intelligence (AI) and machine learning (ML) to implement a novel spectrum management approach that allows for the intelligent utilization of aviation spectrum throughout the airspace. This thesis investigates the use of ML and AI to jointly manage communications channel selection and airspace resources to improve and optimize the overall frequency spectrum utilization in the airspace. The technical investigation considers various airspace communications configurations, including air-ground and air-air communications, and can be applied to the existing air traffic control (ATC) system and future applications, such as the emerging Advanced Air Mobility (AAM) initiative. To support the evaluation of the proposed concepts and technologies, a modeling and simulation capability has been

developed, which will continue to evolve in support of new and advanced airspace applications. It is anticipated that this proposed spectrum management concept will allow for increased spectrum utilization efficiency and enhanced airspace capacity, which will better serve the needs of future airspace applications.

ACKNOWLEDGEMENTS

I would like to express my gratitude to Dr. Nghi Tran and Dr. Hamid Bahrami for their guidance and support throughout my studies at the University of Akron. I am also deeply thankful to the members of my dissertation committee—Dr. Yan (Mia) Zhang, Dr. Chen Ling, and Dr. Shengyong Wang—for their thoughtful feedback and constructive suggestions.

Special thanks go to my colleagues and collaborators at NASA, particularly Dr. Rafael Apaza, whose expertise, dedication, and mentorship greatly enriched this work. I am also especially grateful to Dr. Michael Gasper for his significant contributions to the success of this effort. I would also like to express gratitude to my supervisors, Ms. Dawn Emerson and Dr. Félix Miranda, for being supportive of my desire to complete my dissertation. This research was made possible through the support of NASA's Transformational Tools and Technologies project, for which I am sincerely appreciative.

Finally, I extend my sincerest appreciation to my family for their support and patience throughout this journey—most especially my wife, Angela; my children, Aedan and Elexa; and my mother, Judy.

TABLE OF CONTENTS

Page

LIST OF TABLES ix

LIST OF FIGURES x

CHAPTER

I. INTRODUCTION 1

1.1 Background and Research Motivation 1

1.2 Research Objectives and Contributions 2

1.3 Related Spectrum Studies 5

1.4 Dissertation Outline 8

II. SPECTRUM MANAGEMENT APPROACH 9

2.1 Autonomous Spectrum Concept 9

2.2 Resource Allocation 11

2.3 Prediction Methods 14

2.4 Modeling and Simulation Tool Development 17

2.5 Open Challenges 19

2.6 Conclusions and Follow-on Studies 20

III. RESOURCE ALLOCATION USING GRAPH NEURAL NETWORKS . 21

3.1 Overview and Methods 21

3.2 Problem Simplification 22
3.3 System Model and Problem Formulation 23
3.4 Resource Allocation via Graph Coloring 31
3.5 Simulations 39
3.6 Observations 48
IV. RESOURCE ALLOCATION USING REINFORCEMENT LEARNING . 52
4.1 System Model and Problem Formulation 53
4.2 Reinforcement Learning Solution 58
4.3 Multi-Agent PPO 60
4.4 QMIX 64
4.5 Simulation and Case Study 69
4.6 MAPPO Performance Analysis 72
4.7 QMIX Performance Analysis 82
4.8 MAPPO and QMIX Analysis 87
4.9 Observations and Potential Follow-on Studies 92
V. SUMMARY AND CONCLUSIONS 94
5.1 Summary of Research 94
5.2 Key Findings 95
5.3 Conclusions 96
5.4 Future Work 96
BIBLIOGRAPHY 98
APPENDICES 111

APPENDIX A. MODELING AND SIMULATION TOOL DEVELOPMENT 112

APPENDIX B. ACRONYM LIST 119

LIST OF TABLES

Table		Page
3.1	Summary of simulation parameters for the Urban Air Mobility resource allocation study. The table lists air traffic characteristics, airspace design details, and communication link settings used in the experiments.	42
3.2	Summary of interference events for each allocation method. The table reports total events, mean events per timestep, and standard deviation for greedy, dynamic greedy, and GCN-based approaches.	45
4.1	Summary of key notation used in the reinforcement learning system model. The table defines variables for aircraft positions, velocities, channel assignments, and other parameters relevant to optimization.	55
4.2	Configuration parameters for the Urban Air Mobility simulation environment. The table lists vertiport coordinates, base station positions, communication settings, and propagation characteristics used in the study.	71
4.3	Flight plans for all aircraft in the simulation scenario. Each entry specifies the departure and destination vertiports for evaluating multi-agent resource allocation.	72
4.4	Hyperparameter settings for the MAPPO algorithm. The table includes values for episodes, minibatch size, training iterations, learning rate, and other parameters used in reinforcement learning experiments.	73
4.5	Hyperparameter settings for the QMIX algorithm. The table lists values for episodes, batch size, learning rate, discount factor, and other parameters used in cooperative multi-agent training.	82

LIST OF FIGURES

Figure		Page
1.1	Conceptual diagram of an Urban Air Mobility (UAM) scenario in the Cleveland metropolitan area. The figure shows multiple aircraft traveling between designated vertiports along defined air corridors. Each aircraft maintains two-way radio communication with ground-based base stations to ensure safe operations and data exchange. This example illustrates the network structure and communication links that support UAM operations.	3
2.1	Functional flow of the autonomous spectrum management concept. The diagram shows how prediction and learning modules process air traffic, communication demands, and airspace constraints to allocate spectrum resources dynamically. This approach enables efficient and adaptive frequency management for aviation communications.	10
2.2	Example of an airspace represented as a graph for resource allocation. Aircraft are modeled as nodes, and edges indicate potential interference between aircraft operating in close proximity. This representation supports graph-based algorithms for channel assignment.	12
2.3	Illustration of an airspace scenario modeled as a Markov game for reinforcement learning. Each agent represents an aircraft that selects actions such as channel or speed adjustments based on its state and interactions with other agents.	14
2.4	Screenshot of the custom modeling and simulation tool visualizing Cleveland airspace. The tool displays aircraft positions, communication links, and sector boundaries, enabling evaluation of spectrum allocation strategies in realistic scenarios.	18
3.1	System model for an Urban Air Mobility scenario. The diagram shows aircraft traveling between vertiports and communicating with ground-based base stations. It illustrates the network structure used for channel assignment and interference analysis.	24

3.2 Air-to-ground interference model used to construct the interference graph. The figure depicts how distances between aircraft and base stations influence co-channel interference calculations for spectrum allocation. 25

3.3 Example of an interference graph for a four-aircraft scenario. Nodes represent aircraft, and edges indicate potential interference relationships based on spatial proximity and frequency reuse constraints. . . . 31

3.4 Architecture of the graph convolutional network (GCN) used for channel assignment. The model includes two TAGConv layers with ReLU and Softmax activations, transforming the input adjacency matrix into a probability matrix for color (channel) assignments. 38

3.5 Training process for the graph convolutional network. The adjacency matrix is processed through the network to generate channel assignment probabilities. The loss function guides weight updates using the Adam optimizer during backpropagation. 39

3.6 Comparison of training performance for two GCN variants at simulation timestep 55. The plot shows how the loss function decreases across epochs for TAGConv and SAGEConv implementations. 40

3.7 Visualization of the simulation scenario for Urban Air Mobility resource allocation. The figure shows aircraft trajectories, airspace sectors, and base station locations used in the evaluation of spectrum management strategies. 41

3.8 Process flow for the simulation framework. The diagram outlines steps for generating airspace states, applying allocation algorithms, and evaluating interference performance. 44

3.9 Interference events observed at each simulation timestep for different allocation methods. The plot compares greedy allocation, dynamic greedy, and two GCN-based approaches (GNN-Train100 and GNN-Train1). 45

3.10 Cumulative interference events across all simulation timesteps for each allocation method. Lower cumulative values indicate better spectrum allocation performance. 46

3.11 Comparison of graph complexity between full airspace and simplified predicted communication scenarios. The plot shows how prediction-based simplification reduces the number of edges, lowering potential interference. 47

3.12 Channel assignment results for three consecutive timesteps during peak airspace congestion. The figure illustrates how dynamic greedy coloring minimizes conflicts compared to other methods. 51

4.1 System model for an Urban Air Mobility scenario. The diagram shows aircraft traveling between vertiports and communicating with base stations using shared frequency channels, forming the basis for reinforcement learning-based resource allocation. 54

4.2 Architecture of the QMIX algorithm integrated with V-D3QN agent networks. The figure illustrates how individual agent Q-values are combined through a mixing network to produce a joint Q-value for cooperative decision-making. 66

4.3 Diagram of the Urban Air Mobility environment used for reinforcement learning simulations. The figure shows aircraft, vertiports, and base stations arranged to evaluate multi-agent resource allocation strategies. 70

4.4 Flight corridor profile for the route between vertiports v_6 and v_2. The figure depicts trajectory distance and altitude used in simulation experiments. 70

4.5 Performance comparison of different minibatch sizes in MAPPO training. The figure shows convergence trends for batch sizes 2, 4, 6, and 8 across training episodes. 74

4.6 Effect of training iteration count on MAPPO convergence. The figure compares performance for 5, 10, 15, and 20 iterations, highlighting optimal settings for faster learning. 75

4.7 Accumulated reward trends for MAPPO under three SINR thresholds (2.5, 4.5, and 6.5). The figure illustrates how stricter SINR requirements affect learning speed and convergence. 77

4.8 Frequency allocation and reuse patterns for MAPPO under SINR threshold 6.5 and minimum separation 100 units. The figure shows how channels are reassigned during flight to maintain communication quality. 78

4.9 Detailed view of frequency reuse at time step 13 for SINR threshold 6.5. The figure highlights multiple aircraft sharing the same channel without causing harmful interference. 79

4.10 Impact of path loss exponent on MAPPO learning performance. The figure compares convergence for δ values of 2.0, 2.4, and 2.8 under SINR threshold 4.5 and minimum separation 300 units. 81

4.11 Convergence performance of QMIX under three SINR thresholds (2.5, 4.5, and 6.5). The figure shows how stricter SINR requirements increase training complexity and convergence time. 83

4.12 Frequency allocation and reuse patterns for QMIX under SINR threshold 6.5 and minimum separation 100 units. The figure illustrates how channels are managed to maintain communication quality. 85

4.13 Velocity adjustment strategies used by QMIX to mitigate interference under SINR threshold 6.5 and minimum separation 100 units. The figure shows how speed control supports spectrum efficiency. . . . 86

4.14 Effect of path loss exponent on QMIX convergence. The figure compares training performance for different propagation conditions under SINR threshold 4.5 and minimum separation 300 units. 88

4.15 Comparison of SINR convergence between MAPPO and QMIX under SINR threshold 4.0. The figure highlights MAPPO's faster learning and superior sample efficiency. 90

4.16 Comparison of MAPPO and QMIX performance under a separation constraint of 250 units. The figure shows how both algorithms manage interference and maintain safe distances, with MAPPO converging significantly faster. 91

A.1 Architecture diagram of the modeling and simulation platform. The figure shows modular components and their connections to external interfaces, illustrating how the system integrates prediction, resource management, and visualization for spectrum analysis. 113

A.2 Screenshot of the modeling and simulation tool interface built on NASA WorldWind. The interface displays airspace sectors, aircraft positions, and communication links, enabling interactive configuration and visualization of spectrum management scenarios. 115

A.3 Example visualization of airspace with air-to-ground link analysis results. The figure highlights aircraft positions, ground stations, and calculated signal-to-interference ratios for evaluating communication performance. 116

A.4 Radiation pattern of a dipole antenna used at a ground station. The figure illustrates directional gain characteristics that influence link performance in spectrum management simulations. 117

CHAPTER I

INTRODUCTION

1.1 Background and Research Motivation

The Advanced Air Mobility (AAM) concept is an initiative that facilitates emerging aviation markets to develop air transportation capabilities for the movement of people and cargo throughout regions of airspace not previously served by traditional aviation operations. The AAM ecosystem intends to serve users in local, regional, intra-regional, and urban environments using advanced new aircraft that use low-noise electric propulsion and possess the capability of vertical takeoffs and landings. Further, the AAM community is investigating novel air traffic management approaches that employ recent technological advances in automation and computer networking to enable a more collaborative and information-rich environment.

Within the AAM framework, Urban Air Mobility (UAM) is a subset that focuses on air transportation within the urban environment. Fig. 1.1 illustrates a conceptual Urban Air Mobility (UAM) scenario in the Cleveland metropolitan area, where multiple aircraft depart from vertiports, navigate designated air corridors, and maintain communication with ground-based systems via radio links to base stations.

The complexity of air operations in the AAM environment will require diverse air-to-ground and air-to-air communications capabilities to ensure safety and crucial information distribution for users within the airspace [1]. The communications systems of the new aerial users must seamlessly interconnect to other aircraft and ground assets for the timely exchange of data for ensured safe travel throughout varying conditions and environments. With the growth of aviation operations due to AAM other new airspace entrants, there will be an increasing demand for aviation spectrum to support the aeronautical communications demands. However, the existing VHF/UHF frequency allocations for NAS operations have become increasingly scarce and will be unable to support the projected growth in demand [2]. While traditional approaches to address the spectrum scarcity issue (such as reduction of channel bandwidth) have provided some relief, there is a need for a more scalable and sustainable solution.

1.2 Research Objectives and Contributions

To address the spectrum challenge, a team of researchers at the NASA Glenn Research Center, in support of the agency's Aeronautics Research Mission Directorate, is investigating methods to modernize the management of aviation spectrum through the implementation of advanced concepts in communications, networking, artificial intelligence (AI), and machine learning (ML). Specifically, this research investigates the use of ML and AI to jointly manage communications channel selection and airspace resources to improve and optimize frequency utilization. The

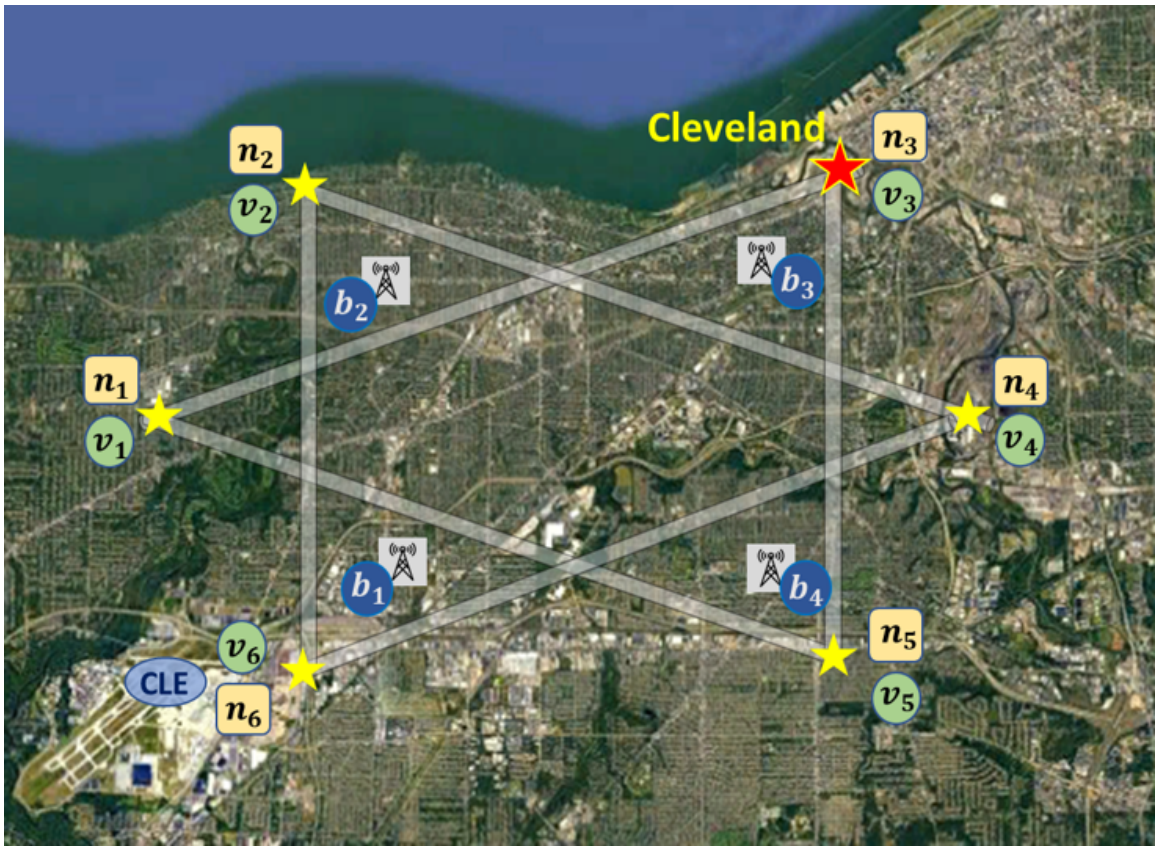


Figure 1.1: Conceptual diagram of an Urban Air Mobility (UAM) scenario in the Cleveland metropolitan area. The figure shows multiple aircraft traveling between designated vertiports along defined air corridors. Each aircraft maintains two-way radio communication with ground-based base stations to ensure safe operations and data exchange. This example illustrates the network structure and communication links that support UAM operations.

outcomes from this research, along with other technological advancements, are expected to significantly improve the safety and efficiency of future aviation operations and enable future operational concepts, including remote piloted aircraft operation and autonomous flight.

This work is being conducted in three inter-related research and development areas. The first research area focuses on the development of algorithms to predict

air traffic and communications demands throughout the airspace. Based on these predicted demands and other inputs, the second research area focuses on learning-based resource allocation for aeronautical communications. In general, use cases involving both air-to-ground and air-to-air communications are considered, where the optimization problems are mathematically formulated and solved by learning-based algorithms. Finally, the third research area focuses on the development of a modeling and simulation capability that enables the evaluation of the proposed spectrum concepts through the implementation and use of graphical, mathematical and data analysis tools. This simulation environment allows researchers to evaluate the performance of the integrated functionality of demand prediction and learning-based resource allocation using realistic airspace operational scenarios to assess concept feasibility, accuracy, scalability, and efficiency. It is anticipated that the outcomes of this study will meet the spectrum needs of future airspace communications, including the existing air traffic control system as well as future AAM, high-altitude Class E airspace communications [3], and others. The primary contributions of this research effort are as follows:

- Explore spectrum allocation for UAM to devise a new management approach, utilizing elements of AI and ML for dynamic frequency management and assignment.
- Formulate mathematical models of airspace scenarios and solve the channel allocation and resource management problem using machine learning-based methods. Specifically, graph convolutional neural networks (GCN) were used to

solve the channel allocation problem in a centralized fashion, and multi-agent reinforcement learning (MARL) was used in a decentralized fashion to address the joint problem of channel allocation and velocity selection. Using MARL, the intent was to minimize flight completion time while meeting communications quality-of-service and aircraft separation requirements.

- Propose an integrated evaluation framework for both frequency and airspace resources to optimize airspace operations. This transition from isolated resource management enables a holistic assessment that improves system-wide resource utilization, enhancing the efficiency of frequency usage.

1.3 Related Spectrum Studies

A substantial body of literature exists on spectrum studies across various wireless communication networks, encompassing theoretical capacity analysis, interference management, protocol and algorithm design, and system-level validation. In recent years, the resurgence of AI has significantly influenced many scientific and technological domains, including wireless communications and networking. This convergence has given rise to a learning-based wireless communication paradigm, where ML algorithms—such as supervised, unsupervised, and reinforcement learning (RL)—are deployed at different nodes within communication networks [4, 5, 6].

Supervised learning has been applied in wireless networks for tasks such as link quality prediction [7, 8, 9], modulation classification [10], signal-to-noise ratio (SNR) ordering [11], and signal detection [12]. In contrast, unsupervised learning has

been explored for clustering [13, 14], MIMO beamforming [15], and tracking and localization. Reinforcement learning, known for its ability to solve large-scale problems modeled as Markov Decision Processes (MDPs), has also been applied to spectrum sharing. For instance, Bower et al. [16] investigated RL in both competitive and cooperative spectrum sharing systems as part of the DARPA Spectrum Collaboration Challenge.

Dynamic Spectrum Allocation (DSA) is a cornerstone of modern wireless networks, enabling more efficient use of the radio spectrum. As wireless environments become increasingly complex, AI and ML have emerged as essential tools for enhancing spectrum management. These technologies are used to predict spectrum availability, optimize allocation strategies, and improve spectrum sensing and classification. Gbenga-Ilori et al. (2025) [17] provide a comprehensive overview of AI applications in wireless communications, emphasizing deep learning for spectrum sensing, multi-agent reinforcement learning and generative AI for network optimization. Lakouismi et al. (2025) [18] further explore the challenges of real-time deep learning-based DSA, particularly in terms of energy efficiency and latency.

Cognitive Radio Networks (CRNs), which dynamically access underutilized spectrum bands, benefit significantly from ML techniques. These include enhancements in both cooperative and non-cooperative spectrum sensing, as well as spectrum auction mechanisms and predictive modeling. Upadhye et al. (2021) [19] provide a broad survey of ML algorithms applied to CRNs, while Arjoune and Kaabouch (2019) [20] offer a detailed review of spectrum sensing techniques.

Further research into AI/ML applications for next-generation networks includes studies on AI in 6G spectrum management [21], AI for hybrid satellite-5G networks [22], and cognitive reinforcement learning strategies for spectrum allocation in 5G environments [23].

Graph Neural Networks (GNNs), a class of deep learning models designed for graph-structured data, are particularly well-suited to wireless networks, where nodes (e.g., base stations, users) and their interactions (e.g., interference, connectivity) naturally form graphs. GNNs can effectively model these relationships to optimize resource allocation, including spectrum access. Lu et al. (2024) [24] provide a comprehensive overview of GNNs in wireless networks, detailing how to represent wireless environments as graphs, design appropriate GNN architectures, and evaluate their performance. MathWorks (2024) [25] demonstrates the use of GNNs for power allocation, modeling the network as a graph and solving constrained optimization problems.

The SpectrumX Project [26] frames interference management as a graph coloring problem, using GNNs to dynamically assign spectrum while minimizing interference. Similarly, a GitHub project [27] introduces a GNN-based framework for jointly optimizing power control and spectrum allocation, showcasing the flexibility of GNNs in multi-objective optimization. A dissertation from the University of Rhode Island [28] explores GNN-based learning for dynamic spectrum access, emphasizing the need for intelligent models to address spectrum scarcity and user mobility. Shen et al. (2022) [29] demonstrate that GNNs can achieve near-optimal performance in

large-scale wireless networks with significantly fewer training samples than traditional neural networks. Suárez-Varela et al. [30] further discuss the generalization capabilities of GNNs in communication networks, highlighting their robustness across diverse network topologies and configurations.

1.4 Dissertation Outline

The remainder of this document is organized as follows. Chapter 2 describes the overall spectrum management approach, including the approach for demand prediction and resource allocation. Chapter 3 describes a resource allocation approach using graph theory and graph neural networks. Chapter 4 describes the assessment of both a policy-based and a value-based multi-agent reinforcement learning based approach for dynamic resource allocation. Finally, Chapter 5 presents conclusions and future research possibilities.

CHAPTER II

SPECTRUM MANAGEMENT APPROACH

2.1 Autonomous Spectrum Concept

The proposed concept for autonomous spectrum management aims to provide an intelligent, efficient, flexible, and scalable solution to improve spectrum utilization (allocation, access and sharing) using recent advancements in artificial intelligence and machine learning. The system under consideration is the band of frequencies used for aviation communications including the air-to-ground and air-to-air communication networks.

Fig. 2.1 depicts a process flow of the proposed autonomous spectrum management concept. Based on the input of available data items (i.e., air traffic, communication demands, airspace configurations and constraints), the learning and prediction module (the core functions distinct from previous systems) uses various AI algorithms to predict traffic and communications demands, to perform resource management (spectrum allocation, access and sharing) and to guide the selection of RF transceiver parameters at both the physical layer (PHY) and the medium access control (MAC) layer to meet the overall communication requirements throughout the airspace system.

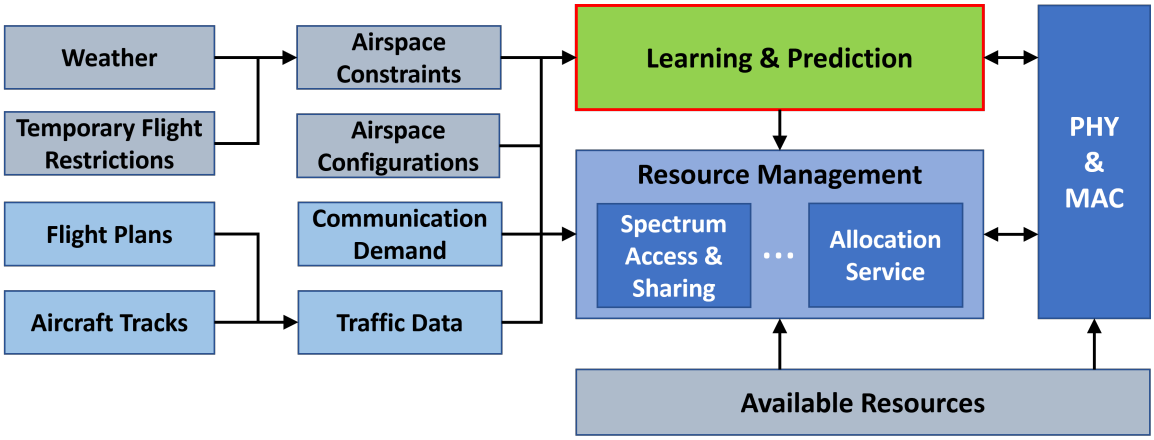


Figure 2.1: Functional flow of the autonomous spectrum management concept. The diagram shows how prediction and learning modules process air traffic, communication demands, and airspace constraints to allocate spectrum resources dynamically. This approach enables efficient and adaptive frequency management for aviation communications.

The prediction functionality is a prerequisite for the ensuing resource allocation module. Specifically, the prediction module outputs air traffic and communications demands (i.e., prediction products) throughout the airspace, whereby the resource allocation module dynamically reserves resources based on the predicted results and other inputs. Then these reserved resources are dynamically allocated to communication assets throughout the air and ground portions of the airspace system. To achieve the goal of safe and efficient air operations, the resource allocation function strives to maximize the utilization efficiency of available resources (spectrum, power, time, etc.), while meeting the desired Quality of Service (QoS) requirements for users throughout the airspace [31].

