Towards Intelligent UAV Swarms through Active Inference-Driven World Modeling

Kaleem Arshid

Department of Electrical, Electronic, Telecommunications Engineering and Naval Architecture (DITEN)
University of Genoa
and
Department of Systems Engineering and Automation (DISA)
University Carlos III of Madrid

This dissertation is submitted for the degree of
Doctor of Philosophy in
Electrical Engineering, Electronics and Automation

Joint Doctorate in Interactive and
Cognitive Environments - Cycle 38

March 2026

Towards Intelligent UAV Swarms through Active Inference-Driven World Modeling

Kaleem Arshid

Joint Doctorate in Interactive and Cognitive Environments
JD-ICE

XXXVIII cycle

Acknowledgements

This PhD Thesis has been developed in the framework of, and according to, the rules of the Joint Doctorate in Interactive and Cognitive Environments JD-ICE with the cooperation of the following Universities:

Università degli Studi di Genova (UNIGE)

DITEN - Dept. of Electrical, Electronic, Telecommunications Engineering and Naval Architecture

ISIP40 - Information and Signal Processing for Cognitive Telecommunications

Primary Supervisor: Prof. Carlo Regazzoni
Co-Supervisor: Prof. Ali Krayani
Co-Supervisor: Prof. Lucio Marcenaro

UNIVERSITÀ DEGLI STUDI DI GENOVA

Carlos III University of Madrid (UC3M)

DSEA - Department of Systems Engineering and Automation

LSI - Intelligent Systems Lab

Supervisor: Prof. David Martin Gomez

Universidad Carlos III de Madrid

Finanziato dall'Unione europea NextGenerationEU · Ministero dell'Università e della Ricerca · Italiadomani · Università di Genova

Borsa di dottorato cofinanziata con risorse dell'Unione europea-NextGeneration EU Piano Nazionale di Ripresa e Resilienza la Missione 4, componente 2 ("Dalla Ricerca all'Impresa")

I would like to dedicate this thesis to my loving family, especially my daughter, Umaima, and my son, Muhammad for their endless love, patience, and inspiration that gave me strength throughout this journey.. . .

Acknowledgements

First and foremost, I am extremely grateful to Allah Almighty, who, through His infinite mercy and grace, guided me, gave me the courage, perseverance, and blessings, and enabled me to complete my PhD in Joint Doctorate in Interactive and Cognitive Environments (JD-ICE). I am also deeply thankful to our Holy Prophet Muhammad (peace and blessings of Allah be upon him), who has been a constant source of guidance, light, and inspiration throughout my life.

I express my heartfelt gratitude to my supervisors, Professor Carlo Regazzoni and Professor Ali Krayani, for their invaluable guidance, continuous feedback, and unwavering support throughout these years of research. Having supervisors who provide sincere encouragement and assistance at every stage is truly a blessing.

I am also deeply grateful to Professor David Martín Gómez, whose patience, understanding, and guidance supported me at every stage, from theoretical research to practical experimentation. Likewise, I wish to extend my sincere thanks to Professor Lucio Marcinaro and Professor Pamela Zontone, whose availability and mentorship have always been a source of strength and motivation.

I would also like to thank all the esteemed members of the dissertation committee, who carefully reviewed my work and helped me improve my research with their valuable insights and constructive feedback.

I am thankful to all my colleagues at UniGe and UC3M for providing a pleasant and supportive research environment. Their friendship, collaboration, and encouragement made my academic journey both productive and memorable.

I am also deeply thankful to my friends in Italy and Spain, whose companionship, encouragement, and kindness provided me with wonderful moments of support and joy throughout my PhD journey.

Finally, I owe my deepest gratitude to my parents, wife, sisters, brothers, and relatives in Europe, whose love, prayers, and constant encouragement have been my greatest strength. Their sacrifices and support are the true foundation of my success. I consider myself truly fortunate to have been blessed with such love and unwavering support.

Published and Submitted Content

Below is a list of publications resulting from this PhD research:

0.1 Published Content

- **Kaleem Arshid**, Ali Krayani, Lucio Marcenaro, David Martin Gomez, and Carlo Regazzoni. "Toward Autonomous UAV Swarm Navigation: A Review of Trajectory Design Paradigms." ***Sensors***, Sep. 2025.

  doi:10.3390/s25185877.

  © 2025 MDPI.

  (This journal paper is wholly included in the thesis, in Chapter 2. The material from this source included in this thesis is not singled out with typographic means and references.)

- **Kaleem Arshid**, Ali Krayani, Lucio Marcenaro, David Martin Gomez, and Carlo Regazzoni. "UAV Swarm Trajectory Design for Wireless Networks Using Genetic Algorithm-Driven Repulsion Forces." ***IEEE Access***, Sep. 2025.

  doi:10.1109/ACCESS.2025.3606121.

  © 2025 IEEE.

  (This journal paper is wholly included in the thesis, in Chapter 3. The material from this source included in this thesis is not singled out with typographic means and references.)

- **Kaleem Arshid**, Ali Krayani, Lucio Marcenaro, David Martin Gomez, and Carlo Regazzoni. "Active Inference-Driven World Modeling for Adaptive UAV Swarm Trajectory Design." ***ICASSP 2026***.

  © 2025 IEEE.

(This conference paper is wholly included in the thesis, in Chapter 4. The material from this source included in this thesis is not singled out with typographic means and references.)

Other Research Merits

- Ahmad, Fawad, Muhammad Yasir Masood Mirza, Iftikhar Hussain, and **Kaleem Arshid**. A Multi-Ray Channel Modelling Approach to Enhance UAV Communications in Networked Airspace. *Inventions (MDPI)*, June 2025.

  doi:10.3390/inventions10040051. © 2025 MDPI.

- Noonari, Nooruddin, **Kaleem Arshid**, Shamaila Fardous, Javed Karim, and Sathishkumar Duraisamy. Implementation of Maximum Weighted Independent Set through Brute Force and Greedy Heuristics Approaches. *2024 International Conference on Intelligent Algorithms for Computational Intelligence Systems (IACIS)*, IEEE, 2024. doi:10.1109/IACIS61494.2024.10721981. © 2024 IEEE.

- Maitlo, Nizamuddin, Nooruddin Noonari, **Kaleem Arshid**, Naveed Ahmed, and Sathishkumar Duraisamy. AINS: Affordable Indoor Navigation Solution via Line Color Identification Using Mono-Camera for Autonomous Vehicles. *2024 IEEE 9th International Conference for Convergence in Technology (I2CT)*, IEEE, 2024. doi:10.48550/arXiv.2402.04750. © 2024 IEEE.

- Anum Malik, **Kaleem Arshid**, Nooruddin Noonari, and Rizwan Munir. Artificial Intelligence-Driven Cybersecurity Framework Using Machine Learning for Advanced Threat Detection and Prevention. *Scholars Journal of Engineering and Technology (SAS Publisher)*, June 2025. doi:10.36347/sjet.2025.v13i06.005.

- Hussain, Afifa, Nooruddin Noonari, and **Kaleem Arshid**. “TruthSentry: A Scalable Framework for Fake News Detection via Advanced NLP and Ensemble Learning.” In *Proceedings of the ApplePies 2025 Conference*, Politecnico di Torino, Turin, Italy: Springer, 2025.

Abstract

This research focuses on developing a self-learning, rational-based trajectory design framework for swarms of unmanned aerial vehicles (UAVs). The proposed framework enables autonomous, effective, safe, and energy-efficient trajectory planning, particularly in dynamic and uncertain environments where real-time adaptation, inter-UAV coordination, and collision avoidance are simultaneously required.