Learning-based algorithms require various datasets in order to train, test, and validate model performance. Of particular note, the diversity and size of the datasets

will continue to grow as the airspace system evolves with new applications. For now, the concept focuses on three types of data: flight traffic data, communication data, and airspace regulatory data including constraints and configurations. More specifically, airspace traffic data can be obtained using data items such as flight plans and aircraft tracks; additionally, airspace constraints data can be determined by information contained in weather data and Temporary Flight Restrictions (TFRs). The learning and prediction algorithms use both the historical traffic and communication data and the airspace constraints and configurations to predict future airspace traffic and communication demands. All the data items combined with the predicted data serve as the input for dynamic resource management.

2.2 Resource Allocation

To achieve the ultimate goal of safe and efficient air operations, the resource allocation function strives to maximize the utilization efficiency of available resources (spectrum, power, time, etc.), while meeting the desired safety and QoS requirements for users throughout the airspace. In this research, two methods of resource allocation were investigated: reinforcement learning-based approach, and an approach that investigated graph coloring using a graph neural network for channel assignment.

2.2.1 Resource Allocation using Graph Neural Networks

The use of graph coloring in the context of resource allocation has been a widely-studied problem for spectrum management applications [32, 33, 34, 35, 36], and this investigation focused on the use of a graph coloring approach to address

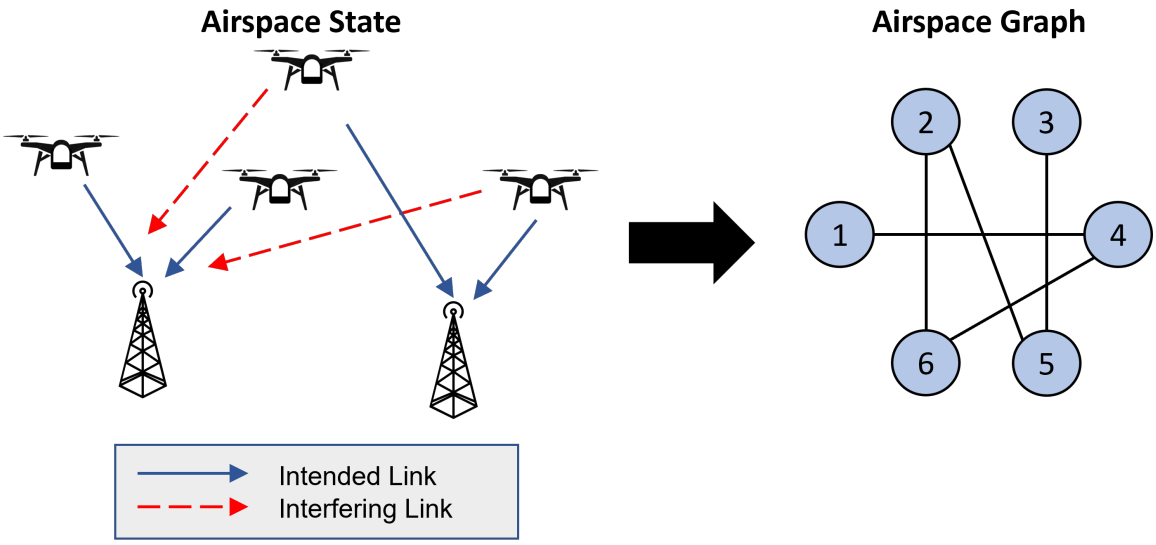


Figure 2.2: Example of an airspace represented as a graph for resource allocation. Aircraft are modeled as nodes, and edges indicate potential interference between aircraft operating in close proximity. This representation supports graph-based algorithms for channel assignment.

the channel assignment problem in AAM scenarios. Graph coloring is a method that assigns colors to vertices of a graph such that no two vertices linked by an edge share the same color, and Fig. 2.2 provides an example of how aircraft proximity can be translated to an interference graph whose vertices and edges are defined by regions of airspace where an aircraft may experience detrimental interference. Aircraft that are sufficiently separated in space would not present a risk of interference and would thus not be joined by an edge in the graph representation. The team has chosen to implement a graph convolutional neural network inspired by [37, 38]. As an input to the network, an adjacency matrix is formed based on the relative positioning of aircraft within a region of airspace. Given the adjacency matrix as an input, the neural network model returns a coloring solution that attempts to satisfy the graph

coloring problem, or in the absence of a solution, minimizes the number of conflicts through an energy-based loss function.

2.2.2 Resource Allocation using Reinforcement Learning

Reinforcement learning is a form of machine learning where an intelligent agent learns how to optimally interact with its environment via feedback (or rewards) based on its selected actions. More specifically, a RL model includes the following basic elements:

- Environment: the physical space in which the intelligent agent operates and learns.
- State: position of the agent at an instance of time; after issuing an action, the agent receives a reward and a change in state.
- Rewards: the feedback issued to the agent from the environment; rewards can be both positive and negative, depending on the goal of the agent.
- Policy: a mapping of the agent's current state to its next selected action.
- State-Value: expected cumulative reward that an agent can obtain starting from state s and following a specific policy thereafter.

The RL agent issues actions based on its policy; actions may include channel and power selections, as well as others, depending on the specifics of the use case. The environment would be the airspace system (both air and ground elements) and information relating to the users in the airspace, such as predicted positions, actual

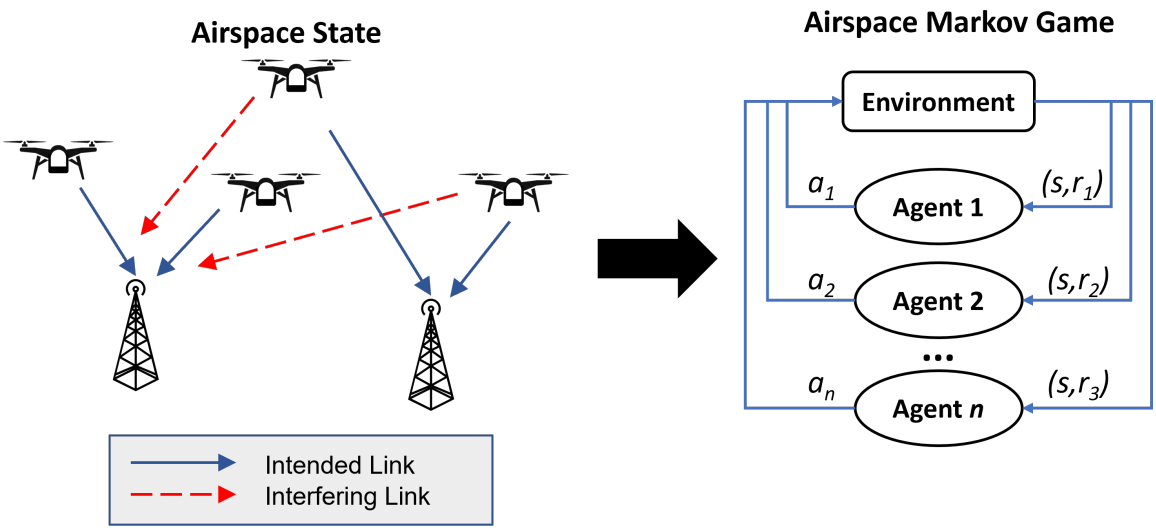


Figure 2.3: Illustration of an airspace scenario modeled as a Markov game for reinforcement learning. Each agent represents an aircraft that selects actions such as channel or speed adjustments based on its state and interactions with other agents.

positions, channel conditions, desired data services and user priorities. The agent receives both an updated state and a reward from the environment. Examples of positive rewards may be issued as a result of increased aggregate user data rates, or an increase in the airspace capacity. Negative rewards may be a result of link losses or QoS violations. In general, however, the specifics of the RL elements will be dependent on the selected use case. Fig. 2.3 illustrates a mapping of an airspace state to a Markov game, which extends the single-agent RL framework to a multi-agent interaction.

2.3 Prediction Methods

As part of the overall spectrum management concept, the prediction module is a prerequisite for the ensuing allocation module. Specifically, the prediction module

outputs air traffic and communications demands (i.e., prediction products) throughout the airspace, whereby the allocation module dynamically reserves resources based on the predicted results and other inputs. Then these reserved resources are dynamically allocated to communication assets throughout the air and ground portions of the airspace system. This effort focused on three types of predictions: aircraft trajectory prediction, airspace density prediction, and communications demand prediction.

2.3.1 Flight Trajectory Prediction

Flight trajectory prediction serves as a data input to resource allocation decision engines. A sufficiently accurate trajectory prediction model can serve to forecast the regions of airspace that an aircraft may occupy, potentially indicating the needed communication resources for each flight segment. The location of the aircraft can furthermore be used to estimate communication link path loss and other channel quality metrics, providing insight for power budgeting and resource allocation. Finally, given a set of rules which define the occurrence of communication events, the spectrum demand for airspace users can be inferred by comparing an accurate trajectory prediction against airspace sector boundaries and the last-filed flight plan of the aircraft.

This research effort investigated both learning-based and non-learning-based methods of flight trajectory prediction. Specifically, in [39] the team investigated a generalized hybrid-recurrent predictive model for flight trajectory prediction. The generalizable deep learning approach showed improvement in trajectory prediction accuracy, but also can be contextualized by exploring a large amount of data. Exper-

imental results illustrated a tradeoff between horizontal and vertical errors as flight data are generalized across dates and routes.

2.3.2 Communications Demand Prediction

Airspace communications are necessary for a variety of potential events, each requiring unilateral access to a given channel for the duration of the event. As such, communications demand prediction attempts to forecast these resource requirements (channel access duration, bandwidth, power budget, etc.) based on dynamics within the airspace system. The primary challenge in this area of demand prediction is sufficiently capturing and representing communication events and parameters in terms of available data. While airspace density is a key component to representing this demand, it fails to capture the total complexity of airspace communications.

In [40], the team investigated a federated learning approach to predict communication events in terms of the type of event occurring (sector coordination, conflict resolution, etc). As part of this, an investigation of various data products was conducted to identify trends, sufficiency, and correlations of each product to identified events. The model enabled a generalized approach to understanding airspace communications which is able to inform long-term changes to aviation spectrum management.

In [41], the team investigated a non-learning based approach for communications demand prediction. In this case, rather than allocate channels to each aircraft in the simulation, the function only allocated resources to aircraft that were predicted to require communications using a simple method of examining the flight path of the

aircraft with respect to airspace boundaries. This method assumed that an aircraft only requires an allocated channel when transitioning between regions of airspace. This is described in more detail in Chapter 3.

2.3.3 Airspace Density Prediction

Airspace density (i.e., aircraft count) prediction considers an aggregated approach to determining sector demand. In existing research, each airspace sector is treated as a unique state with an associated probability for transitioning to any other state [7]. Using air traffic metadata, a periodic record of sector densities can be combined with weather and air traffic management data to train a predictive model, which can be used to forecast airspace sector density, and consequently estimate the need for communication resources.

2.4 Modeling and Simulation Tool Development

To implement the intelligent spectrum management concept and evaluate its performance, a modeling and simulation platform consisting of computational tools and environment visualization has been developed. An important aspect of the tool is the development, integration, and evaluation of learning-based models in a simulated airspace operational environment. Additional design objectives include the development of a live, virtual and constructive operational mode in which concepts can be incrementally evaluated as maturity and confidence is achieved.

A modularized architecture design is employed to enable a flexible and scalable approach to evaluate and test various communications systems, airspace environ-

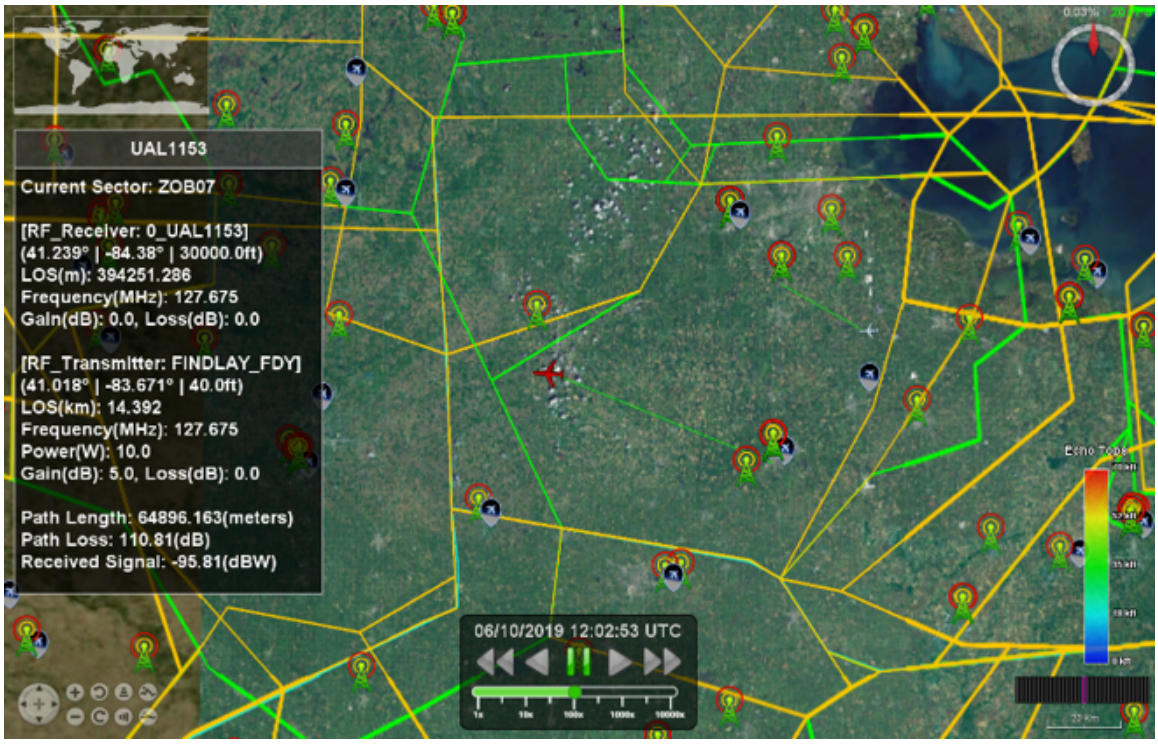


Figure 2.4: Screenshot of the custom modeling and simulation tool visualizing Cleveland airspace. The tool displays aircraft positions, communication links, and sector boundaries, enabling evaluation of spectrum allocation strategies in realistic scenarios.

ments, and neural network models. The tool is designed to enable rapid development of new airspace configurations and to apply new channel assignment strategies for evaluation. Simulations generate a dataset that is representative of the airspace environment (i.e., adjacency matrix) and provides this information to either internal or external functions for additional processing. External functions may include various artificial intelligence or machine learning applications. The simulation software then utilizes the outputs of these prediction and allocation functions to examine the quality of services for each active communication link in the airspace. Using this simulation tool, various learning-based methods can be evaluated as allocation functions

for channel resource assignment. Fig. 2.4 provides a visualization of the Cleveland Airspace System using the modeling and simulation tool.

2.5 Open Challenges

Implementation of the proposed intelligent spectrum management capability poses a number of challenges, including spectrum coordination, computational capabilities, data requirements, and others. The following paragraphs provide a brief summary of the challenges to infusion of this concept [31].

- Spectrum Coordination: Proposing the re-use and re-allocation of existing aviation frequency allocations would require coordination with appropriate spectrum management offices, such as the FAA Spectrum Engineering Office, and regulatory bodies, such as the National Telecommunications and Information Administration (NTIA), to ensure alignment with applicable policies.

- Computational Capabilities: Execution of artificial intelligence algorithms inherently require enhanced computational capabilities in order to manage and process massive amounts of data within a bounded time allocation. As such, new aircraft systems must possess sufficient computing capabilities for successful implementation.

- Data Engineering: Effective use of learning-based methods depends heavily upon the quantity, availability, and integrity of the datasets used for training, validation, and testing of algorithms. Consequently, optimal implementation of

the described concepts relies heavily upon data engineering efforts, including data acquisition and processing.

- Transition of New Approach into the Airspace: Integration of any new technology into the airspace system will have to accommodate established spectrum assignment practices to enable traditional operational continuity, while supporting users' upgrades to radio technologies that support new approaches.

- Technology Implementation Independence: Implementation of a new spectrum allocation concept must remain agnostic of the underlying communications system to reduce risk of limiting future modernization efforts and interoperability. The solution requires the ability to meet the needs of existing technologies while enabling the use and implementation of novel technologies.

2.6 Conclusions and Follow-on Studies

This chapter presented an intelligent spectrum management framework for aviation, combining prediction, learning, and allocation to improve resource efficiency. Two dynamic allocation methods were mentioned: graph neural networks (GNNs) for spatially-aware channel assignment, and reinforcement learning for adaptive decision-making in complex airspace environments.

The following chapters will explore the GNN and multi-agent RL methods in greater detail, focusing on their application to dynamic spectrum allocation and airspace management.

CHAPTER III

RESOURCE ALLOCATION USING GRAPH NEURAL NETWORKS

3.1 Overview and Methods

This chapter focuses on the examination of a graph coloring approach to address the channel assignment problem for a hypothetical AAM use case under interference constraints. The investigation compares the performance of a relatively simple (e.g., "greedy") algorithm to the performance gained by using a learning-based algorithm, specifically a graph convolutional neural network, to solve the resource allocation problem. Utilizing multiple methods to dynamically assign resources allows for selection depending on the complexity of the problem. In cases where the number of available resources exceeds demand, a simple greedy (or first-come, first-served) algorithm is more than sufficient. However, in cases where demand exceeds the available resources, there is potential for interference between many of the aircraft in the scenario. In such cases, the use of a more sophisticated approach based on machine learning may serve to provide a better solution. Accordingly, this chapter describes the performance comparison of a relatively simple greedy algorithm with the performance of a learning-based algorithm, specifically a graph convolutional neural network.

3.2 Problem Simplification

Depending on the complexity of the airspace modeling scenario, determining a valid solution via graph coloring may be difficult or impossible considering the large number of nodes with many edges. One approach to simplify the problem complexity is to only assign spectrum when needed, whereby predictive methods may be used to eliminate interference scenarios that may not be relevant at the time of consideration. For example, airspace nodes will likely not interfere with each other if there is no need for concurrent transmission. Additionally, airspace nodes that are substantially separated in distance are not able to pose unwanted interference, and may also be eliminated from the problem space. As a result, the implementation of methods to predict when communication events may occur can substantially reduce problem complexity and allow for the determination of a solution. Elimination of irrelevant airspace users will remove potential instances of interference, resulting in a simpler graph representation with fewer edges. For this effort, a more simple approach is used to 1) generate flight tracks between vertiports and 2) estimating communication demand by evaluating airspace sector crossing events, which will be described in further detail later in the chapter.

3.3 System Model and Problem Formulation

3.3.1 UAM System Model

This study considers a UAM scenario where multiple aircraft are traveling en-route from an origin vertiport to a destination vertiport, as depicted in Fig. 3.1. The network consists of a finite set of all aircraft $\mathcal{N} = \{n_1, ..., n_N\}$ operating in the airspace system. To maintain safe operations, two-way reliable communications between the aircraft and the base stations (BS) is required, and each aircraft communicates with base stations using bidirectional communications where uplink represents aircraft-to-ground exchanges and downlink represents ground-to-aircraft exchanges. The set of base stations is denoted as $\mathcal{B} = \{b_1, ..., b_M\}$. Spectrum reuse is employed such that a finite set of available channels, denoted as $\mathcal{K} = \{k_1, ..., k_K\}$, may be simultaneously shared by multiple users provided the interference is maintained within acceptable levels. The network is modeled by using discrete the time steps denoted as $T = \{1, 2, ..., T\}$ of duration ΔT.

3.3.2 Interference Considerations

For interference calculations, this study focuses on the inter-site frequency engineering procedures for very high frequency (VHF) air-to-ground links as described by the Federal Aviation Administration (FAA) in [42]. Specifically, the procedures require that at worst-case, an aircraft will receive the desired signal at a level that is γ(dB) stronger than the signal received from a co-channel interferer. This desired-to-undesired ratio is dependent upon the distances of the intended and interfering

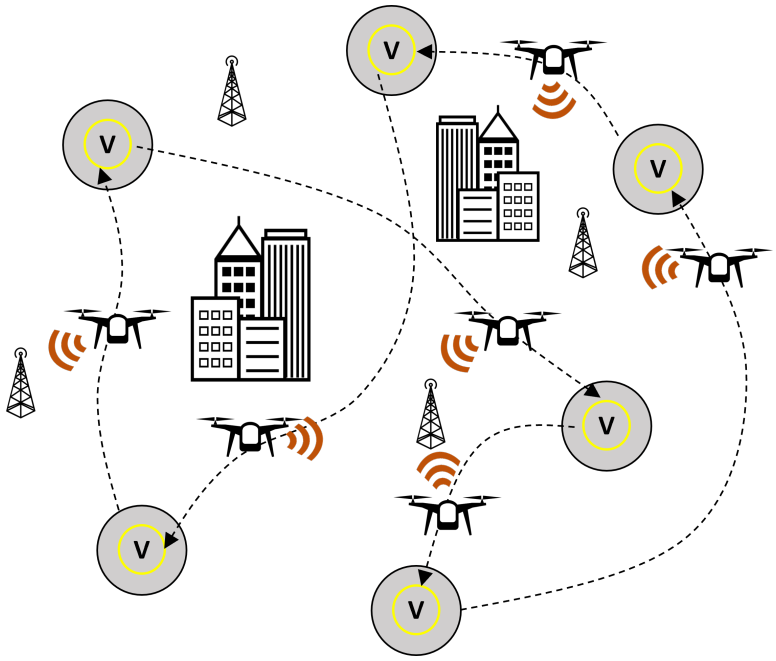


Figure 3.1: System model for an Urban Air Mobility scenario. The diagram shows aircraft traveling between vertiports and communicating with ground-based base stations. It illustrates the network structure used for channel assignment and interference analysis.

aircraft to the receiver and the path loss exponent, $\delta \in \mathbb{R}$. Fig. 3.2 illustrates the air-to-ground interference model where aircraft n_i in the airspace system transmits at a power level of p_i, and the distance between aircraft n_i and base station b_m is denoted as $d_{i,m}$. The interfering aircraft n_j transmits at a power level p_j, and the distance between aircraft n_j and the base station b_m is $d_{j,m}$. The distance ratio $d_{j,m}/d_{i,m}$ can be calculated to meet the desired-to-undesired ratio of γ(dB) at a receiver at base station b_m starting from (3.1), which uses the ratio of received powers γ as scalar quantity (i.e., not in dB). The distance ratio required for a power ratio in terms of dB is shown in (3.3), where (3.2) provides the intermediate operation.

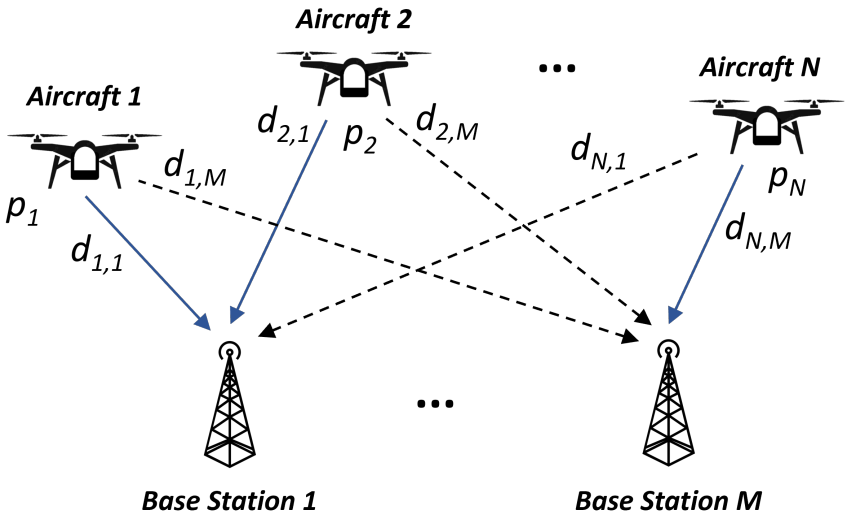


Figure 3.2: Air-to-ground interference model used to construct the interference graph. The figure depicts how distances between aircraft and base stations influence co-channel interference calculations for spectrum allocation.

$$\frac{p_i d_{i,m}^{-\delta}}{p_j d_{j,m}^{-\delta}} \geq \gamma \tag{3.1}$$

Since the desired-to-undesired ratio is measured in dB, that translates to the log ratio of received powers must exceed the desired-to-undesired threshold in dB

$$10 \log \frac{p_i d_{i,m}^{-\delta}}{p_j d_{j,m}^{-\delta}} \geq \gamma(\text{dB}) \tag{3.2}$$

Simplifying and rearranging terms results in the desired-to-undesired distance ratio

$$\frac{d_{j,m}}{d_{i,m}} \geq \left(\frac{p_j}{p_i} 10^{0.1\gamma(\text{dB})} \right)^{1/\delta} \tag{3.3}$$

Assuming all aircraft transmit at the same power level, then $p_i = p_j \forall \{i, j\}$, and the distance ratio $d_{j,m}/d_{i,m}$ as shown in (3.4) is a function of the path loss exponent, δ and the desired-to-undesired ratio γ(dB). Selection of these parameters depends on

various factors.

$$D(\gamma(\text{dB}), \delta) = \left(10^{0.1\gamma(\text{dB})}\right)^{1/\delta} \tag{3.4}$$

For the desired-to-undesired signal ratio, denoted as γ (dB), guidance is provided by the FAA's communications frequency engineering specifications as described in [42]. Specifically, for VHF air-to-ground communications, a ratio of 14 dB is required. However, this specification pertains to analog pilot-controller communications and does not directly apply to Urban Air Mobility (UAM) data communication links. Since a standardized ratio for UAM has not yet been established, and because the specific choice of this ratio does not significantly affect the analysis or results of this study, a value of 14 dB is assumed for simplicity.

The path loss exponent, δ, varies depending on the propagation environment. In a pure line-of-sight scenario—where multipath interference is negligible—a value of $\delta = 2$ is appropriate. In more complex environments, such as urban cellular networks, δ typically ranges from 2.7 to 3.5. In obstructed indoor environments (e.g., with walls, floors, and ceilings), values can range from 4 to 6 [43].

For aircraft operating in urban environments, the path loss exponent depends on the phase of flight. At cruising altitudes, propagation conditions approximate free-space, with δ values between 2.0 and 2.1. Near vertiports—during arrival or departure—more obstructions and reflections are present, leading to higher path loss exponents, typically between 2.4 and 2.8 [44].

3.3.3 Problem Formulation

In this problem formulation, each of the N aircraft in the airspace system must be assigned one of K available channels while maintaining the prescribed interference constraints. This channel assignment challenge can be formulated as a type of graph coloring problem, where for a given graph $G = (V, E)$, each vertex $v_i \in V(i = 1, ..., N)$ represents an aircraft in the airspace system, and each edge $\{u, v\} \in E$ defines the interference relationship between the aircraft denoted by vertices u and v. Graph coloring, in general, strives to assign each vertex $v \in V$ an integer $c(v) \in \{1, 2, ...K\}$ such that $c(u) \neq c(v) \forall \{v, u\}$ and K is minimal [45]. The relationship between vertices and edges in a graph can be represented by an adjacency matrix, $\mathbf{A}_{N \times N}$, for which $A_{ij} = 1$ if and only if vertices v_i and v_j are adjacent (i.e., share an edge), and $A_{ij} = 0$ otherwise. Further, the assignment of colors to vertices is represented by a color assignment matrix, $\mathbf{C}_{N \times K}$, which uses one-hot encoding such that $C_{ij} = 1$ if vertex v_i is assigned color j, and 0 otherwise. Each vertex v_i can only be given a single color assignment; therefore $\sum_{j=1}^{K} C_{ij} = 1 \forall v_i \in V$.

In this study, it is assumed that the number of available communication channels (i.e., colors) is fixed, so this becomes a graph k-coloring problem in which the coloring algorithms will strive to obtain a coloring solution that reduces color conflicts by minimizing a loss function. To obtain a suitable loss function, it has been shown that the graph k-coloring problem can be naturally formulated as a quadratic unconstrained binary optimization (QUBO) problem [46] [37] [38], which in general form is the minimization of $f : \{0, 1\}^N \rightarrow \mathbb{R}$, with the objective function as the

following:

$$\min f(\mathbf{x}) = \mathbf{x}^T \mathbf{Q} \mathbf{x} = \sum_{i,j=1}^{N} Q_{i,j} x_i x_j$$

$$\mathbf{x}^* = \arg \min_{x \in \{0,1\}^N} f(\mathbf{x})$$

where $\mathbf{Q}$ is a real symmetric matrix, f is the cost function and $\mathbf{x}^*$ is a solution bit string or a global minimizer of f [46]. Here, $x_{v,i}$ uses one-hot encoding, i.e., if vertex v is assigned the color j then $x_{v,j} = 1$ and for all $i \neq j$, $x_{v,i} = 0$. This is similar to how the color assignment matrix, $\mathbf{C}$ was defined previously. For all pairs (v, w) of neighboring sites, i.e., $A_{vw} = 1$, those with the same colors assignments will be penalized by the term $\sum_{i=1}^{K} x_{v,i} x_{w,i}$. Further, [46] shows that the loss (or cost) function for k-coloring of a graph with N nodes and adjacency matrix A can be written as:

$$f(x) = \sum_{v,w=1}^{N} A_{vw} \sum_{i=1}^{K} x_{v,i} x_{w,i} = \sum_{i=1}^{K} \mathbf{x_i}^T \mathbf{A} \mathbf{x_i}$$

Accordingly, the loss function for this study follows a similar approach. For a given adjacency matrix, $\mathbf{A}$ and color matrix $\mathbf{C}$ the loss function is defined in (3.5), where $\mathbf{c}_i$ represents each column of the color matrix $\mathbf{C}$.

$$\min L(\mathbf{C}) = \sum_{i=1}^{K} \mathbf{c}_i^T \mathbf{A} \mathbf{c}_i \tag{3.5}$$

In the case where a valid coloring solution exists (i.e., no adjacent vertices are assigned the same color), then the loss function will be 0. As the number of color conflicts increases, the loss function increases proportionally.

3.3.4 System Model as Interference Graph

To solve the channel assignment problem presented in the system model, an interference graph is created based on the relative positions of aircraft and base stations in the airspace system at each instance of time. As described previously, co-channel interference is acceptable provided that the ratio of the distances any pair of transmitting nodes from the receiver does not exceed a threshold as described in (3.3). Accordingly, the interference graph is constructed as follows. For all N aircraft in the airspace system, create a vertex (or node) in the interference graph G. At this initial step, there are no edges created between vertices in the graph, and the edges will be determined based on relative positions from transmitting nodes to receivers at the base stations according to (3.4). For every base station receiver b_m in the airspace system, for every 2-tuple unique combination of aircraft transmitters (n_i, n_j) that have line-of-sight with b_m, calculate the ratio of distances, $d_{i,m}/d_{j,m}$ from the base station receiver b_m, and if the distance ratio exceeds the threshold limit from (3.3) then aircraft nodes n_i and n_j share an edge in the interference graph. The process is repeated for all base station receivers from $\mathcal{B}$. The interference graph construction algorithm is shown in Alg. 1, and an illustration of a 4-node interference graph is shown in Fig. 3.3.

Algorithm 1: Interference graph construction algorithm.

Input : Locations of N Aircraft and B Base Stations

Input : Interference threshold, γ (dB)

Input : Path loss exponent, δ

Output: Interference Graph, G

Initialize Graph $G(V, E)$ with N nodes and no edges;

`// Loop through all Base Station receivers`

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

    At base station b_m, $L_m \leftarrow$ LineOfSight($\mathcal{N}$);

    foreach *2-tuple combination of transmitters* $(n_i, n_j) \in L_m$ **do**

        `// Check distance ratio`

        if $\frac{d_{j,m}}{d_{i,m}} < D(\gamma, \delta)$ **then**

            `// Unacceptable interference`

            `// Join nodes with edge in graph` G

            CreateEdge(n_i, n_j);

return G

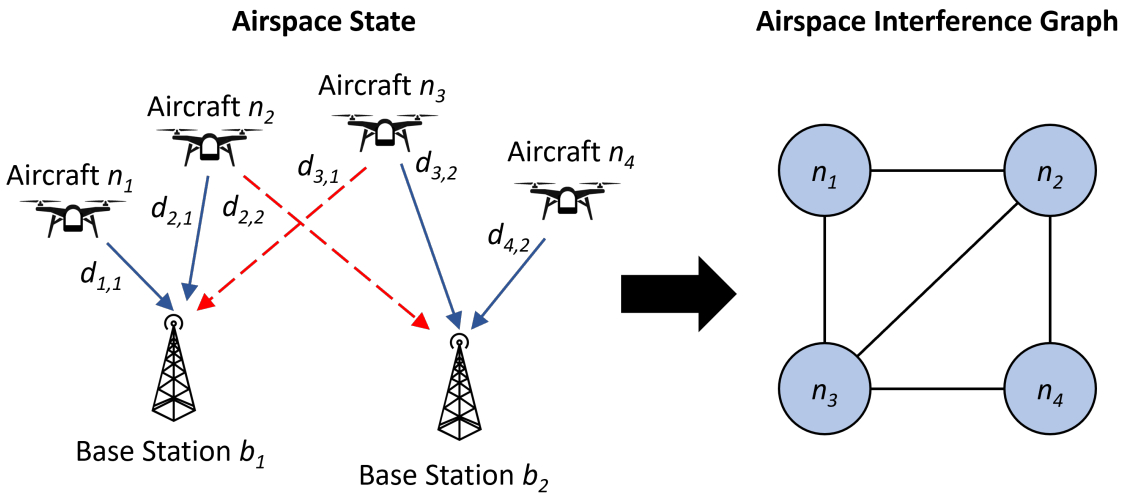


Figure 3.3: Example of an interference graph for a four-aircraft scenario. Nodes represent aircraft, and edges indicate potential interference relationships based on spatial proximity and frequency reuse constraints.

3.4 Resource Allocation via Graph Coloring

3.4.1 Preliminaries

Graph coloring is an NP-hard problem. Exact solutions can be obtained using methods that operate in exponential time; however, the use of exact algorithms is generally only feasible for small graphs or those with simple topologies. Heuristic methods may achieve approximate solutions in polynomial time, and in many cases, may obtain reasonable solutions in tractable execution time. Common heuristic graph coloring algorithms include greedy, DSATUR, and Recursive Largest First (RLF). Empirical comparisons of the coloring performance, in terms of colors required and execution time, for these three methods has shown that RLF achieves the best performance in terms of fewest colors required, but requires the most execution time.

The greedy algorithm performs poorest in terms of colors required but exhibits the shortest execution time [45]. As an alternative to heuristic methods, recent investigations of deep learning methods for graph structures, specifically, graph neural networks, has been shown to be effective in solving the graph coloring problem [47], [37], [38], [48], which is the subject of this study for solving the airspace channel selection problem.

3.4.2 Greedy Coloring

A fundamental method to solve the graph coloring problem is via a greedy coloring algorithm, which is analogous to a "first-come, first-served" scheduling process or the functionality of cognitive radios that "listen" for open channels and make use of any unoccupied spectrum at the time. The greedy algorithm has at its disposal a list of available channels (or colors) for each node within the airspace. Based on observations of the airspace state, a graph is constructed at each timestep that consists of a node for each aircraft and edges between nodes depending on whether nearby aircraft may pose undesired co-channel interference. The algorithm operates by addressing each node one-by-one according to an arbitrarily-selected ordering scheme, often selected as a largest-first ordering [45], and then the algorithm assigns each node the first available color. For this study, however, due to limited available spectrum channels (or colors), the algorithm was modified to randomly assign a color in cases where there is no available color that would satisfy the problem. Clearly, this method may not provide an optimal solution and may even fail when a valid solution exists. Computational complexity is an additional consideration and may become an issue

when examining larger systems. The greedy coloring algorithm is shown in Algorithm 2.

Algorithm 2: Greedy Graph Coloring.

Input: Graph $G = (V, E)$, color palette $\mathcal{C}$, global color map M

Output: Updated color map M with initial assignments

```
1  foreach v ∈ V do
2      if v ∈ M then
3          continue ;                    // Node already colored globally
4      N_c ← {M[u] | u ∈ Neighbors(v) and u ∈ M};
5      foreach c ∈ C do
6          if c ∉ N_c then
7              M[v] ← c;
8              break;
9      if v ∉ M then
10         M[v] ← random color from C;
11         increment conflict count;
12 return M
```

3.4.3 Dynamic Graph Coloring

Generating an interference graph from scratch at every timestep can be inefficient in situations where the vertices and edges of the graph do not change significantly from one timestep to the next. Accordingly, the dynamic graph coloring problem involves a sequence of graphs $G = (G_1, G_2, \ldots, G_n)$, and for each graph

$G_i \in G$, a coloring solution must be obtained within the time frame before the next graph G_{i+1} must be considered. Various non-learning-based dynamic coloring algorithms have been developed in the literature [49] [50] [51]. Additionally, combining the GCN solution with a recurrent neural network or other suitable construct may offer another potential solution to address the challenge of the time-dynamic graph coloring problem [52, 53, 54].

As part of this study, a dynamic graph coloring scheme was devised that, after an initial greedy coloring, attempts to preserve coloring assignments across graphs from timestep to timestep. The algorithm implements a conflict reassignment algorithm that, after the initial coloring, iteratively reassigns colors to nodes that are involved in conflicts, while preserving the assignment of colors to nodes that are not in conflict. The algorithm terminates when no further improvements are possible or after a set number of iterations. The conflict reassignment method is shown in Algorithm 3.

Algorithm 3: Conflict Reassignment Optimization.

Input: Graph $G = (V, E)$, color palette $\mathcal{C}$, color map M, max iterations k

Output: Optimized color map M with reduced conflicts

```
for i ← 1 to k do
    improved ← false;
    foreach v ∈ V do
        c_curr ← M[v];
        conflicts_curr ← number of neighbors of v with color c_curr;
        foreach c ∈ C \ {c_curr} do
            conflicts_new ← number of neighbors of v with color c;
            if conflicts_new < conflicts_curr then
                M[v] ← c;
                improved ← true;
                break;
    if not improved then
        break ;                          // No further improvements
return M
```

3.4.4 Graph Convolutional Neural Network

As airspace scenarios become more complex, artificial intelligence and machine learning can be used as tools to aid in the resource allocation problem [55].

However, given the large number of possible scenarios that may or may not have solutions nor labeled training data, investigation of unsupervised learning techniques allows for a viable approach. Accordingly, the team has chosen to implement a graph convolutional neural network (GCN) inspired by [37, 38]. GCNs are a type of graph neural network (GNN) that extends the concept of convolution from convolutional neural networks (CNNs) to graph-structured data. GCNs are able to learn vector representations (or embeddings) for each node by aggregating information from neighboring nodes using a message-passing process (with each pass referred to as a "hop"), thereby capturing both individual node features and as well as features relating to the the overall graph structure [56]. In effect, GCNs are able to extract features of the graph in a similar manner in which CNNs are able to extract features from grid-structured data.

3.4.5 GCN Architecture

Several variants of GCNs have been developed in the literature [55] [57]. In this study, the graph network utilizes two topology adaptive graph convolutional layers (TAGConv) from [58], since it offers enhanced performance over other implementations at a reduced computational complexity. The output of the TAGConv layer, H^W, after W hops (defaults to 2) is described by (3.6), where A denotes the adjacency matrix, $D_{ii} = \sum_{j=0} A_{ij}$ is its diagonal degree matrix, and Θ_w denotes the linear weights to sum the hop results.

$$H^W = \sum_{w=0}^{W} (D^{-1/2} A D^{-1/2})^w X \Theta_w \tag{3.6}$$

As shown in Fig. 3.5, the input to the first TAGConv layer is the adjacency matrix, and has a parameter input feature of shape (N, e), where N represents the number of nodes (vertices) of the graph and e represents the embedding dimension, which is the length (i.e., number of elements) of the vector used to represent information (or embeddings) about the nodes in the graph. The output feature is of shape (N, h) where h represents the number of hidden dimensions which refers to the size of the vector used to represent each node within the hidden layers of the network. The input layer uses a ReLU activation function which serves as the input to the second TAGConv layer, of which the output feature is of shape (N, K) where N is the number of nodes in the graph and K is the total number of possible colors (or channels) assignments. The second TAGConv layer uses a Softmax activation function to produce the output probability matrix, $\mathbf{P}$.

3.4.6 GCN Training

As mentioned, at each timestep an interference graph G is constructed based on relative positions of aircraft and base stations in the system model. The graph is represented by an $N \times N$ adjacency matrix $\mathbf{A}$, which describes the relation of nodes (or vertices) and edges in the graph. The adjacency matrix serves as the input to the graph neural network, as illustrated in Fig. 3.5. The output of the network is an $N \times K$ probability matrix $\mathbf{P}$ that contains the set of probabilities that a node N (i.e., aircraft) would be assigned any 1 of K possible colors. Ideally, the probability of a

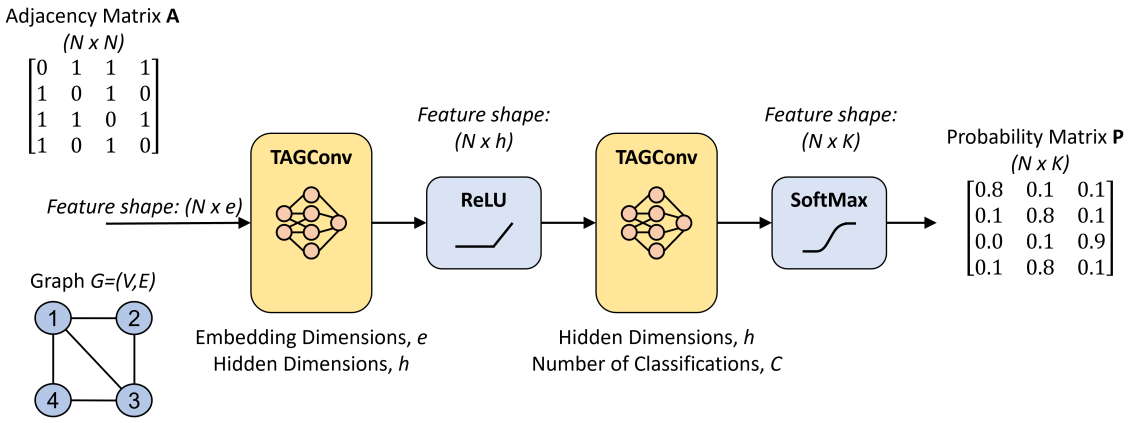


Figure 3.4: Architecture of the graph convolutional network (GCN) used for channel assignment. The model includes two TAGConv layers with ReLU and Softmax activations, transforming the input adjacency matrix into a probability matrix for color (channel) assignments.

single color should approach 1 in a well-trained network. In this context, reducing the loss to zero indicates a coloring solution with no conflicts (i.e., no adjacent vertices share the same assignment). However, in cases where there may be no valid color assignment solution, the neural network will strive to minimize conflicts in accordance with the loss function from (3.5).

As is typical for neural network training, each epoch consists of one complete pass of the training data (i.e., adjacency matrices) through the GCN layers. As each passthrough of the epoch is completed, neural network training is performed by calculating a loss value from loss equation in (3.5), where in this case the calculation uses the probable colors assignments as opposed to the "hard" assignments, $L(\mathbf{p}) = \sum_{i=1}^{K} \mathbf{p}_i^{\intercal} \mathbf{A} \mathbf{p}_i$ where $\mathbf{p}_i$ is a length N column vector of matrix $\mathbf{P}$, the probability of assigning the i^{th} color to each node in the graph represented by the adjacency matrix $\mathbf{A}$. The calculated loss, $L(\mathbf{p})$ is a scalar value that is used to perform back

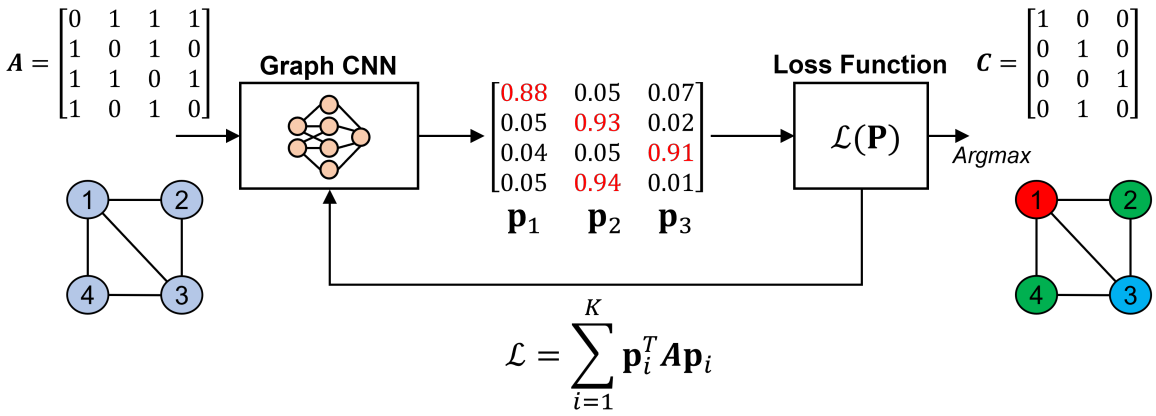


Figure 3.5: Training process for the graph convolutional network. The adjacency matrix is processed through the network to generate channel assignment probabilities. The loss function guides weight updates using the Adam optimizer during backpropagation.

propagation to adjust the neural network weights using the adaptive moment estimation (Adam) optimizer, and then the next training epoch begins. Fig.3.6, illustrates how the loss function is reduced through each training epoch. Specifically, this shows a comparison of training for two GCN variants, the TAGConv (as part of this study) and SAGEConv [59], where the TAGConv implementation is able to converge to a solution in fewer training epochs than the SAGEConv implementation.

3.5 Simulations

3.5.1 Simulation Configuration

The interference performance of graph coloring algorithms for aircraft channel assignment is compared by way of simulation. This scenario assumes communications links are established between aircraft and base stations on the ground (i.e., no air-to-air communications). Transceiver parameters assume constant transmit power,

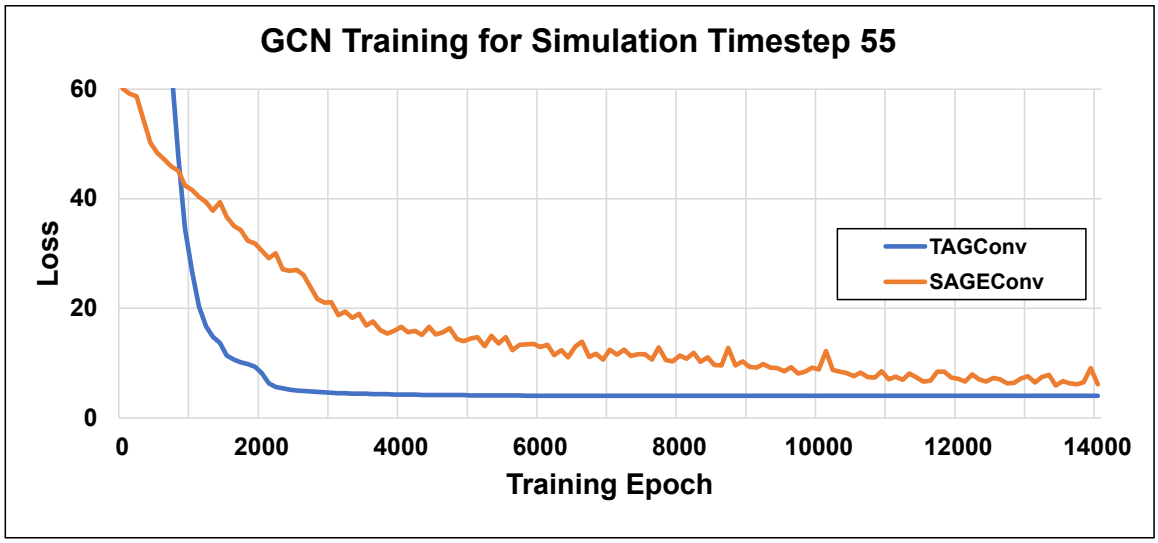


Figure 3.6: Comparison of training performance for two GCN variants at simulation timestep 55. The plot shows how the loss function decreases across epochs for TAGConv and SAGEConv implementations.

antenna gain, and channel bandwidth. Due to the strong line-of-sight component in air-to-ground communications, the path loss exponent for flights in an urban environment has been experimentally measured to be on the order of 2.1, as determined by [44] from their flight testing campaigns in the Cleveland metropolitan area in late 2024. Given a finite number of available frequency channels to share among all aircraft within the airspace, the resource allocation function will strive to provide a unique channel to each aircraft to eliminate the possibility of interference. In accordance with the assumptions of mid-term UAM operations, this simulation considers $N = 100$ aircraft traveling at an altitude of 3000 feet mean sea level (MSL). All aircraft traverse a linear lateral trajectory at a constant altitude and velocity between departure and arrival points over $T = 100$ simulation steps. Fig. 3.7 provides a visualization of the simulation scenario, where multiple aircraft are shown entering and

exiting regions of airspace and communicating with a total of $B = 269$ base stations that are located within hexagonal regions of airspace.

Assuming $p_i = p_j \forall \{i, j\}$ and given the desired-to-undesired ratio γ(dB) is 14 dB, the distance ratio $d_{j,m}/d_{i,m}$ must be at least 4.6 for a path loss exponent $\delta = 2.1$ (typical for air-to-ground UAM communications). For a path loss exponent of 2 (equating to free-space path loss), the distance ratio must be at least 5. For additional interference protection, this study assumes that any pair of transmitting nodes that are within a distance ratio of 6 to a receiver will be considered co-channel interferers. The simulation parameters are summarized in Table 3.1.

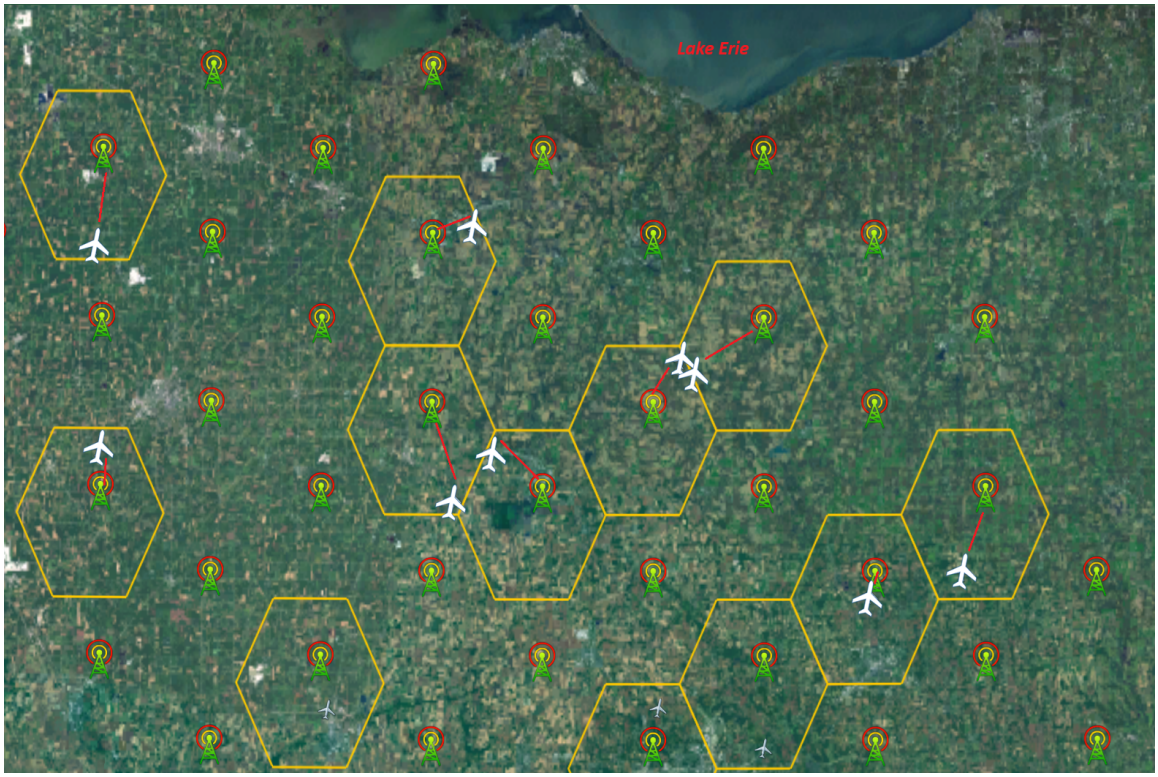


Figure 3.7: Visualization of the simulation scenario for Urban Air Mobility resource allocation. The figure shows aircraft trajectories, airspace sectors, and base station locations used in the evaluation of spectrum management strategies.

Table 3.1: Summary of simulation parameters for the Urban Air Mobility resource allocation study. The table lists air traffic characteristics, airspace design details, and communication link settings used in the experiments.

Air Traffic Parameters	
Number of aircraft	100
Simulation time steps	100
Flight paths	Linear lateral trajectory
Departure/arrival locations	Distributed across region
Aircraft Airspeed	Constant
Aircraft Altitude	Constant (3000 ft MSL)
Airspace Design Parameters	
Number of sectors	269
Sector shape	Hexagonal
Number of base stations	1 per sector
Communications Link Parameters	
Links	Air-to-ground
Transmit power	Constant
Transmit antenna gain	Constant
Channel bandwidth	Constant
Path loss exponent	2.1
SINR threshold (dB)	14

The allocation function utilizes the information on the state of the airspace in the form of an adjacency matrix. Initial simulations compare the number of interference events observed within a scenario described by the system model. Given these assumptions, a graph is created to represent the state of the airspace at each simulation step, and three allocation methods are examined: 1) greedy allocation, 2) dynamic greedy allocation strategy that maintains assignment continuity across timesteps while allowing recoloring to minimize conflicts, 3) GNN-Train100: a graph network trained using the entire simulation dataset (i.e., iterate over 100 input graphs), and 4) GNN-Train1: a graph network trained for each simulation step (i.e., a new network is trained for each input graph). The TAGConv layers use embedding dimension size $e = 60$ and hidden dimension size $h = 60$, and the number of available channels was constrained to $K = 20$. The simulation process flow is shown in Fig. 3.8.

3.5.2 Simulation Results

The number of observed interference events at each timestep of the simulation is shown in Fig. 3.9 and the cumulative sum of interference events over the simulation is shown in Fig. 3.10. The greatest number of interference events occur when the aircraft density is highest (between timesteps 30 and 70), and are represented by interference graphs with the greatest number of edges. For the allocation methods investigated, the greedy algorithm performs the worst when the aircraft density is high, while both GNN cases show improved performance in those situations. However at low aircraft density, the greedy outperforms the GNN-Train100, which is trained

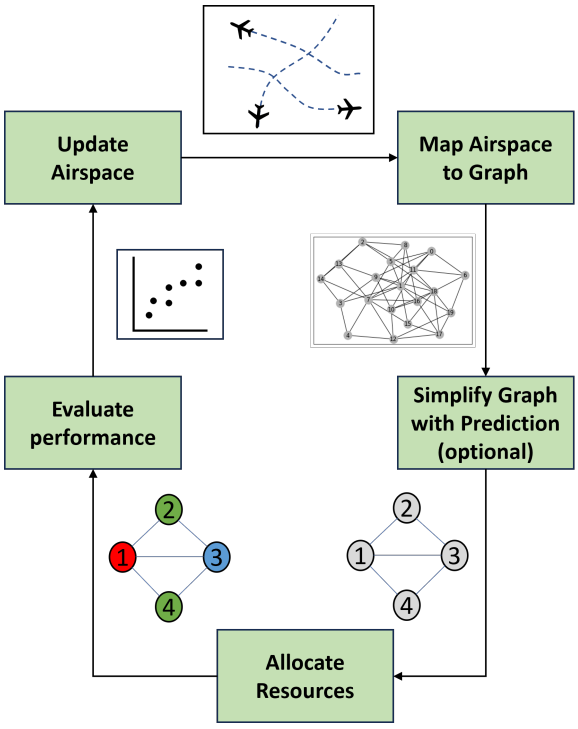


Figure 3.8: Process flow for the simulation framework. The diagram outlines steps for generating airspace states, applying allocation algorithms, and evaluating interference performance.

over the entire time series. The GNN-Train1, which trained a neural network on each individual simulation timestep, shows the best performance overall; however, due to required training time, a practical implementation of this method would be infeasible. Since the network weights of GNN-Train100 are already trained, channel assignments may be decided in near real-time, possibly quicker than the greedy implementation. The dynamic greedy method performed slightly better than the greedy method, but not as well as the GNN trained at each timestep. The interference performance is summarized in Table 3.2.

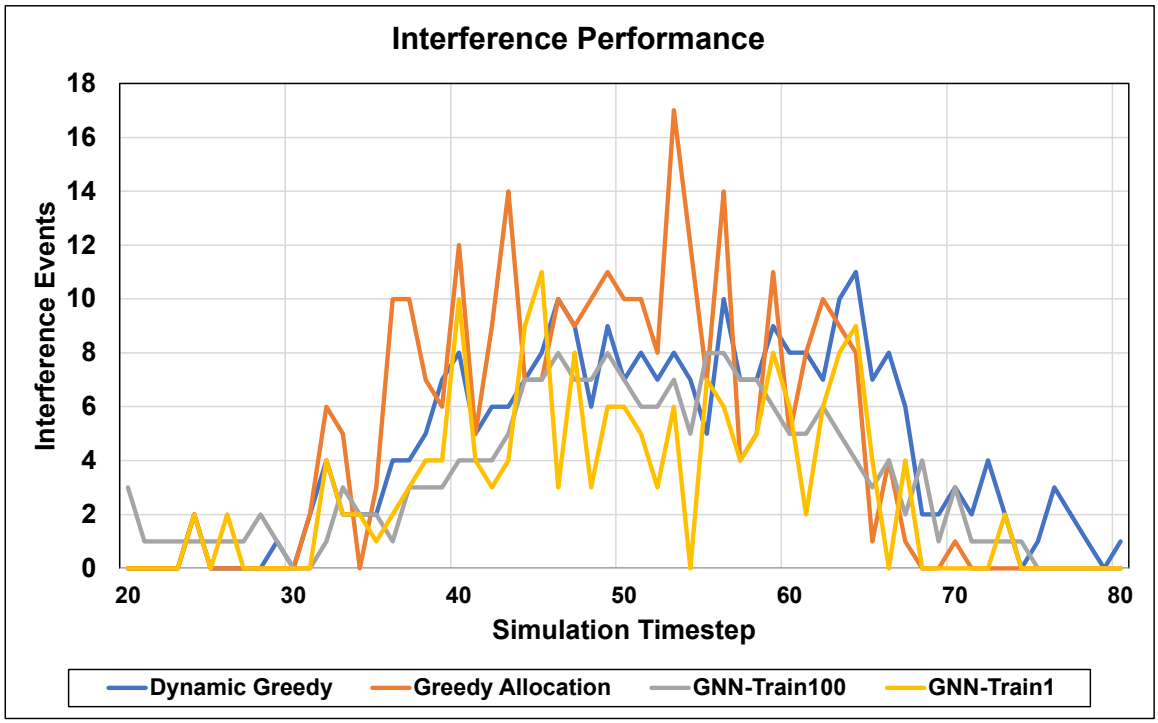


Figure 3.9: Interference events observed at each simulation timestep for different allocation methods. The plot compares greedy allocation, dynamic greedy, and two GCN-based approaches (GNN-Train100 and GNN-Train1).

Table 3.2: Summary of interference events for each allocation method. The table reports total events, mean events per timestep, and standard deviation for greedy, dynamic greedy, and GCN-based approaches.

Algorithm	**Total**	**Mean**	**Std Dev**
Greedy	290	2.871	19.142
Dynamic Greedy	273	2.703	11.496
GNN-Train100	244	2.416	5.926
GNN-Train1	178	1.762	7.706

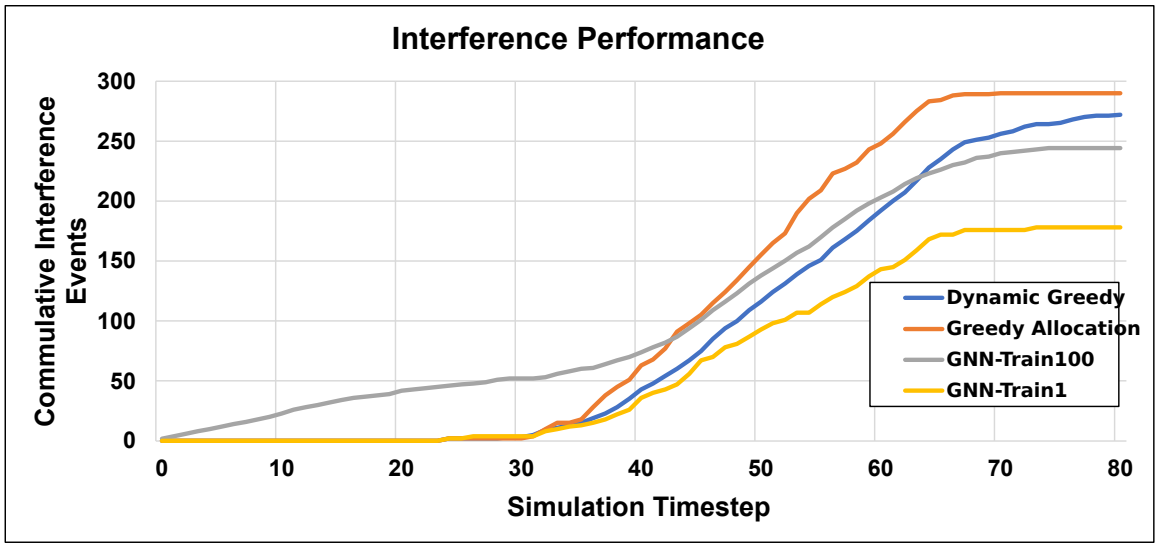


Figure 3.10: Cumulative interference events across all simulation timesteps for each allocation method. Lower cumulative values indicate better spectrum allocation performance.

3.5.3 Problem Simplification using Prediction

In order to simplify the graph provided to the resource allocation network, this work also considers the case where an aircraft is only assigned a channel when it needs to communicate. Rather than allocate channels to all aircraft in the simulation, the allocation function will only allocate resources to aircraft that are predicted to require communications. The use of machine learning [40] can provide insight into when an aircraft would need to communicate; statistical methods may be employed as well. However, in this study, a simple method for communication prediction is explored by examining the flight path of the aircraft with respect to airspace boundaries. Here, the assumption is that an aircraft only requires a channel assignment when transitioning between regions of airspace (to facilitate handovers). In the examined scenario, the airspace boundary crossing lies approximately 7.5 km from a

ground station, and aircraft that are greater than 6 km from a ground station are considered to require communications for a boundary-crossing event. Fig. 3.11 displays how the number of edges in the graph, i.e., the number of potential interference events, is reduced from the full airspace scenario.

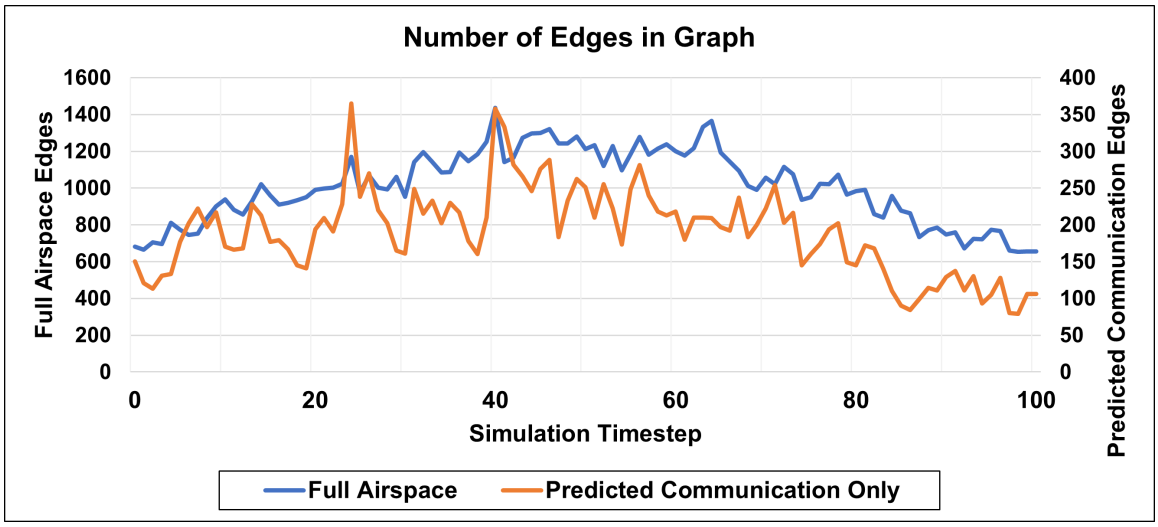


Figure 3.11: Comparison of graph complexity between full airspace and simplified predicted communication scenarios. The plot shows how prediction-based simplification reduces the number of edges, lowering potential interference.

The three allocation methods were re-examined for the resulting simplified adjacency matrix. In both the greedy allocation and the graph network that was trained for each simulation step, there are no instances of interference. However, the graph network that was trained using all simulation data continues to show instances of interference, albeit significantly fewer with only two instances of interference occurring at four simulation steps. Clearly, in situations where the aircraft density is lower and the resulting interference graphs have fewer edges, then fewer communication channels will be required to serve the needs the airspace system.

3.6 Observations

This study compared different allocation techniques to evaluate communications capacity in a hypothetical airspace scenario. Based on the simulations and their results, key items of observation are as follows:

- Performance: Graph coloring using the GCN learning-based method shows overall improved performance over greedy and dynamic greedy coloring, most notably in high density/demand use cases. However, in instances of low aircraft density, the greedy method outperformed the GCN that was trained over the entire time series. For this reason, there are instances where it may be best to dynamically select an allocation method based on available resources and complexity of the situation. Further, the number of channels available for assignment should vary depending on the anticipated density of the air traffic, since nearly 40% of the simulation showed no instances of interference and therefore resulted in a surplus of available channels that could be used elsewhere in the system.

- Transfer Learning: Neural networks and machine learning models are typically trained for specific tasks starting from randomly initialized parameters. However, when exposed to new datasets, some neural networks may need to be retrained to achieve the desired results, which contrasts with human learning in that previous knowledge can be applied to new contexts. This ability to reuse what was learned in previous settings is called transfer learning. In neural net-

works, transfer learning can be achieved by reusing the weights of a previously trained network. Transfer learning has been demonstrated to be very successful in traditional deep learning contexts, leading to faster training times and enhanced performance. Recent studies have also shown that transfer learning is effective with GNNs [60] [61] and the simulation results of this study tend to show that the GCN trained over the entire 100 simulation timesteps was able to perform reasonably well as compared to the greedy method, particularly in the high air traffic density time periods.

- Structured Airspaces: UAM operations are expected to use designated air traffic corridors with specific tracks and waypoints to manage traffic flow and ensure separation [62]. This could create consistent traffic patterns across the airspace, and could provide a bounded number of airspace representations which could potentially allow neural network models to be trained and optimized based on the predicted traffic patterns. However, structured UAM airspace configurations and separation standards are still in formulation.

- Dynamic Greedy Coloring: The dynamic greedy method, as previously described, demonstrated improved performance over the standard greedy approach by reducing assignment conflicts and maintaining most assignments across timesteps. This leads to a more efficient implementation in practice, as fewer reassignment messages need to be distributed to nodes within the network. Figure 3.12 illustrates channel assignments at timesteps 57, 58, and 59—some of the most congested periods in the simulation. As shown, only a small

number of nodes required reassignment. Overall, the dynamic greedy method outperformed both the greedy method and even the GNN-based method that had been trained on the entire time series, in terms of minimizing assignment conflicts.

3.6.1 Potential Follow-on Studies

Possible follow-on efforts may include additional constraints in the defined graphs such as transmitter power, aircraft velocity, etc. Methods of introducing time dynamics into the graph neural network may be explored along with more sophisticated communication prediction algorithms to reduce the complexity of the graph structure. Further analysis in a variety of different configurations of structured airspaces and corridors could aid in the development of AAM communications by enabling more efficient use of aeronautical spectrum resources.

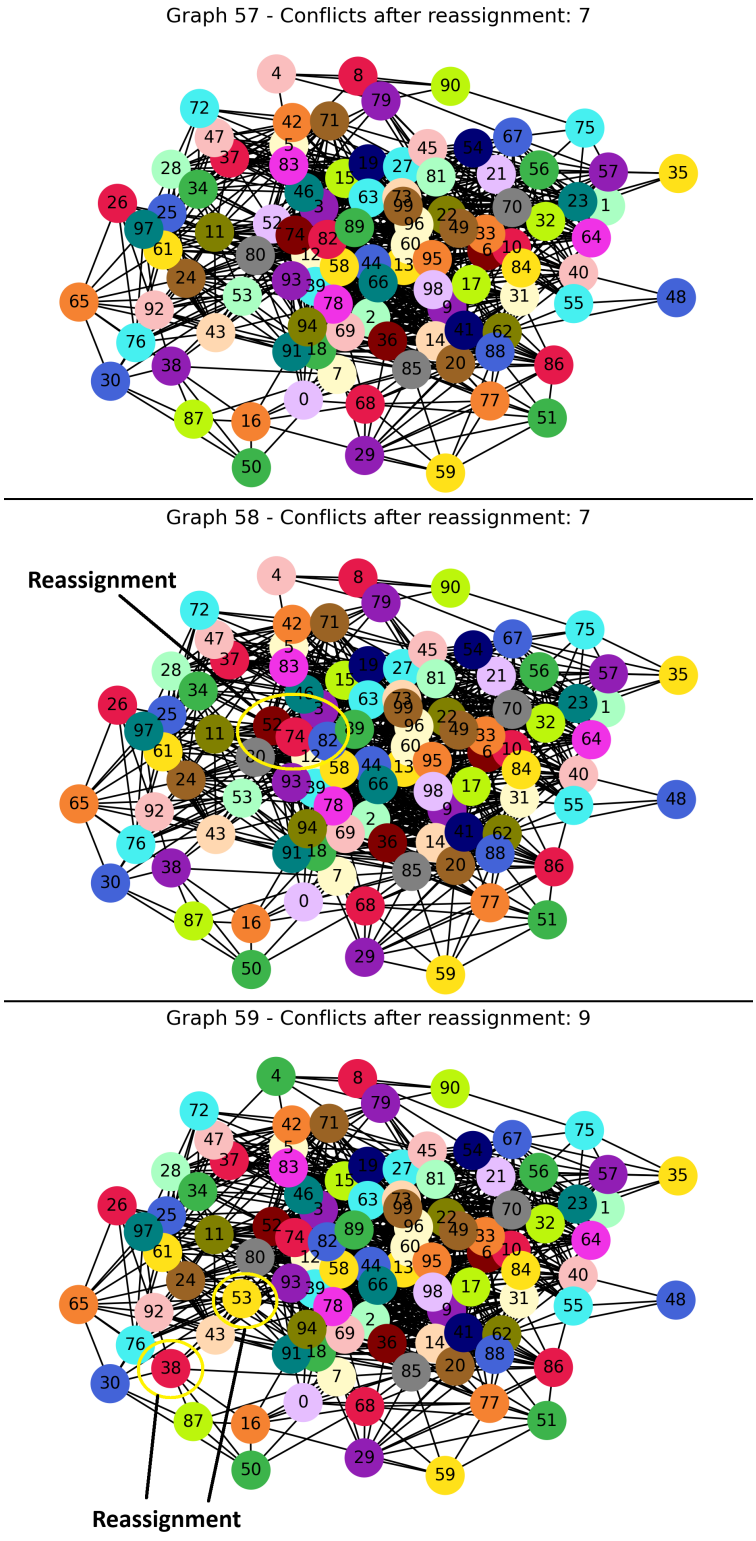


Figure 3.12: Channel assignment results for three consecutive timesteps during peak airspace congestion. The figure illustrates how dynamic greedy coloring minimizes conflicts compared to other methods.

CHAPTER IV

RESOURCE ALLOCATION USING REINFORCEMENT LEARNING

This chapter investigates the coordinated management of constrained spectrum and airspace resources by leveraging modern multi-agent reinforcement learning techniques. In the RL framework, an agent interacts with its environment through sequential policy-driven actions and receives feedback from the environment regarding the effectiveness of these actions. This feedback, comprising rewards and observational data from the environment, is essential for the agent's learning process, where the goal is for the agent to iteratively learn an optimal policy that maximizes the accumulated reward. Similarly, multi-agent RL focuses on sequential decision-making problems to optimize a learned policy, but involves multiple agents working interactively within the same environment. In particular, both the evolution of the system state and the reward received by each agent are influenced by the joint actions of all agents [63, 64].

The previous section investigated a UAM communications scenario where a single centralized agent (e.g., graph neural network) observed the state of the airspace and assigned channels to aircraft to minimize aircraft communications interference. This follow-on study also investigates a UAM communications scenario, but implements a distributed decentralized approach where multiple agents work cooperatively

to jointly select not only channel assignment but also vehicle velocity to minimize total flight travel time while maintaining the required signal-to-interference-plus-noise ratio (SINR) and aircraft separation requirements. This investigation compares a state-of-the-art cooperative multi-agent policy-based method, specifically proximal policy optimization (PPO) [65] in a multi-agent implementation (MAPPO) [66], to identify an optimal solution to the joint resource time minimization problem. The results obtained from this approach are compared with QMIX, a multi-agent value-based method [67]. This effort examines whether a joint resource evaluation strategy can yield benefits in overall operations performance with improved resource utilization efficiency. The focus is to investigate the system's ability to find an optimal solution that effectively manages spectrum resource utilization, ensures aircraft maintain separation, and minimizes flight travel time.

4.1 System Model and Problem Formulation

4.1.1 System Model

This study considered a UAM scenario in which multiple aircraft travel en-route from an origin vertiport to a destination vertiport, as shown in Fig. 4.1. The network consists of a finite set of aircraft, $\mathcal{N} = \{n_1, ..., n_N\}$ operating in an airspace system. For safe operations, two-way reliable communication between the aircraft and base stations is required. Each aircraft communicates with a set of base stations $\mathcal{B} = \{b_1, ..., b_M\}$ using bidirectional communications, and spectrum reuse is employed such that a finite set of available frequency channels $\mathcal{F} = \{f_1, ..., f_K\}$

may be simultaneously shared by multiple users, provided the interference is maintained within acceptable levels. Each aircraft's 3-D position vector is represented by $\mathbf{x}_n(t) = [x_n(t), y_n(t), z_n(t)]^T$, and the set of all aircraft position vectors is $\mathcal{X} = \{\mathbf{x}_1(t), ..., \mathbf{x}_N(t)\}$. The velocity of each aircraft is represented by $\dot{\mathbf{x}_n}(t)$, and the set of all aircraft velocity vectors is $\mathcal{V} = \{\dot{\mathbf{x}_1}(t), ..., \dot{\mathbf{x}_N}(t)\}$. The network is modeled using discrete time steps denoted by $T = \{1, 2, ..., T\}$ of duration ΔT. The notations used for the system model and the problem formulation are listed in Table 4.1.

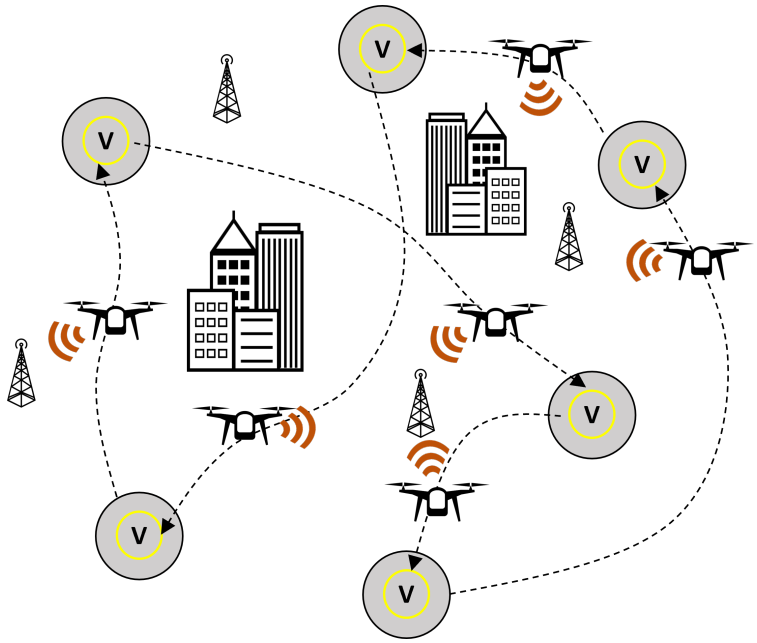


Figure 4.1: System model for an Urban Air Mobility scenario. The diagram shows aircraft traveling between vertiports and communicating with base stations using shared frequency channels, forming the basis for reinforcement learning-based resource allocation.

It is assumed that all in-flight aircraft traversing a UAM corridor at its assigned cruise altitude will have an unobstructed signal path between the antennas of

Table 4.1: Summary of key notation used in the reinforcement learning system model. The table defines variables for aircraft positions, velocities, channel assignments, and other parameters relevant to optimization.

Notation	
Set of all aircraft	$\mathcal{N} = \{n_1, ...n_N\}$
Aircraft 3-D position vector	$\mathbf{x}_n(t) = [x_n(t), y_n(t), z_n(t)]^T$
Set of all aircraft position vectors	$\mathcal{X} = \{\mathbf{x}_1(t), ..., \mathbf{x}_N(t)\}$
Set of all aircraft velocity vectors	$\mathcal{V} = \{\dot{\mathbf{x}_1}(t), ..., \dot{\mathbf{x}_N}(t)\}$
Aircraft n_n position at start of flight path	$\mathbf{x}_n(0)$
Aircraft n_n position at end of flight path	$\mathbf{x}_n(T_n)$
Distance between aircraft n_n and n_m	$D_m = \|\mathbf{x_n} - \mathbf{x_m}\|$
Aircraft minimum separation	$d_{\min}$
Set of all M base stations	$\mathcal{B} = \{b_1, ..., b_M\}$
Set of all K available channels	$\mathcal{F} = \{f_1, ..., f_K\}$
Aircraft n_n channel assignment	$\mathbf{c}_n$
Set of all aircraft channel assignments	$\mathcal{C} = \{\mathbf{c}_1, ..., \mathbf{c}_N\}$
Aircraft maximum speed	$V_{\max}$

the aircraft system and at least one base station antenna. For optimal signal strength, aircraft n_n connects with base station b_m which is situated at the shortest longitudinal distance and therefore represents the closest associated base station. The distance between base station b_m and aircraft n_n is referred to as $d_{n,m}$. The uplink trans-

mission SINR is calculated using (4.1), where p_n^u is the uplink power of the aircraft transmitter and p_j^d is the interference produced by other aircraft transmitters on the same frequency channel [68] [43].

$$\gamma_n^u(t) = \frac{p_n^u d_{n,m}^{-\delta}(t)}{\sum_{j \neq n} p_j^u d_{j,m}^{-\delta}(t) + N_0} \tag{4.1}$$

Similarly, the downlink transmission SINR is calculated using (4.2), where p_n^d is the downlink power of the base station transmitter and p_j^d is the interference produced by other base station transmitters on the same frequency channel.

$$\gamma_n^d(t) = \frac{p_n^d d_{n,m}^{-\delta}(t)}{\sum_{j \neq n} p_j^d d_{j,m}^{-\delta}(t) + N_0} \tag{4.2}$$

The transmit powers for all aircraft and base stations were assumed to be fixed real values: $p_n^u \in \mathbb{R}$ and $p_n^d \in \mathbb{R}$, respectively. The path loss exponent $\delta \in \mathbb{R}$ is a product of the propagation environment, and the noise term N_0 is the additive white Gaussian noise (AWGN) power spectral density. Equations (4.1) and (4.2) show that the SINR depends on transmitter power, distance, channel selection, and path loss. While aircraft transmit powers may vary, FAA engineering practices typically assume uniform EIRP across aircraft [42]. The path loss exponent varies with the environment, with urban values ranging from 2.1–3.5 based on the flight phase [44] [69]. Among these parameters, only the distance and channel selection can be dynamically adjusted by an intelligent agent to jointly optimize SINR and flight time. Because the flight paths are predefined, distance control is achieved by adjusting aircraft velocity at each time step.

4.1.2 Problem Formulation

The goal of this study was to minimize the total time required for all aircraft to complete their assigned flight plans. To achieve this, each aircraft must optimize its velocity selection, $\dot{\mathbf{x}}_n(t)$ and channel assignment $\mathbf{c}_n$, while maintaining continuous link connectivity and meeting the minimum SINR requirements. The minimization of the total flight completion time is formulated using the following objective function:

$$
\begin{aligned}
\text{(P1):}\ \min_{\mathcal{V},\mathcal{F}} & \left[\sum_{n} T_n\right], \\
\text{subject to}\quad & \text{C1: } \gamma_n^u(t) \geq \text{SINR}_{\min}^u, \quad \forall n_n \in \mathcal{N}, \\
& \text{C2: } \gamma_n^d(t) \geq \text{SINR}_{\min}^d, \quad \forall n_n \in \mathcal{N}, \\
& \text{C3: } \sum_{k=1}^{K} c_{n,k}^u = 1, \forall n_n \in \mathcal{N}, \\
& \text{C4: } \sum_{k=1}^{K} c_{n,k}^d = 1, \forall n_n \in \mathcal{N}, \\
& \text{C5: } \mathbf{x}_n(0) = \mathbf{x}_n^0, \forall n_n \in \mathcal{N}, \\
& \text{C6: } \mathbf{x}_n(T_n) = \mathbf{x}_n^F, \forall n_n \in \mathcal{N}, \\
& \text{C7: } \dot{\mathbf{x}}_n(t) < V_{\max}, \forall n_n \in \mathcal{N}, \\
& \text{C8: } D_m \geq d_{\min}, \forall m \neq n.
\end{aligned}
\tag{4.3}
$$

The objective function minimizes the total flight travel time for different combinations of velocities and channel selection for all aircraft in the UAM scenario. Constraints, C1 and C2 ensure that all aircraft $n_n \in \mathcal{N}$ comply with the minimum signal quality threshold for both uplink and downlink communications, respectively. These constraints illustrate the relationship between aircraft position using a selected

velocity from $\mathcal{V}$ and the assigned channel from $\mathcal{C}$ in the SINR calculation. Constraints C3 and C4 enforce that each aircraft can only select a single uplink and downlink channel at any time, respectively. Constraint C5 denotes each aircraft's starting coordinate location for departure, and constraint C6 denotes the aircraft's final location once arriving at its destination. Constraint C7 enforces each aircraft's speed limitation, and C8 enforces airspace safety separation that must be maintained under all flight operations.

4.2 Reinforcement Learning Solution

In the UAM environment, decisions made for each aircraft are influenced by the actions of all other aircraft in the region. This makes the optimization problem defined in (P1) a Markov game that extends the Markov decision process to multiple interacting agents. Particularly, this implementation is referred to as a decentralized fully observable Markov decision process with shared rewards [70], and for N agents, the process is defined by four elements: state space (S), action space (A), reward function (R), and state transition probability (P). At each time step, an agent observes the current state s_n and chooses an action a_n. The new state s'_n is determined by the state transition probability $P(s'_n \mid s_n, a_1, a_2, \ldots, a_N)$. The environment then provides a reward, r based on the reward function, R, which measures how well the agent's goal is achieved. Agents employ a policy denoted as $\pi_\theta(a_i \mid o_i)$ and parameterized by θ, to generate action a_i based on observation o_i. Together, they

aim to maximize the discounted cumulative reward $J(\theta)$. The components of the decision-making process are as follows:

- **State** S: Owing to the existence of multiple aircraft in the UAM system, each agent has its own state observations. For aircraft agent n_n, its observed state s_n includes its self-flight completion status as well as the locations of nearby aircraft.

- **Action** A: At every time step, the agent selects both the aircraft velocity and channel assignment. Accordingly, the joint action for agent n_n can be denoted as $a_n = (a_{v,n}, a_{c,n})$, where $a_{v,n}$ and $a_{c,n}$ represent the velocity and channel-selection actions, respectively. Continuous velocity selection is discretized into L speed levels, such that both velocity and channel selections are integrated into a single discrete action space. With N agents, each of which may choose between L speed levels and K channels, each of the N agents has a total of $L \times K$ available actions at its disposal.

- **State Transition Probability** P: The state transition function f, generates the next-state progression in response to agent-supplied actions. Once aircraft n_n departs its vertiport and is in-flight, its next position is defined by $\mathbf{x}'_n = \mathbf{x}_n + \Delta T a_{v,n}$, where the next state is determined by $s'_n : f(\mathbf{x}'_n, \mathbf{x}'_j)$, and the state transition representation is $P : s_n \to s'_n$. Function $f : (\mathbf{x}_n, \mathbf{x}_j) \to s_n$ maps the aircraft position to state.

- **Reward Function** R: In the decision process, the reward function $R : S \times A \rightarrow \mathbb{R}$ maps state-action pairs to scalar rewards r, which provides an indication of how well the agent has achieved its goal. For this UAM system model, the reward at each time step includes 1) a penalty for violating the minimum communications quality-of-service requirement, 2) a penalty for violating the minimum aircraft separation requirement, and 3) a penalty for waiting in place (i.e., hovering) while airborne.

The following sections describe the two multi-agent deep reinforcement learning algorithms used to solve the $(P1)$ optimization problem in the UAM system model described previously. The first algorithm is a multi-agent implementation of the PPO algorithm, which is a policy-based learning algorithm, and QMIX, which is a value-based learning algorithm.

4.3 Multi-Agent PPO

4.3.1 Overview

PPO is a modern reinforcement learning algorithm in the policy gradient family. Policy gradient methods are a type of reinforcement learning algorithm that directly optimizes the policy, that is, the mapping from states to actions, by performing gradient ascent on the expected rewards. Unlike value-based methods, such as Q-learning, which learn a value function and derive a policy from it, policy gradient methods learn the policy directly.

The PPO algorithm alternates between collecting experience from the environment and optimizing a surrogate objective via stochastic gradient ascent. Unlike traditional policy gradient methods, which typically perform a single update per data point, PPO enables multiple epochs of minibatch updates. This approach retains some of the stability benefits of Trust Region Policy Optimization (TRPO) [71] while being simpler to implement, more versatile, and often more sample-efficient, allowing the method to learn effective policies with fewer environment interactions. This can be particularly advantageous in reinforcement learning scenarios, where collecting data (e.g., running simulations or real-world trials) can be expensive or time-consuming. Empirical results on benchmark tasks, such as simulated robotic locomotion and Atari games, demonstrate that PPO outperforms other online policy gradient methods, offering a balance between sample efficiency, simplicity, and training time [65].

As an on-policy algorithm, the PPO method involves a sampling and training loop at each iteration. During sampling, rollouts are generated from agent-environment interactions using the current policy. These rollouts are then used immediately in the training phase to update the policy through backpropagation. A critic network estimates the value of the states (expected discounted return), which is used to compute the advantages of the actions taken. The PPO loss function compares the actual returns to these estimates and guides policy updates to improve the performance. In traditional policy gradient methods, large policy updates can lead to performance collapse. PPO addresses this by restricting the extent to which the new

policy can deviate from the old one during each update. PPO modifies the objective function using a clipped surrogate objective.

$$L^{\text{CLIP}}(\theta) = \mathbb{E}_t \left[\min \left(r_t(\theta)\hat{A}_t,\ \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t \right) \right] \tag{4.4}$$

Where $r_t(\theta) = \pi_\theta(a_t|s_t)/\pi_{\theta\text{old}}(a_t|s_t)$ is the probability ratio between the new and old policies, $\hat{A}_t$ is the advantage estimate at time t, and ϵ is a small hyperparameter (on the order of 0.1 or 0.2). The clip function ensures that $r_t(\theta)$ remains within the range of $[1-\epsilon, 1+\epsilon]$, and the primary concept behind the clipping function is that if the new policy improves the advantage, $\hat{A}_t$, too much, the clipping prevents the update from being too extreme. If the update is within the nominal range, it proceeds normally. This prevents the policy from changing drastically, which helps maintain training stability and sample efficiency. In general, the advantage function $A^\pi(s,a)$ is defined as

$$A^\pi(s,a) = Q^\pi(s,a) - V^\pi(s) \tag{4.5}$$

Where $Q^\pi(s,a)$ is the expected return (cumulative reward) starting from state s, taking action a, and then following policy π. $V^\pi(s)$ is the expected return starting from state s and following policy π regardless of the action taken. Clearly, if $A^\pi(s,a) > 0$, action a is better than the average in state s. If $A^\pi(s,a) < 0$, action a performs worse than the average, and if $A^\pi(s,a) = 0$, action a performs as well as the average. This helps the policy gradient algorithm focus on updates based on actions that are better than expected, thereby improving learning efficiency. Because the

true values of Q^{π} and V^{π} are typically not known, the advantage can be estimated by

$$\hat{A}_t = \delta_t + (\gamma\lambda)\delta_{t+1} + (\gamma\lambda)^2\delta_{t+2} + \dots \tag{4.6}$$

where $\delta_t = r_t + \gamma V(s_{t+1}) - V(s_t)$ is the temporal difference (TD) error, γ is the discount factor, and λ is a smoothing parameter. The PPO method uses the advantage estimate $\hat{A}_t$ in its clipped objective function to determine how much to increase or decrease the probability of taking action a_t in state s_t which makes the learning more targeted and stable.

4.3.2 MAPPO Implementation

The implementation of the multi-agent PPO algorithm used in this case study is described in Algorithm 4, where an aircraft operation starts at departure vertiport v_s and arrives at destination vertiport v_d, which traverses from source to destination and constitutes an episode, also known as an epoch. A set of $\hat{N}_{ep}$ episodes were executed, and each episode required a finite number of time steps to complete. Simulation information, such as boundary parameters, buffer sizes, weight parameters, and infrastructure location, was provided as input to the simulation system. The simulation outputs consist of trained weight parameters for the actor and critic networks. Line 1 initializes the actor θ_a and critic θ_c weight parameters. Line 2 executes the maximum number of aircraft operation episodes, $\hat{N}_{ep}$. Line 3 executes minibatch episodes used for actor-critic training. Lines 4-5 initialize the time steps and aircraft departure information. Line 6 executes aircraft flight procedures for all

participating aircraft n. Line 7-13 obtains environment observations s_n utilizing a mapping function f, actor action a_n and softmax probability evaluations, reward r_n calculation, evaluates critic output $V(s)$ and increments time. Lines 14-16 prepare the information for network training, where $\hat{N}_b$ transitions are retrieved from buffer $\mathcal{D}$, temporal difference error δ_t and advantage $\hat{A}_t$ are calculated. Lines 17-22 execute $\hat{T}_i$ training iterations for actor and critic networks, where ratio r_t is calculated, actor loss function L_a evaluation is selected and actor network training is conducted resulting in parameters θ_a; similarly, critic loss L_c is calculated yielding parameters θ_c. The actor and critic networks had three and two hidden layers, respectively, and both were configured to use the Adam optimizer with a learning rate of 0.001.

4.4 QMIX

4.4.1 Overview

QMIX is a multi-agent reinforcement learning method designed for cooperative tasks. It enables centralized training with decentralized execution by decomposing a global action value function, Q_{tot}, into individual agent value functions Q_n. A mixing network ensures that Q_{tot} is monotonic with respect to each Q_n, allowing agents to act independently while still optimizing the team performance. This structure is particularly effective for scenarios that require tight coordination between agents [67].

Algorithm 4: Policy Optimization with Actor-Critic Learning.

Output: Learned parameters θ_a, θ_c

Initialize θ_a, θ_c;

for $n_{ep} = 1$ **to** $\hat{N}_{ep}$ **do**

  for $n_b = 1$ **to** $\hat{N}_b$ **do**

    Set $t \leftarrow 0$;

    Reset AV locations: $\mathbf{x}_n(t) \leftarrow \mathbf{x}_n^0$;

    while $\mathbf{x}_n(t) \neq \mathbf{x}_n^F \ \forall n \in \mathcal{N}$ *and* $t \leq \hat{N}_{step}$ **do**

      for $n = 1$ **to** N **do**

        Get state $s_n \leftarrow f(\mathbf{x}_n(t))$;

        Actor output $a_n \sim \pi_\theta(a_t|s_t)$;

        Perform a_n, observe s'_n, update $\mathbf{x}_n$;

        Compute reward r_n, critic output $V(s_n)$;

      $t \leftarrow t + 1$;

      Store $(\mathbf{a}, \mathbf{s}, r_T, \mathbf{s}')$ in buffer $\mathcal{D}$;

    Sample $\hat{N}_b$ transitions from $\mathcal{D}$;

    Compute $\delta_t = r_t + \gamma V(s_{t+1}) - V(s_t)$;

    Compute $\hat{A}_t = \sum_{l=0}^{\infty} (\gamma\lambda)^l \delta_{t+1}$;

    for $t_i = 1$ **to** $\hat{T}_i$ **do**

      $r_t = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)}$;

      $L_a = \hat{\mathbb{E}}_t \left[\min \left(r_t \hat{A}_t,\ \text{clip}(r_t, 1-\epsilon, 1+\epsilon) \hat{A}_t \right) \right]$;

      $\theta_a \leftarrow \theta_a - \eta \nabla_\theta L_a$;

      $L_c = \frac{1}{N} \sum (A_t + V_\theta(s_t) - V_\theta(s_{t+1}))^2$;

      $\theta_c \leftarrow \theta_c - \alpha \nabla_\theta L_c$;

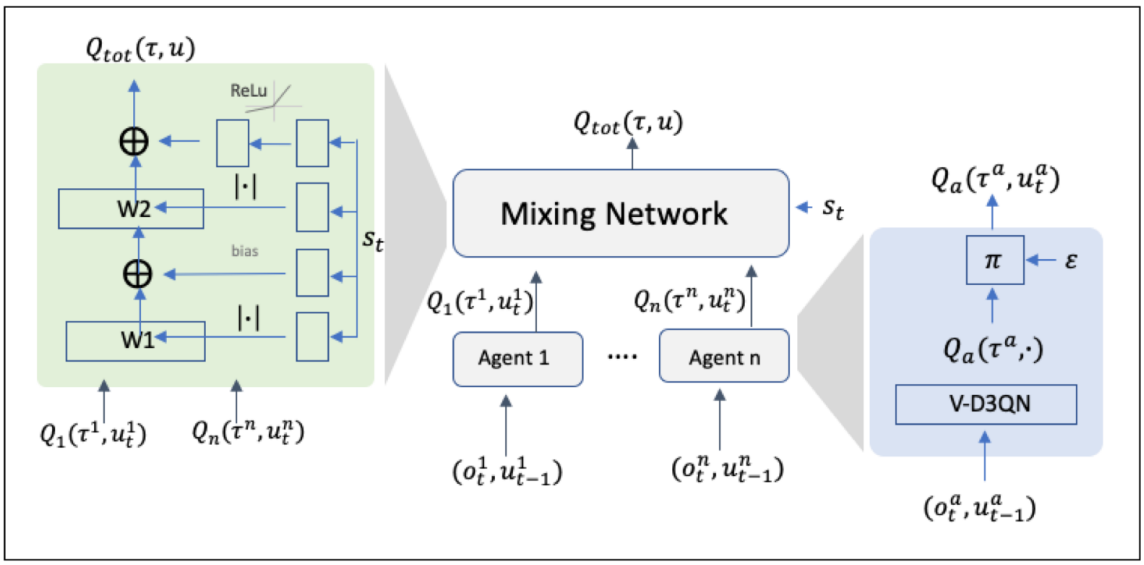


Figure 4.2: Architecture of the QMIX algorithm integrated with V-D3QN agent networks. The figure illustrates how individual agent Q-values are combined through a mixing network to produce a joint Q-value for cooperative decision-making.

4.4.2 QMIX Mathematical Formulation

In the QMIX framework, each agent n maintains its own individual Q-function, denoted as $Q_n(\tau^n, u^n)$, where τ^n represents the agent's observation and u^n is the action taken by the agent. These individual Q-functions are then integrated into a global perspective through a joint Q-function, $Q_{tot}(\boldsymbol{\tau}, \boldsymbol{u})$, which depends on the collective observations $\boldsymbol{\tau} = (\tau^1, \tau^2, \ldots, \tau^n)$ and actions $\boldsymbol{u} = (u^1, u^2, \ldots, u^n)$ of all the agents. To achieve this integration, QMIX employs a mixing network that combines individual Q-values into a joint Q-value by using the function f_{mix}, parameterized by θ. This relationship is expressed as follows:

$$Q_{tot}(\boldsymbol{\tau}, \boldsymbol{u}) = f_{mix}(Q_1(\tau^1, u^1), Q_2(\tau^2, u^2), \ldots, Q_n(\tau^n, u^n); \theta) \tag{4.7}$$

The mixing network f_{mix} was implemented as a neural network and specifically designed to satisfy a monotonicity constraint. This constraint ensures that the joint Q-value is a monotonic function of each individual Q-value, which is formalized as:

$$\frac{\partial Q_{tot}}{\partial Q_n} \geq 0 \quad \forall n \tag{4.8}$$

This property guarantees that increasing the Q-value of an individual agent does not decrease the overall joint Q-value. As a result, it becomes feasible to optimize each agent's Q-function independently while still contributing positively to the global objective.

4.4.3 Hypernetwork Definition

QMIX introduces a hypernetwork to ensure that the joint Q-function Q_{tot} remains monotonic with respect to each agent's individual Q-function $Q_n(\tau^n, u^n)$. This design allows the optimal joint action to be derived by independently maximizing individual Q-values, thereby enabling an efficient decentralized execution. The hypernetwork generates the weights and biases for the mixing network based on the global state s_t. Formally, the mixing network parameters θ are produced as follows:

$$\theta = h(s_t; \phi) \tag{4.9}$$

where h is the hypernetwork parameterized by ϕ. Specifically, it outputs weights w_{ij} and biases b_i for the mixing network layers.

$$w_{ij} = h_{w_{ij}}(s_t; \phi), \quad b_i = h_{b_i}(s_t; \phi) \tag{4.10}$$

4.4.4 Loss Function and Training

The QMIX method is trained by minimizing the squared temporal difference error between the predicted joint Q-value and target value derived from the V-D3QN architecture [72]:

$$L(\theta) = \mathbb{E}\left[\left(y^{V-D3QN} - Q_{tot}(\boldsymbol{\tau}, \boldsymbol{u}; s, \theta)\right)^2\right] \tag{4.11}$$

The target value is computed as:

$$y^{V-D3QN} = r + \gamma \cdot \max_{\boldsymbol{u'}} Q_{tot}(\boldsymbol{\tau'}, \boldsymbol{u'}, s'; \theta^-) \tag{4.12}$$

where r is the reward, γ is the discount factor, and θ^- are parameters of the target network.

Fig. 4.2 illustrates the QMIX architecture, which consists of a set of n agent networks and a mixing network. Each agent network receives its full observation τ^n and previous action u^n_{t-1} to compute the local Q-value $Q_n(\tau^n, u^n)$. These individual Q-values are then passed to the mixing network, which uses hypernetwork-generated weights and biases to compute the joint Q-value Q_{tot}. During the centralized training, the full state was used to guide the mixing process. In decentralized execution, agents act based on their local observations using the learned individual Q-functions. Our implementation leverages the V-D3QN architecture for online execution, whereas centralized training ensures coordination through the joint Q-function.

4.5 Simulation and Case Study

A series of simulation experiments was conducted to investigate the effectiveness of PPO with MAPPO for solving the MARL challenge. This section describes the hypothetical UAM model used for the simulations, the case study description, and the experimental results.

4.5.1 Environment Model

The environment for this analysis is illustrated in Fig. 4.3 and represents a UAM scenario comprising multiple aerial vehicles traversing between vertiports and using a set of base stations for air-to-ground communications. The proposed environment provides a means of training and evaluating the learned behavior of intelligent agents. This model consists of six aircraft, $\mathcal{N} = \{n_1, n_2, ..., n_6\}$, and six vertiports, $\mathcal{P} = \{v_1, v_2, ..., v_6\}$. The UAM model was configured such that the nodes were equidistant from each other. The network has a total of four communication base stations, denoted by set $\mathcal{B} = \{b_1, b_2, b_3, b_4\}$, deployed at arbitrary locations, and each base station antenna structure has a height of H_b. A total of four frequency channels denoted by set $\mathcal{F} = \{f_1, f_2, f_3, f_4\}$ are used. Fig. 4.4 shows the flight profile for the trajectory between vertiports v_6 and v_2, where the trajectory distance was 1500 units and the flight altitude was 150 units.

4.5.2 Simulation Parameters and Configuration

In this case study, we evaluate the ability of MARL agents to determine the optimal policy for various environmental configurations. Table 4.2 presents the

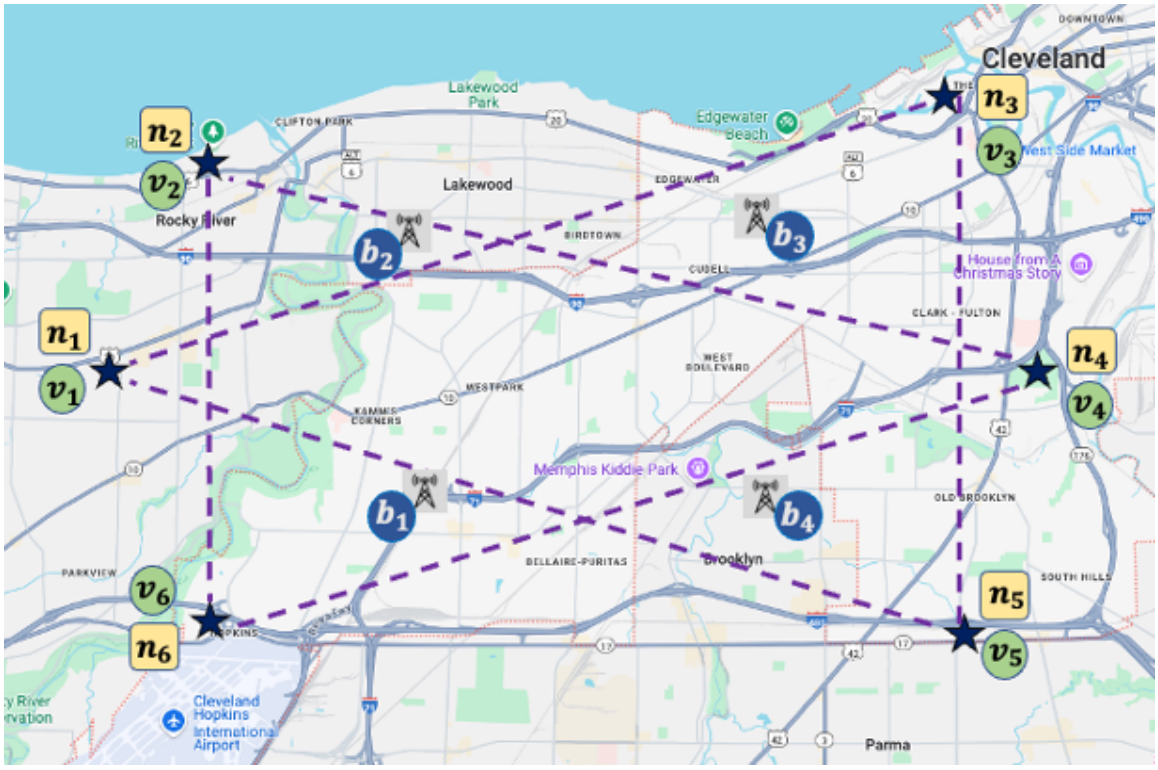


Figure 4.3: Diagram of the Urban Air Mobility environment used for reinforcement learning simulations. The figure shows aircraft, vertiports, and base stations arranged to evaluate multi-agent resource allocation strategies.

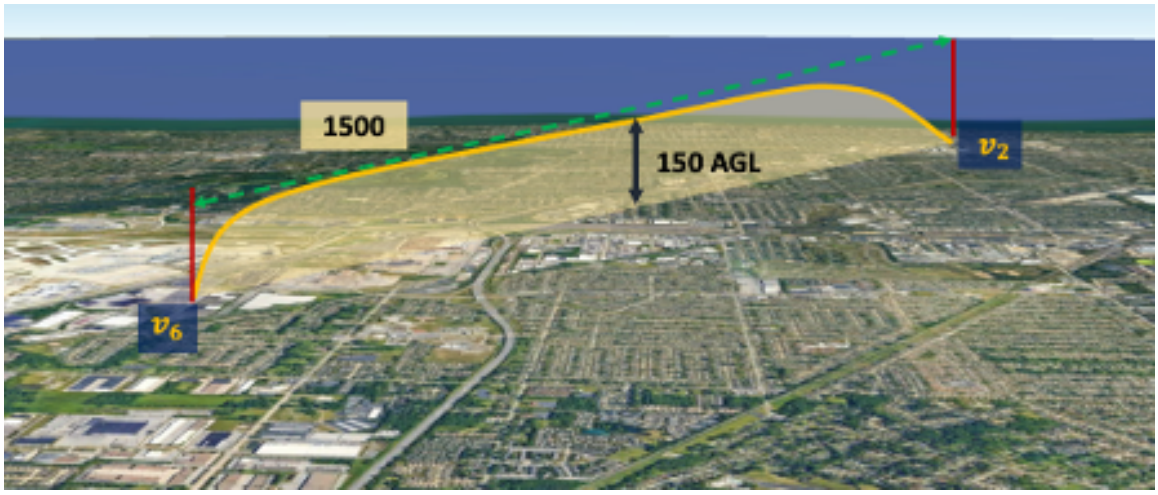


Figure 4.4: Flight corridor profile for the route between vertiports v_6 and v_2. The figure depicts trajectory distance and altitude used in simulation experiments.

configuration and simulation parameters used in the evaluation. The flight plan for each aircraft is given in Table 4.3, where the flight trajectory between the associated departure and arrival points is shown for all participating aircraft.

Table 4.2: Configuration parameters for the Urban Air Mobility simulation environment. The table lists vertiport coordinates, base station positions, communication settings, and propagation characteristics used in the study.

Model Parameters	**Value**
Vertiport Coordinates ($v_1 - v_6$)	[0, 750], [433, 1500], [1299, 1500], [1732, 750], [1299, 0], [433, 0]
Base Station Coordinates	[250, 375], [250, 1100], [1482, 1100], [1482, 375]
Base Station height H^b	15 units
Aircraft cruise altitude H^n	150 units
Aircraft velocity V (units/s)	[0, 25, 50]
Flight distance ($v_s - v_d$)	1500 units
Aircraft Tx Power P^u	30 dBm
Base Station Tx Power P^d	40 dBm
AWGN N_0	-96 dBm
Path loss δ	[2, 2.4, 2.7]
Pads per node C_v	1

Table 4.3: Flight plans for all aircraft in the simulation scenario. Each entry specifies the departure and destination vertiports for evaluating multi-agent resource allocation.

Aircraft	**Flight Plan**
n_1	$l_1(v_1, v_3)$
n_2	$l_2(v_2, v_4)$
n_3	$l_3(v_3, v_5)$
n_4	$l_4(v_4, v_6)$
n_5	$l_5(v_5, v_1)$
n_6	$l_6(v_6, v_2)$

4.6 MAPPO Performance Analysis

4.6.1 Hyper-parameter Tuning

Actor-critic methods, such as MAPPO, require hyperparameter optimization to improve system performance, including defining the number of episodes to run before training. Episode data were stored in memory and used to train the model. Different values were evaluated to determine the optimal number of episodes that yielded the best performance. The epoch hyperparameter, referred to as the mini-batch size (MB Size), represents the number of episodes executed before commencing the actor and critic training. Additionally, we identified the number of training cycles for the actor and critic networks as another critical hyperparameter, referred to as

Training Iterations (Train Iter), which specifies the number of times each actor and critic network was trained using the collected episode data.

Table 4.4: Hyperparameter settings for the MAPPO algorithm. The table includes values for episodes, minibatch size, training iterations, learning rate, and other parameters used in reinforcement learning experiments.

Model Parameters	**Value**
Max. episodes $\hat{N}_{\text{ep}}$	1000
Minibatch Size $\hat{N}_b$	4
Training Iterations $\hat{T}_i$	5
Discount factor α, η	0.0005
Learning rate	0.001
Smoothing factor λ	0.95
Clip parameter ϵ	0.2
Simulation Time	2 s
Time slot duration ΔT	1 s

The hyperparameter selection involves a series of simulations conducted in the UAM scenario. A controlled set of simulations was executed using the defined reward values and constraints. The constraints were set to 1) the SINR threshold value $\gamma_{\text{qos}} = 2$, and 2) the minimum separation $d_{min} = 100$. The reward values were set to 1) separation violation penalty = -50, 2) SINR violation penalty = -50, 3) and airborne waiting penalty = -50. A total of 32 simulation runs were performed with varying parameter values to assess the system performance.

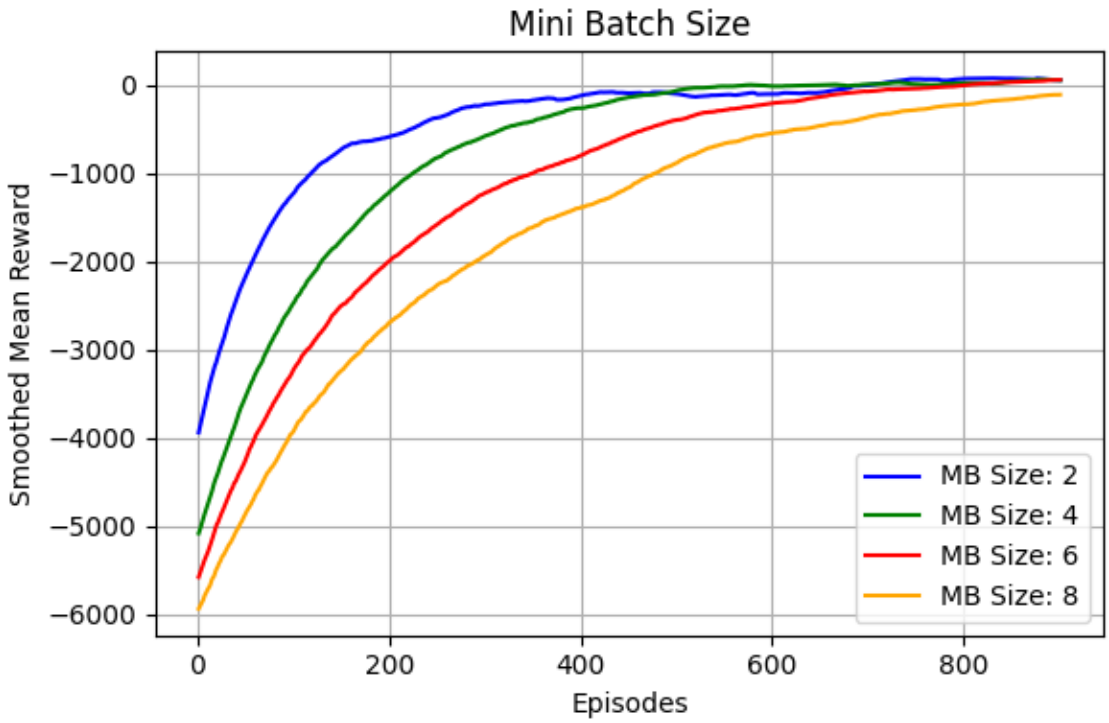


Figure 4.5: Performance comparison of different minibatch sizes in MAPPO training. The figure shows convergence trends for batch sizes 2, 4, 6, and 8 across training episodes.

As shown in Fig. 4.5, the system was evaluated using MB size of 2, 4, 6, and 8. The number of training iterations was fixed at 20 for both the actor and critic networks, and four evaluations were conducted for each MB Size value. While all evaluations converged, the blue and green traces in the figure showed faster convergence than the red and yellow traces. A closer inspection of MB Size = 4 reveals a slightly better performance than MB Size = 2, particularly beyond episode 500. Based on these findings, we selected an MB Size = 4 for network training.

For actor-critic training iteration tuning, we evaluated the Train Iter values from the set $\{5, 10, 15, 20\}$, as shown in Fig. 4.6. In this evaluation, the MB Size was

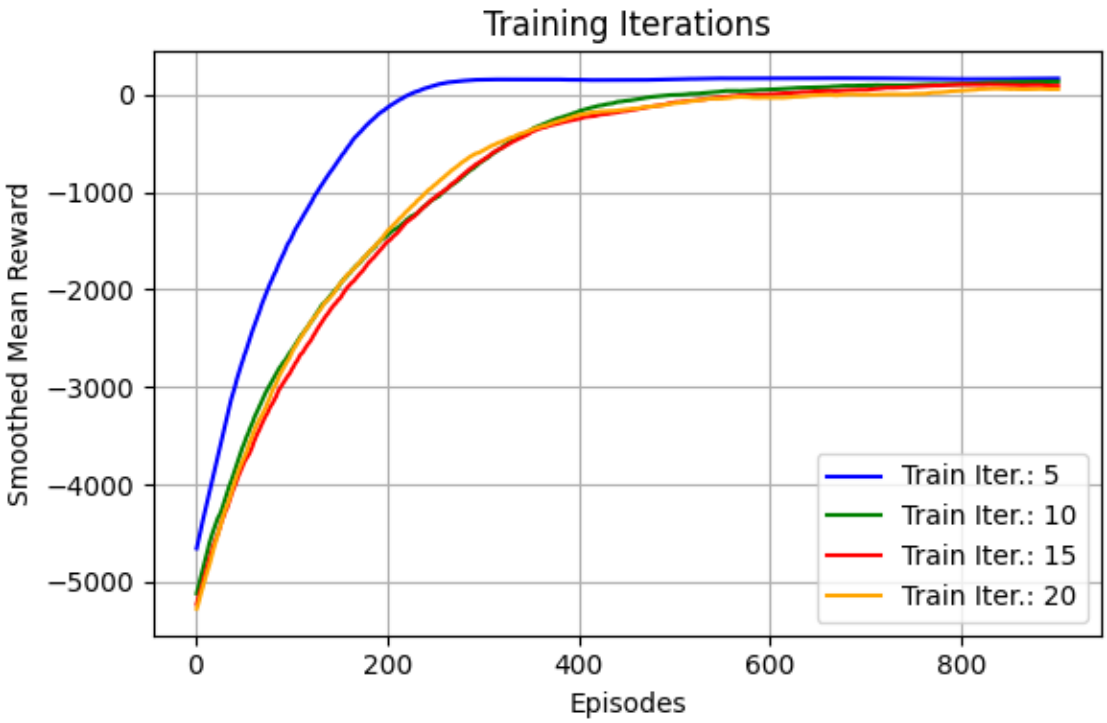


Figure 4.6: Effect of training iteration count on MAPPO convergence. The figure compares performance for 5, 10, 15, and 20 iterations, highlighting optimal settings for faster learning.

fixed at four, and four episode trials were conducted for each Train Iter value. The results indicate that the blue trace (Train Iter = 5) achieves convergence significantly faster than the other traces. Training the system for 10, 15, or 20 iterations did not yield an improved performance, with convergence occurring only after approximately 800 episodes. Based on these results, we selected Train Iter = 5 to train our system.

4.6.2 SINR Evaluation

Aviation spectrum depletion requires efficient frequency utilization across all flight phases. Efficiency can be improved by optimizing frequency reuse while min-

imizing harmful interference. This simulation investigated the performance of the actor-critic system under three different aircraft SINR threshold values, γ_{qos}. It evaluates the system's ability to maintain safe, interference-free communication between aircraft by selecting appropriate channels and applying speed adjustments.

In this test, the system was tasked with managing frequency assignments and reusing them while maintaining an interference-free environment. Specifically, it was evaluated for its capacity to assign communication channels and optimize reuse, considering the SINR threshold values γ_{qos} of 2.5, 4.5, and 6.5. Vehicle separation was fixed at $d_{min} = 100$ units, with reward assignments as follows: separation violation = -50, SINR violation penalty = -50, and airborne waiting penalty = -50. Fig. 4.7 shows the reward accumulation versus training epochs for three different SINR threshold values. A running average over the most recent 100 episodes is illustrated, where the blue, orange, and green traces represent SINR values of 2.5, 4.5, and 6.5, respectively.

As shown in the figure, when the SINR threshold was 2.5, the system accumulated higher rewards in fewer episodes, indicating a more efficient learning. When the threshold increased to 4.5, the system required approximately 400 episodes to converge. At an SINR value of 6.5, the system took even longer to converge compared with cases 2.5 and 4.5. This extended learning time at higher SINR values is expected owing to the increased communication quality-of-service requirements.

The frequency selection for each aircraft at each instance is illustrated in Fig. 4.8. As the aircraft executes its flight path, it changes the frequency assignments based on channel availability and γ_{qos} constraint. This figure illustrates how each fre-

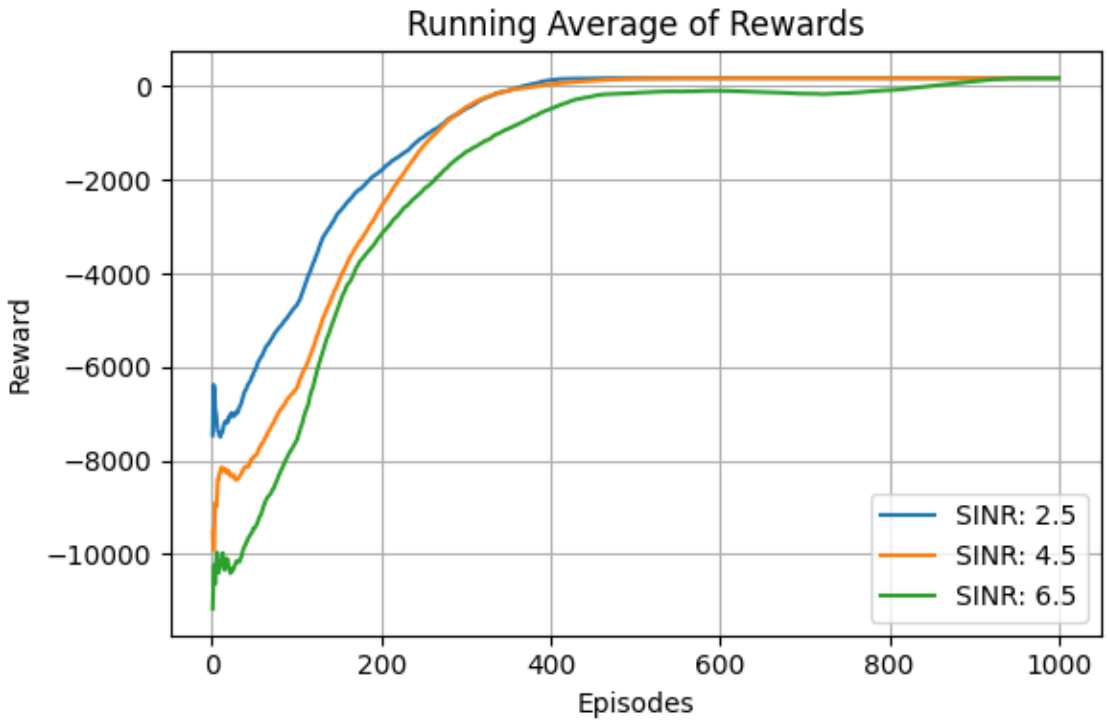


Figure 4.7: Accumulated reward trends for MAPPO under three SINR thresholds (2.5, 4.5, and 6.5). The figure illustrates how stricter SINR requirements affect learning speed and convergence.

quency is reused throughout the flight. The MAPPO method assigns a frequency to all aircraft, and the assignments adhere to the minimum SINR constraint parameter. The system was able to allocate channels without producing harmful co-channel interference for all aircraft operating in the airspace. Given that there are more aircraft than channel assignments, the system is required to conduct frequency reuse. Fig. 4.8 shows that at t_{13}, frequency f_4 is reused three times, and as shown in Fig. 4.9, frequency f_4 is used by aircraft n_2, n_3, n_6 at time t_{13}. This demonstrates the ability of the system to maximize spectral efficiency while minimizing interference.

Frequency Selection							Freq. Reuse			
t	n1	n2	n3	n4	n5	n6	f1	f2	f3	f4
t1	2	3	4	2	1	4	1	2	1	2
t2	2	3	4	2	1	4	1	2	1	2
t3	2	3	4	2	1	4	1	2	1	2
t4	2	3	4	2	1	4	1	2	1	2
t5	2	3	4	2	1	4	1	2	1	2
t6	2	3	4	2	1	4	1	2	1	2
t7	2	3	4	2	1	4	1	2	1	2
t8	2	3	4	2	1	4	1	2	1	2
t9	2	3	4	3	1	4	1	1	2	2
t10	2	3	4	3	1	4	1	1	2	2
t11	2	1	4	3	1	4	2	1	1	2
t12	2	1	4	3	1	4	2	1	1	2
t13	2	4	4	3	1	4	1	1	1	3
t14	2	3	4	3	1	4	1	1	2	2
t15		1					1	0	0	0

Figure 4.8: Frequency allocation and reuse patterns for MAPPO under SINR threshold 6.5 and minimum separation 100 units. The figure shows how channels are reassigned during flight to maintain communication quality.



Figure 4.9: Detailed view of frequency reuse at time step 13 for SINR threshold 6.5. The figure highlights multiple aircraft sharing the same channel without causing harmful interference.

4.6.3 Path Loss Exponent Evaluation:

In this test, we evaluated the learning performance of the system for different path loss exponent values. The path loss exponent δ from (4.1) and (4.2) quantifies how the received signal power between the aircraft and base station decays with distance owing to the propagation conditions in the airspace environment. A higher δ indicates more severe path loss, resulting in faster signal attenuation, reduced coverage area, and a more challenging interference situation. Aircraft at cruising altitudes are expected to experience propagation conditions close to free space, with δ values in the range of 2.0 to 2.1. When aircraft are near vertiports (i.e., during arrival or departure), δ may range between 2.4 and 2.8 [73] [69]. In our simulation, we evaluated the ability of the system to train under three path loss exponent values: 2.0, 2.4, and 2.8. The aircraft separation threshold was set to $d_{min} = 300$ units and the SINR threshold was $\gamma_{\text{qos}} = 4.5$.

Fig. 4.10 shows the learning performance under three path loss conditions, each representing different propagation environments. The blue trace ($\delta = 2.0$) corresponds to free-space conditions and converges the fastest, which is consistent with the minimal signal degradation. The orange trace ($\delta = 2.4$) reflects moderate path loss, which is typical of lower-altitude flights. The green trace ($\delta = 2.8$) represents urban environments near vertiports, where buildings and obstructions increase the multipath interference. These results demonstrate the ability of the system to converge across the varying wireless conditions encountered during different flight phases.

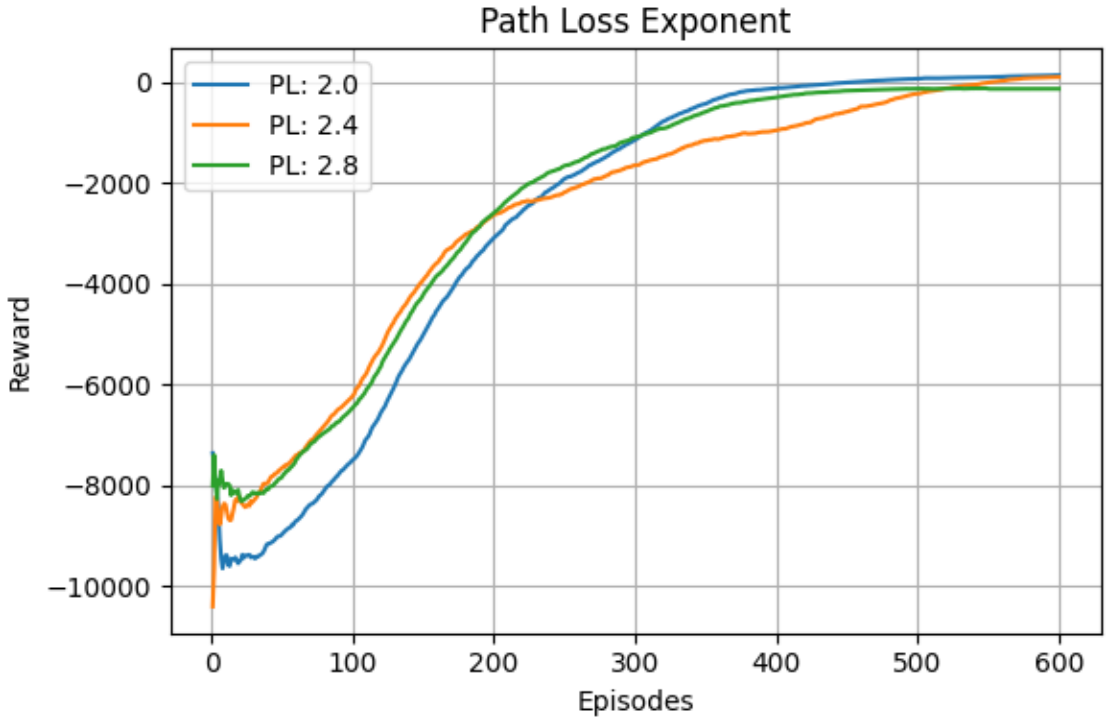


Figure 4.10: Impact of path loss exponent on MAPPO learning performance. The figure compares convergence for δ values of 2.0, 2.4, and 2.8 under SINR threshold 4.5 and minimum separation 300 units.

4.7 QMIX Performance Analysis

The QMIX algorithm was evaluated using the same use case scenarios as those in the MAPPO assessment. Training was conducted under identical MDP configurations, reward structures, agent setups, and mathematical models within the UAM airspace framework. The QMIX configuration parameters are listed in Table 4.5. The scenario involved six aircraft flying predefined trajectories, four communication stations, and six vertiports in the Cleveland airspace, which were identical to the MAPPO setup.

Table 4.5: Hyperparameter settings for the QMIX algorithm. The table lists values for episodes, batch size, learning rate, discount factor, and other parameters used in cooperative multi-agent training.

Model Parameters	**Value**
Max. episodes $\hat{N}_{\text{ep}}$	1000
Max. number of steps N_{step}	55
Batch Size $\hat{N}_b$	64
Soft update rate ρ	0.01
Learning rate α	0.0005
Discount factor γ	0.99
Simulation Time	2 s
Time slot duration ΔT	1 s

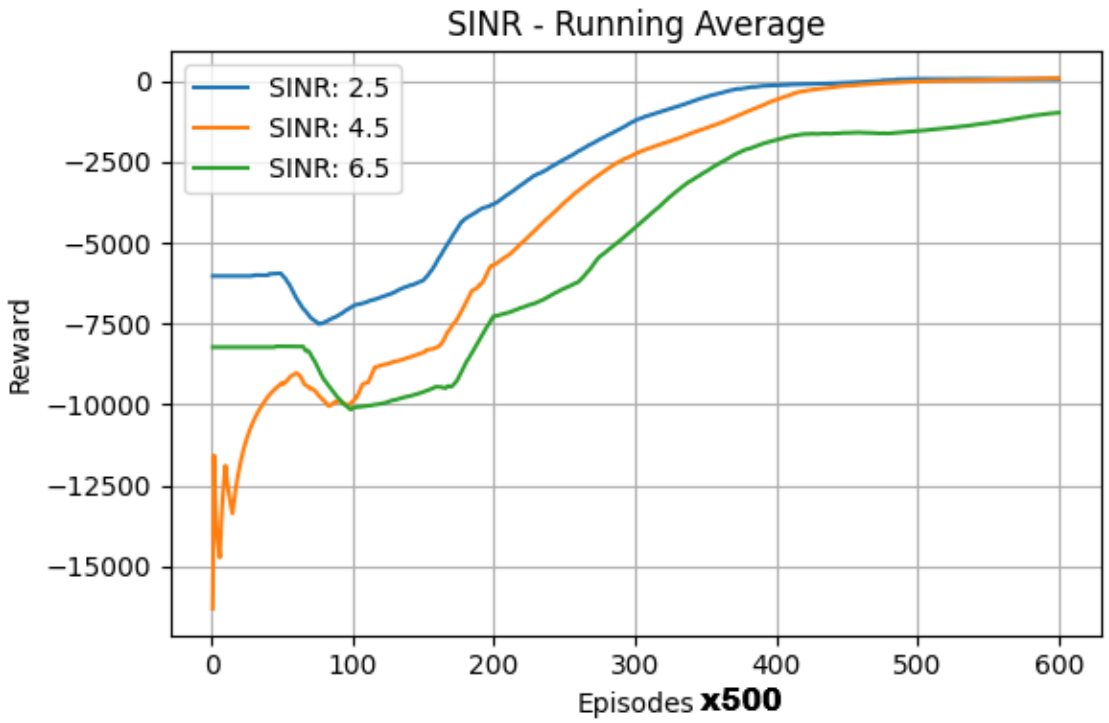


Figure 4.11: Convergence performance of QMIX under three SINR thresholds (2.5, 4.5, and 6.5). The figure shows how stricter SINR requirements increase training complexity and convergence time.

4.7.1 SINR Evaluation

We evaluated the performance of V-D3QN with QMIX in managing the communication channel assignment and reuse performance across different SINR constraint values γ_{qos} of 2.5, 4.5, and 6.5, where we let $\gamma_{\text{qos}} = \gamma^{u}_{\text{qos}} = \gamma^{d}_{\text{qos}}$. For this evaluation, the minimum aircraft separation distance was set at $d_{min} = 100$ units, with SINR violation penalty $= -50$, separation violation penalty $= -50$ and airborne waiting penalty $= -50$.

Fig. 4.11 illustrates the convergence performance of the three γ_{qos} values. Experimental results showed that QMIX required significantly more episodes to achieve

convergence compared to MAPPO. To maintain clarity and readability in the graphical analysis, we opted to record progress at intervals of 500 episodes rather than capturing data for every single episode. While the system continues to optimize parameters at each episode, the data collection for visualization purposes is downsampled by a factor of 500. Consequently, the x-axis in the QMIX convergence graphs is labeled as "Episodes $\times$ 500" to reflect this aggregation.

When $\gamma_{\text{qos}} = 2.5$, the system converged in fewer episodes than when compared to $\gamma_{\text{qos}} = 4.5$ and 6.5. The slowest convergence occurs at $\gamma_{\text{qos}} = 6.5$, owing to a higher SINR threshold that requires stronger signals, making interference avoidance more challenging. As expected, higher SINR values increase the complexity of the frequency deconfliction owing to the greater signal propagation distance. Nonetheless, all the configurations converged to a policy by episode 600 ($\times$ 500).

The QMIX algorithm effectively manages spectrum assignments while maintaining the SINR and separation constraints. As shown in Fig. 4.12, the system allocates frequencies for all aircraft under a stringent SINR threshold of 6.5 and a separation requirement of 100 units. Owing to the limited channel availability, each aircraft is assigned a single channel per time step. The system also reuses frequencies while avoiding interference; for example, at $t = 12$, frequency f_2 is assigned to n_1 and n_3, whereas f_3 is reused by n_2 and n_6. Fig. 4.13 shows how velocity control is used to support interference mitigation. At $t = 12$, aircraft n_1, n_2, and n_3 reduce their speeds to maintain safe and efficient communication.

Frequency Selection							Freq. Reuse			
t	n1	n2	n3	n4	n5	n6	f1	f2	f3	f4
t1	4	2	2	4	3	1	1	2	1	2
t2	3	2	3	4	1	1	2	1	2	1
t3	3	2	3	4	1	4	1	1	2	2
t4	3	2	3	4	1	4	1	1	2	2
t5	3	2	3	4	1	4	1	1	2	2
t6	3	2	3	4	1	4	1	1	2	2
t7	3	2	3	4	1	4	1	1	2	2
t8	3	2	3	4	1	4	1	1	2	2
t9	3	2	3	4	1	4	1	1	2	2
t10	3	2	3	4	1	4	1	1	2	2
t11	3	2	3	1	4	2	1	2	2	1
t12	3	2	3	1	4	2	1	2	2	1
t13	3	2	3	1	4	2	1	2	2	1
t14	3	2	4	1	4	2	1	2	1	2
t15	3	2	4	1		2	1	2	1	1
t16	3	1	2	1		2	2	2	1	0
t17	3		1	2		4	1	1	1	1
t18	3		1	4		4	1	0	1	2
t19	3						0	0	1	0
t20	3						0	0	1	0
t21	3						0	0	1	0
t22	3						0	0	1	0

Figure 4.12: Frequency allocation and reuse patterns for QMIX under SINR threshold 6.5 and minimum separation 100 units. The figure illustrates how channels are managed to maintain communication quality.

Velocity						
t	n1	n2	n3	n4	n5	n6
t1	25	50	50	50	50	50
t2	50	50	50	50	50	50
t3	50	50	50	25	50	50
t4	50	50	50	25	50	50
t5	50	50	25	25	50	50
t6	25	50	50	25	50	50
t7	25	50	25	25	50	25
t8	25	50	25	25	50	25
t9	25	50	25	25	50	25
t10	25	50	50	25	50	25
t11	25	50	50	50	50	50
t12	25	25	25	50	50	50
t13	25	25	25	50	50	50
t14	25	25	25	50	50	25
t15	25	50	25	50		25
t16	50	50	50	50		25
t17	50		50	50		50
t18	25		50	50		50
t19	25					
t20	50					
t21	50					
t22	50					

Figure 4.13: Velocity adjustment strategies used by QMIX to mitigate interference under SINR threshold 6.5 and minimum separation 100 units. The figure shows how speed control supports spectrum efficiency.

4.7.2 Path Loss Exponent Evaluation

As noted in the MAPPO evaluation, the path loss exponent is a key parameter in wireless communications because it governs how rapidly the signal power decays with distance. This directly influences the spectrum utilization and maximum achievable channel efficiency. In urban environments, channel quality is expected to vary depending on terrain, building density, aircraft altitude, and other factors. In our evaluations, we aimed to assess the system's ability to effectively manage a set of frequencies and reuse them without causing harmful interference in a given geographical area. Signal decay rates differ depending on the environmental geography, which imposes limitations on frequency reuse.

Simulation experiments were conducted using path loss values representative of both dense urban environments and aircraft cruise altitudes. A value of $\delta = 2.0$ reflects free-space conditions, while $\delta = 3.5$ represents an urban environment in which the aircraft is performing landing or departure procedures. The training converged at approximately 600 ($\times$ 500) episodes for all path loss values as indicated in Fig. 4.14. However, as expected, both the free-space case and the $\delta = 2.4$ case converge sooner than $\delta = 3.5$. The network requires more time to converge in a more complex urban environment.

4.8 MAPPO and QMIX Analysis

Both value-based and policy-based approaches were evaluated using the same UAM environment, objective function, constraints, and flight plan parameters. A

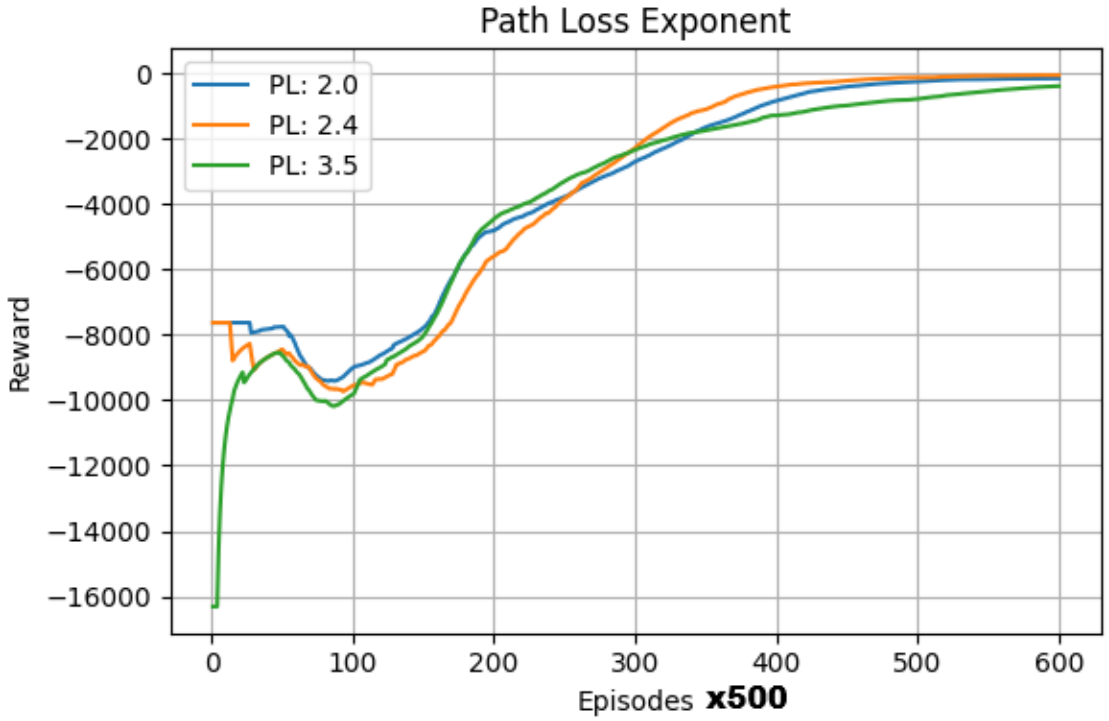


Figure 4.14: Effect of path loss exponent on QMIX convergence. The figure compares training performance for different propagation conditions under SINR threshold 4.5 and minimum separation 300 units.

comparative analysis of the simulation results was conducted to assess the performance and highlight the advantages of each technique.

Fig. 4.15 compares the simulation results of QMIX (dashed orange trace) and MAPPO (solid blue trace). The lower x-axis corresponds to MAPPO episodes, the upper x-axis to QMIX episodes, and the y-axis to the reward. Both algorithms were tested with an SINR target of $\gamma_{\text{qos}} = 4.0$, a separation constraint of $d_{min} = 100$, and penalty values of -50 for SINR violations, separation violations, and airborne waiting. A low separation threshold emphasizes the SINR resolution. MAPPO converged by episode 400, whereas QMIX required approximately 400,000 episodes, highlighting MAPPO's superior sample efficiency.

A second evaluation, as shown in Fig. 4.16, used a higher separation threshold of $d_{min} = 250$ and a reduced SINR target of $\gamma_{\text{qos}} = 3.0$ to shift the focus to separation. The penalty values remain unchanged. In this case, MAPPO converged by episode 300, whereas QMIX required approximately 300,000 episodes, again demonstrating faster convergence of MAPPO.

Both methods successfully learned optimal policies that maintained the SINR below the target and ensured safe aircraft separation. While both used speed control and frequency reuse to manage interference and separation, MAPPO consistently converged in significantly fewer episodes—approximately 1,000 times faster than QMIX.

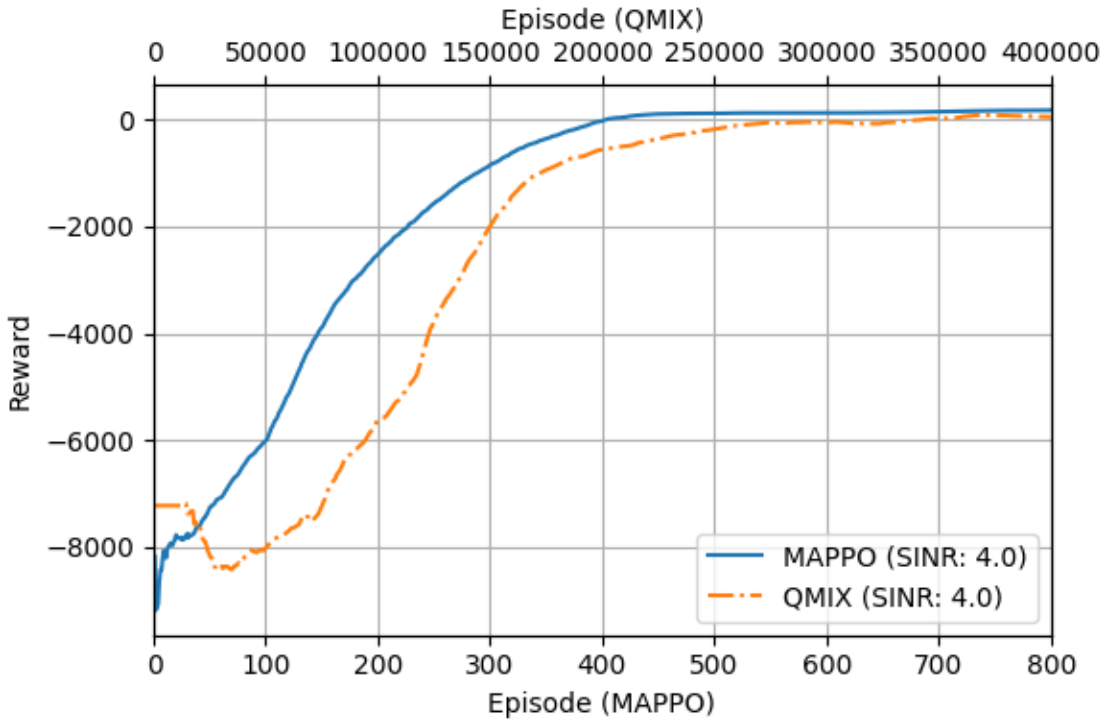


Figure 4.15: Comparison of SINR convergence between MAPPO and QMIX under SINR threshold 4.0. The figure highlights MAPPO's faster learning and superior sample efficiency.

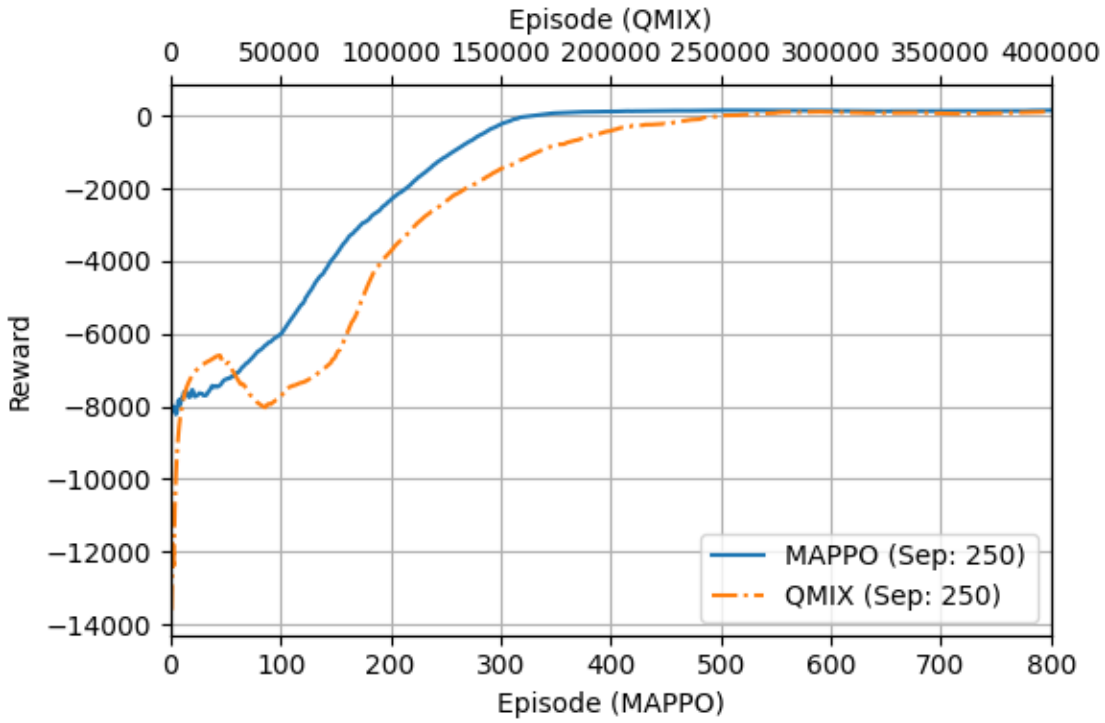


Figure 4.16: Comparison of MAPPO and QMIX performance under a separation constraint of 250 units. The figure shows how both algorithms manage interference and maintain safe distances, with MAPPO converging significantly faster.

4.9 Observations and Potential Follow-on Studies

This study explored the application of multi-agent deep reinforcement learning to the joint optimization of airspace and spectrum resources in UAM environments. Two MARL approaches—policy-based MAPPO and value-based QMIX—were evaluated under varying signal quality and propagation conditions.

Key observations include:

- Both MAPPO and QMIX successfully learned policies that maintained aircraft separation and met SINR constraints while minimizing flight time.
- MAPPO consistently demonstrated superior sample efficiency, converging to optimal policies in significantly fewer episodes than QMIX—often by orders of magnitude.
- The learning performance of both algorithms was sensitive to environmental parameters such as SINR thresholds and path loss exponents. Higher SINR requirements and urban-like propagation conditions (higher path loss) led to slower convergence.
- Both methods effectively leveraged frequency reuse and velocity control to manage interference and maintain safe operations, even under constrained channel availability.

Potential directions for follow-on studies:

- Scalability analysis: Investigate how these MARL methods perform in larger-scale UAM networks with more aircraft, vertiports, and dynamic traffic patterns.
- Real-time adaptability: Extend the framework to handle real-time changes in network topology, such as aircraft rerouting or emergency landings.
- Heterogeneous agent behavior: Explore scenarios where aircraft have different communication capabilities, mission priorities, or learning strategies.
- Integration with air traffic management (ATM): Study how MARL-based resource allocation can be integrated with existing ATM systems and regulatory frameworks.
- Robustness to uncertainty: Evaluate the resilience of learned policies under uncertain or adversarial conditions, such as jamming, signal degradation, or sensor noise.

These findings support the feasibility of using MARL for coordinated resource management in future UAM systems and highlight promising avenues for continued research and development.

CHAPTER V

SUMMARY AND CONCLUSIONS

5.1 Summary of Research

This dissertation presents a comprehensive investigation into the application of machine learning (ML) and artificial intelligence (AI) techniques for optimizing spectrum utilization in the context of Advanced Air Mobility (AAM), with a particular focus on Urban Air Mobility (UAM) scenarios. The increasing demand for aeronautical communications, driven by the emergence of new aerial vehicles and operational paradigms, necessitates a more intelligent and adaptive approach to spectrum management beyond traditional static allocation methods.

The research is structured around three core areas:

- **Prediction of Airspace Demand:** Methods for forecasting aircraft trajectories, airspace density, and communication demands were explored using both learning-based and rule-based approaches. These predictions serve as critical inputs for dynamic resource allocation.
- **Learning-Based Resource Allocation:** Two primary ML approaches were investigated:

  - *Graph Neural Networks (GNNs):* Used to solve the channel assignment problem through graph coloring, where aircraft and their potential interference relationships are modeled as nodes and edges in a graph.
  - *Multi-Agent Reinforcement Learning (MARL):* Specifically, MAPPO (policy-based) and QMIX (value-based) algorithms were applied to jointly optimize channel selection and aircraft velocity, aiming to minimize flight time while maintaining communication quality and safety constraints.

- **Modeling and Simulation Tool Development:** A modular simulation platform was developed to evaluate the proposed concepts in realistic airspace scenarios. The tool integrates prediction modules, resource allocation algorithms, and visualization capabilities, enabling detailed performance analysis.

5.2 Key Findings

- Graph Neural Networks demonstrated superior performance over greedy algorithms in high-density airspace scenarios, effectively reducing interference through intelligent channel assignment.
- Reinforcement Learning Approaches (MAPPO and QMIX) successfully learned optimal policies for joint spectrum and airspace resource management. MAPPO consistently outperformed QMIX in terms of convergence speed and sample efficiency.

- Prediction-driven simplification of the problem space (e.g., allocating channels only when communication is predicted) significantly reduced computational complexity and improved allocation efficiency.
- Simulation results confirmed that intelligent spectrum management can enable frequency reuse, reduce interference, and support the scalability of future AAM operations.

5.3 Conclusions

The research confirms that integrating AI and ML into aviation spectrum management offers a promising path toward meeting the communication demands of future airspace systems. By leveraging predictive analytics and adaptive learning algorithms, it is possible to dynamically allocate spectrum resources in a way that is both efficient and scalable.

The proposed framework supports:

- Enhanced spectrum utilization efficiency,
- Improved airspace capacity,
- Support for autonomous and remotely piloted operations, and
- A foundation for future regulatory and operational frameworks in AAM.

5.4 Future Work

Several avenues for future research are identified:

- Incorporation of time-dynamic graph models to better reflect evolving airspace conditions.
- Transfer learning techniques to improve generalization across different airspace configurations.
- Integration with structured airspace concepts, such as predefined corridors and waypoints.
- Expansion of the simulation environment to include more complex propagation models and real-time data feeds.
- Regulatory alignment and implementation strategies for transitioning to intelligent spectrum management in operational environments.

BIBLIOGRAPHY

[1] Rafael D. Apaza, Eric J. Knoblock, and Hongxiang Li. A New Spectrum Management Concept for Future NAS Communications. In *2020 AIAA/IEEE 39th Digital Avionics Systems Conference (DASC)*, pages 1–7, 2020.

[2] Brian P Hill, Dwight DeCarme, Matt Metcalfe, Christine Griffin, Sterling Wiggins, Chris Metts, Bill Bastedo, Michael D Patterson, and Nancy L Mendonca. UAM Vision Concept of Operations (ConOps) UAM Maturity Level (UML) 4. `https://ntrs.nasa.gov/citations/20205011091`.

[3] Department of Transportation. Upper Class E Traffic Management. `https://www.faa.gov/uas/advanced_operations/upper_class_etm`.

[4] He Li, Kaoru Ota, and Mianxiong Dong. Learning IoT in Edge: Deep Learning for the Internet of Things with Edge Computing. *IEEE Network*, 32(1):96–101, 2018.

[5] Shiqiang Wang, Tiffany Tuor, Theodoros Salonidis, Kin K. Leung, Christian Makaya, Ting He, and Kevin Chan. When Edge Meets Learning: Adaptive Control for Resource-Constrained Distributed Machine Learning. In *IEEE IN-*

FOCOM 2018 - IEEE Conference on Computer Communications, pages 63–71, 2018.

[6] Guangxu Zhu, Dongzhu Liu, Yuqing Du, Changsheng You, Jun Zhang, and Kaibin Huang. Toward an Intelligent Edge: Wireless Communication Meets Machine Learning. *IEEE Communications Magazine*, 58(1):19–25, 2020.

[7] Yong Wang, Margaret Martonosi, and Li-Shiuan Peh. Predicting link quality using supervised learning in wireless sensor networks. *SIGMOBILE Mob. Comput. Commun. Rev.*, 11(3):71–83, 2007.

[8] Eduardo Feo Flushing, Jawad Nagi, and Gianni A. Di Caro. A Mobility-assisted Protocol for Supervised Learning of Link Quality Estimates in Wireless Networks. In *2012 International Conference on Computing, Networking and Communications (ICNC)*, pages 137–143, 2012.

[9] Gianni A. Di Caro, Michal Kudelski, Eduardo Feo Flushing, Jawad Nagi, Imran Ahmed, and Luca M. Gambardella. Online supervised incremental learning of link quality estimates in wireless networks. In *2013 12th Annual Mediterranean Ad Hoc Networking Workshop (MED-HOC-NET)*, pages 133–140, 2013.

[10] Ya Tu, Yun Lin, Jin Wang, and Jeong-Uk Kim. Semi-Supervised Learning with Generative Adversarial Networks on Digital Signal Modulation Classification. *Computers, Materials and Continua*, 55(2), 2018.

[11] Robert C. Daniels, Constantine M. Caramanis, and Robert W. Heath. Adaptation in Convolutionally Coded MIMO-OFDM Wireless Systems Through Supervised Learning and SNR Ordering. *IEEE Transactions on Vehicular Technology*, 59(1):114–126, 2010.

[12] Yo-Seb Jeon, Song-Nam Hong, and Namyoon Lee. Supervised-Learning-Aided Communication Framework for MIMO Systems With Low-Resolution ADCs. *IEEE Transactions on Vehicular Technology*, 67(8):7299–7313, 2018.

[13] Pedro Forero, Alfonso Cano, and G.B. Giannakis. Distributed Clustering Using Wireless Sensor Networks. *IEEE Journal of Selected Topics in Signal Processing*, 5:707–724, September 2011.

[14] Yuchao Chang, Xiaobing Yuan, Baoqing Li, Dusit Niyato, and Naofal Al-Dhahir. A Joint Unsupervised Learning and Genetic Algorithm Approach for Topology Control in Energy-Efficient Ultra-Dense Wireless Sensor Networks. *IEEE Communications Letters*, 22(11):2370–2373, 2018.

[15] Hao Huang, Wenchao Xia, Jian Xiong, Jie Yang, Gan Zheng, and Xiaomei Zhu. Unsupervised Learning-Based Fast Beamforming Design for Downlink MIMO. *IEEE Access*, 7:7599–7605, 2019.

[16] Caleb Bowyer, David Greene, Tyler Ward, Marco Menendez, John Shea, and Tan Wong. Reinforcement Learning for Mixed Cooperative/Competitive Dynamic Spectrum Access. In *2019 IEEE International Symposium on Dynamic Spectrum*

Access Networks (DySPAN), pages 1–6, 2019.

[17] Abiodun Gbenga-Ilori, Agbotiname Lucky Imoize, Kinzah Noor, and Paul Oluwadara Adebolu-Ololade. Artificial intelligence empowering dynamic spectrum access in advanced wireless communications: A comprehensive overview. *AI*, 6(6):126, 2025.

[18] Chaymaa Lakouismi, Hamza Ouamna, Zhour Madini, Younes Zouine, and Anass Kharbouche. A review on ai-driven dynamic spectrum access for next-generation wireless networks. In *International Conference on Advanced Systems Engineering and Technologies (ICASET)*, volume 1555 of *Lecture Notes in Networks and Systems*, pages 46–55. Springer, 2025.

[19] Akshay Upadhye, Purushothaman Saravanan, Shreeram Suresh Chandra, and Sanjeev Gurugopinath. A survey on machine learning algorithms for applications in cognitive radio networks. *arXiv preprint*, 2021.

[20] Youness Arjoune and Naima Kaabouch. A comprehensive survey on spectrum sensing in cognitive radio networks: Recent advances, new challenges, and future research directions. *Sensors*, 19(1):126, 2019.

[21] Bushra Sabir, Shuiqiao Yang, David Nguyen, Nan Wu, Alsharif Abuadbba, Hajime Suzuki, Shangqi Lai, Wei Ni, Ding Ming, and Surya Nepal. Systematic literature review of ai-enabled spectrum management in 6g and future networks. *arXiv preprint*, 2024.

[22] Asrar Ahmad Ansari and Prabhdeep Singh. Ai-based dynamic spectrum allocation for hybrid satellite-5g networks. *Iconic Research and Engineering Journals*, 8(10):472–485, 2025.

[23] G. M. Karthik, V. Pushpa, R. Rekha, C. Nandhakumar, S. Logesswari, and N. Naga Saranya. Crlsa: Cognitive reinforcement learning spectrum allocation in addressing spectrum scarcity in 5g wireless communication. *WSEAS Transactions on Information Science and Applications*, 22, 2025.

[24] Yang Lu, Yuhang Li, Ruichen Zhang, Wei Chen, Bo Ai, and Dusit Niyato. Graph neural networks for wireless networks: Graph representation, architecture and evaluation. *arXiv preprint*, 2024.

[25] The MathWorks Inc. Wireless resource allocation using graph neural network, 2024. Available at: `https://www.mathworks.com/help/signal/ug/wireless-resource-allocation-using-graph-neural-network.html` (Accessed: 2025-11-17).

[26] Yanzhi Li. Graph neural networks for dynamic spectrum access, 2025. Available at: `https://www.spectrumx.org/exhibitors/optimizing-wireless-interference-management-via-graph-neural-networks/` (Accessed: 2025-11-17).

[27] Maher Marwani and Georges Kaddoum. Graph neural networks approach for joint wireless power control and spectrum allocation. *IEEE Transactions on Machine Learning in Communications and Networking*, 2024. Code available at:

https://github.com/mahermarwani/Graph-Neural-Networks-Approach-for-Joint-Wireless-Power-Control-and-Spectrum-Allocation.

[28] He Jiang. *Graph Neural Network Based Learning for Dynamic Spectrum Access*. PhD thesis, University of Rhode Island, 2021.

[29] Yifei Shen, Jun Zhang, S. H. Song, and Khaled B. Letaief. Graph neural networks for wireless communications: From theory to practice. *arXiv preprint*, 2022.

[30] José Suárez-Varela, Pere Almasan, Marc Ferriol-Galmés, and Albert Cabellos-Aparicio. Graph neural networks for communication networks: Context, use cases and opportunities. *IEEE Network*, 36(1):212–219, 2022.

[31] Eric J. Knoblock, Rafael D. Apaza, Michael R. Gasper, Hongxiang Li, Ruixuan Han, Zhe Wang, Nora Schimpf, Nathaniel Rose, and Neil Adams. Intelligent Spectrum Management for Future Aeronautical Communications. *IEEE Aerospace and Electronic Systems Magazine*, 38(5):46–59, 2023.

[32] Changhee Lee, Sung-Min Oh, and Ae-Soon Park. Interference avoidance resource allocation for D2D communication based on graph-coloring. In *2014 International Conference on Information and Communication Technology Convergence (ICTC)*, pages 895–896. IEEE, 2014.

[33] Dimitris Tsolkas, Eirini Liotou, Nikos Passas, and Lazaros Merakos. A graph-coloring secondary resource allocation for D2D communications in LTE networks.

In *2012 IEEE 17th international workshop on computer aided modeling and design of communication links and networks (CAMAD)*, pages 56–60. IEEE, 2012.

[34] Xuejia Cai, Jun Zheng, and Yuan Zhang. A graph-coloring based resource allocation algorithm for D2D communication in cellular networks. In *2015 IEEE International Conference on Communications (ICC)*, pages 5429–5434. IEEE, 2015.

[35] Suzan Basloom, Amani Nazar, Ghadah Aldabbagh, Manal Abdullah, and Nikos Dimitriou. Resource allocation using graph coloring for dense cellular networks. In *2016 International Conference on Computing, Networking and Communications (ICNC)*, pages 1–5. IEEE, 2016.

[36] Emmanouil Pateromichelakis and Konstantinos Samdanis. A graph coloring based inter-slice resource management for 5G dynamic TDD RANs. In *2018 IEEE International Conference on Communications (ICC)*, pages 1–6. IEEE, 2018.

[37] Martin J A Schuetz, J Kyle Brubaker, and Helmut G Katzgraber. Combinatorial Optimization with Physics-Inspired Graph Neural Networks. *arXiv preprint arXiv:2107.01188*, 2021.

[38] Martin JA Schuetz, J Kyle Brubaker, Zhihuai Zhu, and Helmut G Katzgraber. Graph Coloring with Physics-Inspired Graph Neural Networks. *arXiv preprint*

arXiv:2202.01606, 2022.

[39] Nora Schimpf, Zhe Wang, Summer Li, Eric J. Knoblock, Hongxiang Li, and Rafael D. Apaza. A Generalized Approach to Aircraft Trajectory Prediction via Supervised Deep Learning. *IEEE Access*, 11:116183–116195, 2023.

[40] Nathan Schimpf, Hongxiang Li, Eric Knoblock, and Rafael Apaza. Communication Demand in the National Airspace – A Federated Learning Approach. In *2022 Integrated Communication, Navigation and Surveillance Conference (ICNS)*, pages 1–11, 2022.

[41] Eric J. Knoblock, Rafael D. Apaza, Michael L. Gasper, Hongxiang Li, Nora Schimpf, Nathaniel Rose, and Neil Adams. Dynamic Channel Assignments for Efficient Use of Aviation Spectrum Allocations. In *2022 IEEE/AIAA 41st Digital Avionics Systems Conference (DASC)*, pages 1–7, 2022.

[42] Department of Transportation and Federal Aviation Administration. *Spectrum Management Regulations and Procedures Manual*. Department of Transportation and Federal Aviation Administration, 11 2005.

[43] Theodore Rappaport. *Wireless Communications: Principles and Practice*. Prentice Hall PTR, USA, 2nd edition, 2001.

[44] Zeenat Afroze and David Matolak. AAM Air Ground Channel Characterization in the C Band: Initial Laboratory and Drive Test Results. In *2024 IEEE/AIAA*

Digital Avionics Systems Conference (DASC), pages 1–7, September 2024.

[45] R.M.R. Lewis. *A Guide to Graph Colouring - Algorithms and Applications*. Springer International Publishing, 2016.

[46] Zsolt Tabi, Kareem H. El-Safty, Zsofia Kallus, Peter Haga, Tamas Kozsik, Adam Glos, and Zoltan Zimboras. Quantum Optimization for the Graph Coloring Problem with Space-Efficient Embedding. In *2020 IEEE International Conference on Quantum Computing and Engineering (QCE)*, pages 56–62. IEEE, October 2020.

[47] Wei Li, Ruxuan Li, Yuzhe Ma, Siu On Chan, David Pan, and Bei Yu. Rethinking Graph Neural Networks for the Graph Coloring Problem, 2022.

[48] Xiangyu Wang, Xueming Yan, and Yaochu Jin. A Graph Neural Network with Negative Message Passing for Graph Coloring, 2023.

[49] Zi Chen, Keke Liang, Long Yuan, Wenjie Zhang, and Zhengyi Yang. Recent Advances in Efficient Dynamic Graph Processing. *Applied Sciences*, 15(11), 2025.

[50] Menno Theunis and Marcel Roeloffzen. Experimental Analysis of Algorithms for the Dynamic Graph Coloring Problem. *Journal of Graph Algorithms and Applications*, 28(1):313–349, Aug. 2024.

[51] Maxime Flin and Magnús M. Halldórsson. Faster Dynamic $(\Delta + 1)$-Coloring Against Adaptive Adversaries, 2025.

[52] Aldo Pareja, Giacomo Domeniconi, Jie Chen, Tengfei Ma, Toyotaro Suzumura, Hiroki Kanezashi, Tim Kaler, Tao B. Schardl, and Charles E. Leiserson. EvolveGCN: Evolving Graph Convolutional Networks for Dynamic Graphs, 2019.

[53] Franco Manessi, Alessandro Rozza, and Mario Manzo. Dynamic graph convolutional networks. *Pattern Recognition*, 97:107000, 2020.

[54] Shay Solomon and Nicole Wein. Improved Dynamic Graph Coloring. *ACM Transactions on Algorithms (TALG)*, 16(3):1–24, 2020.

[55] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S. Yu. A Comprehensive Survey on Graph Neural Networks. *IEEE Transactions on Neural Networks and Learning Systems*, 32(1):4–24, 2021.

[56] Matthieu Cordonnier, Nicolas Keriven, Nicolas Tremblay, and Samuel Vaiter. Convergence of Message Passing Graph Neural Networks with Generic Aggregation On Large Random Graphs, 2025.

[57] Georgios Gkarmpounis, Christos Vranis, Nicholas Vretos, and Petros Daras. Survey on Graph Neural Networks. *IEEE Access*, 12:128816–128832, 2024.

[58] Jian Du, Shanghang Zhang, Guanhang Wu, José MF Moura, and Soummya Kar. Topology Adaptive Graph Convolutional Networks. *arXiv preprint arXiv:1710.10370*, 2017.

[59] William L. Hamilton, Rex Ying, and Jure Leskovec. Inductive Representation Learning on Large Graphs, 2018.

[60] Nishai Kooverjee, Steven James, and Terence van Zyl. Investigating Transfer Learning in Graph Neural Networks, 2022.

[61] David Buterez, Jon Paul Janet, Steven Kiddle, Dino Oglic, and Pietro Lio. Transfer learning with graph neural networks for improved molecular property prediction in the multi-fidelity setting. *Nature Communications*, 2024.

[62] Department of Transportation. Urban Air Mobility Concept of Operations v2.0. `https://www.faa.gov/air-taxis/uam_blueprint`.

[63] Kaiqing Zhang, Zhuoran Yang, and Tamer Başar. Multi-Agent Reinforcement Learning: A Selective Overview of Theories and Algorithms, 2021.

[64] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. A Bradford Book, Cambridge, MA, USA, 2018.

[65] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal Policy Optimization Algorithms, 2017.

[66] Chao Yu, Akash Velu, Eugene Vinitsky, Jiaxuan Gao, Yu Wang, Alexandre Bayen, and Yi Wu. The Surprising Effectiveness of PPO in Cooperative, Multi-Agent Games, 2022.

[67] Tabish Rashid, Mikayel Samvelyan, Christian Schroeder de Witt, Gregory Farquhar, Jakob Foerster, and Shimon Whiteson. QMIX: Monotonic Value Function Factorisation for Deep Multi-Agent Reinforcement Learning, 2018.

[68] Andrea Goldsmith. *Wireless Communications*. Cambridge University Press, Cambridge, 2005.

[69] Frederick Wieland, Shaymaa Khater, Juan Rebollo, David Matolak, and Zeenat Afroze. Predicting AAM Path Loss through Neural Networks and Statistical Modeling. In *2024 Integrated Communications, Navigation and Surveillance Conference (ICNS)*, pages 1–6, 2024.

[70] Frans Oliehoek and Christopher Amato. A Concise Introduction to Decentralized POMDPs. January 2016.

[71] John Schulman, Sergey Levine, Philipp Moritz, Michael I. Jordan, and Pieter Abbeel. Trust Region Policy Optimization, 2017.

[72] Ying Huang, GuoLiang Wei, and YongXiong Wang. V-D D3QN: the Variant of Double Deep Q-Learning Network with Dueling Architecture. In *2018 37th Chinese Control Conference (CCC)*, pages 9130–9135, 2018.

[73] Gul Afroz, Ifra Khan, and Irfan Majid. Modeling and Simulation of Interference Between Aircraft Radar Altimeter and 5G-C Band Transmissions. In *Proceedings of the First International Conference on Aeronautical Sciences, Engineering and Technology (ICASET 2023)*. Springer, 2023.

[74] NASA. NASA WorldWind: Open Source Visualization Technology for Earth Observation. Technical report, NASA Technical Reports Server (NTRS), 2019. Accessed: 2025-11-13.

[75] Eric J. Knoblock, Rafael D. Apaza, Hongxiang Li, Zhe Wang, Ruixuan Han, Nathan Schimpf, and Nathaniel P. Rose. Investigation and Evaluation of Advanced Spectrum Management Concepts for Aeronautical Communications. In *2021 Integrated Communications Navigation and Surveillance Conference (ICNS)*, pages 1–12, 2021.

[76] SWIM Flight Data Publication Service (SFDPS), August 2021.

APPENDICES

APPENDIX A

MODELING AND SIMULATION TOOL DEVELOPMENT

To implement the intelligent spectrum management concept (i.e., integrated demand prediction and learning-based resource allocation) and evaluate its performance, a modeling and simulation platform is developed. This platform consists of computational tools and environment visualization capabilities. A key feature is the integration, development, and evaluation of learning-based models within a simulated airspace operational environment.

Additional design objectives include the development of a live, virtual, and constructive operational mode, enabling incremental evaluation of concepts as they mature. A modular architecture supports a flexible and scalable approach for testing various communication systems, airspace environments, and neural network models. Initial development efforts focus on communication analysis and visualization of VHF ground-to-air links used in ATC within the NAS, serving as a baseline for evaluating novel spectrum management strategies.

Toolset Architecture: The modeling and simulation tool employs a modular approach to deliver incremental capabilities. This design also facilitates integration with external software and interfaces for modeling diverse scenarios, as illustrated in Fig. A.1.

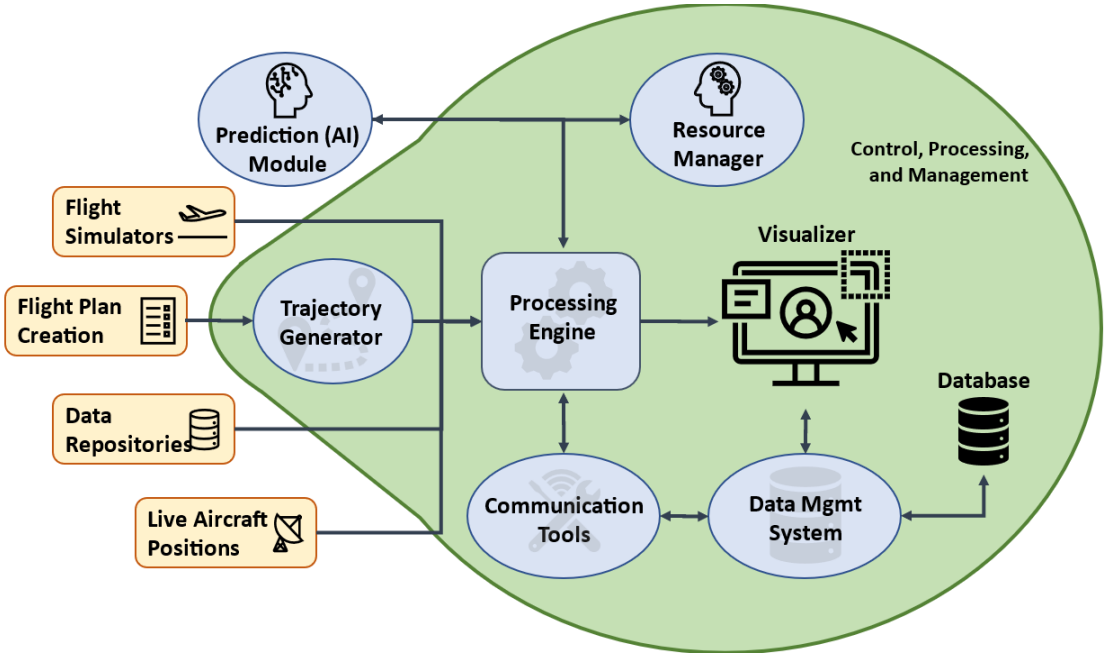


Figure A.1: Architecture diagram of the modeling and simulation platform. The figure shows modular components and their connections to external interfaces, illustrating how the system integrates prediction, resource management, and visualization for spectrum analysis.

Within the processing engine, simulation objects are updated based on a timer using predefined timesteps or real-time data streams. Object locations and properties are updated at each timestep. During each iteration, all tasks required to update and analyze active objects are added to a thread pool and executed concurrently. Once all threads complete, the processing engine redraws the visualizer graphics and waits for the next iteration.

Each timestep involves data processing using various code packages and communication tools. Geospatial calculations determine aircraft sector locations and path lengths between transceivers. Communication tools handle link budget analysis, interference evaluation, and QoS metrics.

Aircraft flight data preloaded into a scenario undergoes preprocessing before visualization. This includes interpolating aircraft positions, determining sector and heading, performing link budget calculations, and evaluating interference. Preprocessed data is rendered by reading time-stamped rows, improving simulation performance.

Live data feeds are processed similarly but in real time. For example, an interface to the X-Plane flight simulator allows the tool to receive live aircraft position updates, which are rendered after airspace and communication calculations are completed.

Visualization: The simulation visualization dynamically displays all objects in a spectrum management scenario. It graphically represents airspace, aircraft, ground stations, and weather data. Rendering is powered by the NASA WorldWind SDK [74], an open-source Java-based virtual globe API. The simulation control interface, built on WorldWind, provides users with an intuitive environment to manipulate and observe the simulation. Users can edit transceiver parameters, import flight and weather data, and configure communication link analysis. A screenshot is shown in Fig. A.2.

Data Management: Simulation parameters for airspace and ground stations are stored in a SQLite database. This includes sectorization, ground station locations, airports, waypoints, and NAVAIDs. Tables define relationships between centers, sectors, and facilities (e.g., ground stations in a sector, neighboring boundaries). NAS center and sector tables include identifiers, names, altitude ranges, bor-

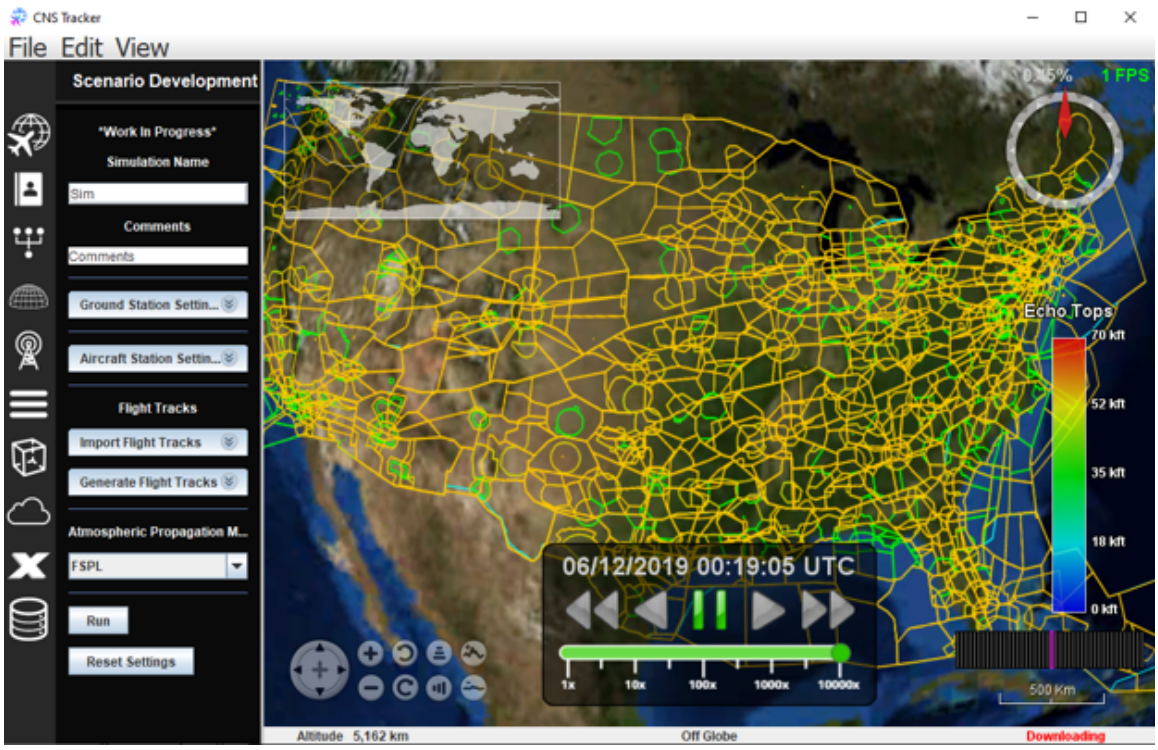


Figure A.2: Screenshot of the modeling and simulation tool interface built on NASA WorldWind. The interface displays airspace sectors, aircraft positions, and communication links, enabling interactive configuration and visualization of spectrum management scenarios.

ders, and centroids. ATC ground station tables include location names, sector IDs, frequencies, coverage radii, coordinates, power levels, and associated airspace [75]. A flight plan table is under development to store FAA-preferred routes for flight track generation. Scenario definitions and output data are also stored for analysis.

Communication Tools: As the engine updates object states, communication tools perform link budget calculations. For a given air-ground link, the controller identifies interfering links and computes the signal-to-interference ratio (SIR) as a QoS metric. The analysis compares the intended link to co-channel and adjacent-channel interference. Assumptions are based on FAA guidelines [42], including channel spac-

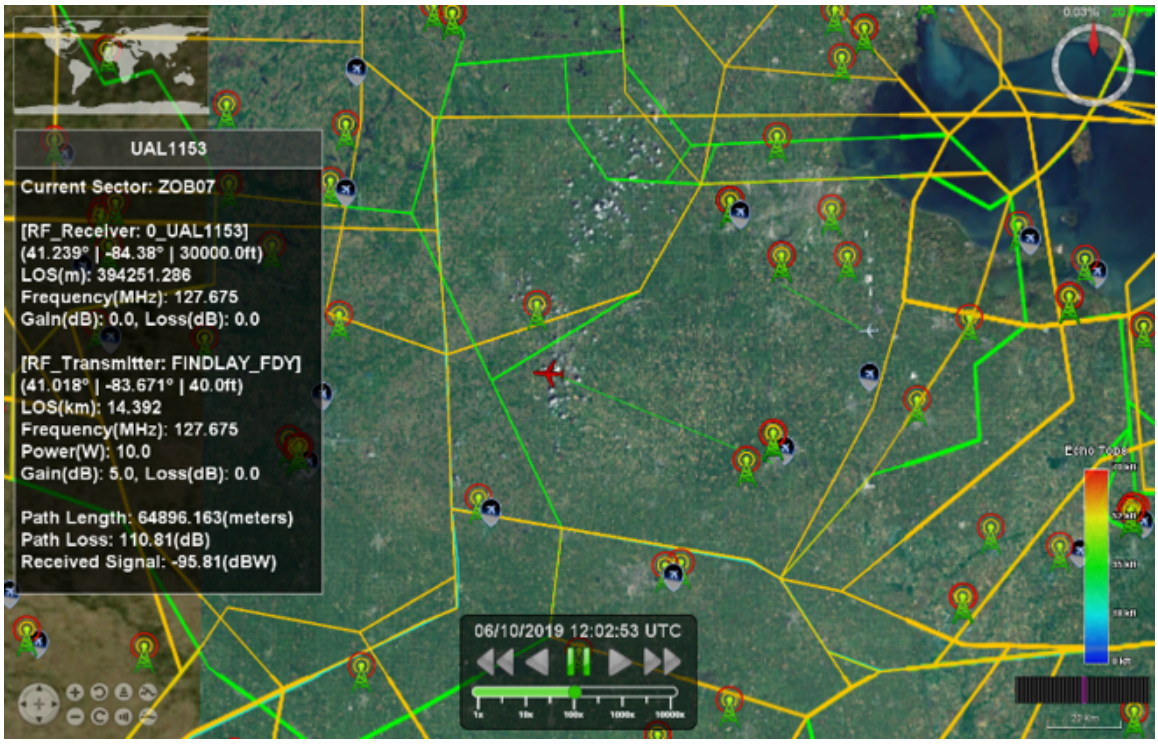


Figure A.3: Example visualization of airspace with air-to-ground link analysis results. The figure highlights aircraft positions, ground stations, and calculated signal-to-interference ratios for evaluating communication performance.

ing, emissions, line-of-sight limits, and power levels. The default propagation model is free-space path loss, with options to integrate more complex models. An example output is shown in Fig. A.3. By default, the tool assumes isotropic antennas, but users can import directional antenna patterns. For example, a VHF dipole pattern is shown in Fig. A.4. The software calculates geometry between transceivers to determine directional gain.

Resource Management: The Resource Manager module uses current and predicted states to allocate spectrum resources while maintaining minimum QoS. In the ATC case, frequencies are allocated to airspace sectors. The Resource Manager

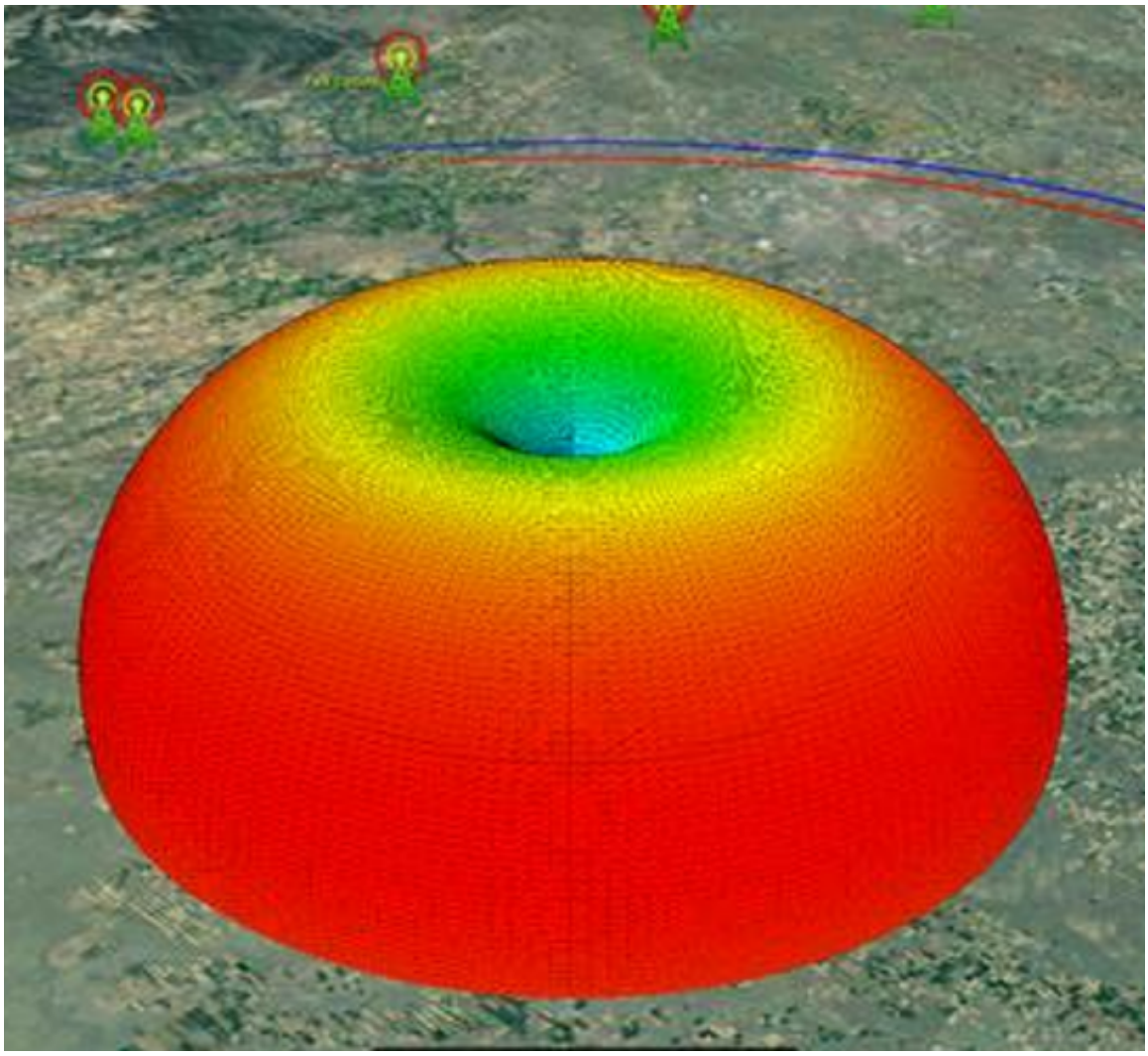


Figure A.4: Radiation pattern of a dipole antenna used at a ground station. The figure illustrates directional gain characteristics that influence link performance in spectrum management simulations.

integrates data from the processing engine and prediction module to optimize spectrum distribution for the scenario under study.

External Interfaces: Access to air traffic data is essential for assessing spectrum needs. The tool supports importing data from multiple sources. Historical data is available via NASA's Sherlock Data Warehouse, which includes processed SWIM

data such as track points and flight events. Simulated traffic from external flight simulators enables custom real-time scenarios. Custom traffic can also be generated from FAA-preferred or user-defined flight plans. Real-time assessments are supported using live data from FAA's SWIM Flight Data Publication Service [76] and ADS-B feeds. This enables simulation of both nominal and off-nominal operations.

Prediction: An interface for trajectory and airspace demand prediction was developed. The module receives current aircraft and weather data, returns predictions, and informs the Resource Manager for improved spectrum allocation.

APPENDIX B

ACRONYM LIST

Acronym	**Definition**
AAM	Advanced Air Mobility
ADS-B	Automatic Dependent Surveillance-Broadcast
AI	Artificial Intelligence
API	Application Programming Interface
ATC	Air Traffic Control
CNN	Convolutional Neural Network
CRN	Cognitive Radio Network
DARPA	Defense Advanced Research Projects Agency
DSA	Dynamic Spectrum Access
FAA	Federal Aviation Administration
GCN	Graph Convolutional Network
GNN	Graph Neural Network
MAC	Medium Access Control
MAPPO	Multi-Agent Proximal Policy Optimization
MARL	Multi-Agent Reinforcement Learning

Acronym	Definition
MIMO	Multiple Input Multiple Output
ML	Machine Learning
MSL	Mean Sea Level
NAS	National Airspace System
NASA	National Aeronautics and Space Administration
NAVAID	Navigation Aids
NTIA	National Telecommunications and Information Administration
PHY	Physical Layer
PPO	Proximal Policy Optimization
QoS	Quality of Service
ReLU	Rectified Linear Unit
RF	Radio Frequency
RL	Reinforcement Learning
RLF	Recursive Largest First
SDK	Software Development Kit
SINR	Signal-to-Interference-plus-Noise Ratio
SWIM	System Wide Information System
TFR	Temporary Flight Restriction
TRPO	Trust Region Policy Optimization
UAM	Urban Air Mobility
UHF	Ultra High Frequency

Acronym	Definition
VHF	Very High Frequency
V-D3QN	Variant of Double Dueling Deep Q-Network