The study begins with a comprehensive and critical review of existing approaches to UAV navigation and trajectory planning, comparing traditional, biologically inspired, and artificial intelligence–based algorithms. Through this comparative analysis, key factors such as computational efficiency, scalability, energy consumption, robustness under uncertainty, and the balance between centralized and decentralized control are systematically evaluated. The review also identifies significant research gaps, particularly in real-time adaptation and explainable decision-making, highlighting the need for more flexible, safety-critical, and generalizable strategies in future UAV swarm systems.

Subsequently, the research introduces a novel GA–RF framework, which integrates genetic algorithms (GA) and repulsion forces (RF) to optimize the paths of multiple UAVs, minimizing collisions, overlaps, and interference. Coordination at the cluster level is enhanced through improved task selection and visiting order classification, particularly within the context of the Multi-Traveling Salesman Problem (MTSP). Simulation experiments demonstrate that this approach achieves shorter travel distances, better interference avoidance, and more efficient navigation compared to traditional heuristic and metaheuristic algorithms such as PSO, ACO, SA, and 2-OPT.

The third part of the research presents a novel Active Inference–driven World Modeling for an adaptive UAV swarm Trajectory Design. The proposed framework enables UAVs to autonomously perform mission distribution, route ordering, and motion planning through probabilistic reasoning and self-learning. In the offline phase, expert trajectories are generated using a GA–RF optimizer and used to train a World Model that captures UAV swarm behavior across mission, route, and motion abstraction levels. During online operation, each UAV infers optimal actions by continuously minimizing divergence between current beliefs and reference states encoded in the world model, allowing the swarm to adapt to new targets and environmental changes in real time. The results demonstrate faster convergence, improved stability, and safer navigation compared to Q-Learning, establishing the proposed framework emerge as a scalable and knowledge-based solution for future intelligent UAV swarm networks. Moreover, the proposed model also performed effectively when tested on real-time simulated data, further strengthening its generalization ability and applicability to real-world scenarios.

Overall, this research presents a scalable, secure, and domain knowledge–driven framework for UAV swarm control, offering an effective and promising direction for the advancement of AI-powered autonomous aerial networks.

Table of contents

List of figures

List of tables

Nomenclature

Acronyms / Abbreviations

2-OPT 2-Optimization / Two-Exchange

ABC Artificial Bee Colony

ABS Aerial Base Station

ACO Ant Colony Optimisation

AI-A Modern AI-based Algorithms

AIn Active Inference

APF Artificial Potential Field

APF Auxiliary Particle Filter

BIA Biologically Inspired Algorithms

C–MGDBN Coupled Multiscale Generalized Dynamic Bayesian Network

CTDE Centralised Training and Decentralised Execution

DAgger Dataset Aggregation

DDPG Deep Deterministic Policy Gradient

DE Differential Evolution

DEM Discrete Element Method

DQN Deep Q-Network

DRL Deep Reinforcement Learning

EKF Extended Kalman Filter

EnKF Ensemble Kalman Filter

FEM Finite Element Method

FVs Flying Vehicles

GA Genetic Algorithm

GA-RF Genetic Algorithm Repulsion Force

GNN Graph Neural Networks

GPF Gaussian Particle Filter

GUs Ground Users

HHO Harris Hawks Optimisation

IRL Inverse Reinforcement Learning

JPS Jump Point Search

MACA Multi-Agent Counterfactual Advantage

MARL Multi-Agent Reinforcement Learning

MCMC-PF Markov Chain Monte Carlo Particle Filter

MIP Mixed Integer Programming

MPC Model-Based Control

MTSP-GARF Multiple Travelling Salesman Problem Genetic Algorithm Repulsion Force

MTSP Multiple Travelling Salesman Problem

PF Particle Filter

PIO Pigeon Inspired Optimisation

PPO Proximal Policy Optimisation

PSO Particle Swarm Optimisation

RBPF Rao-Blackwellized Particle Filter

RF Repulsion Force

RL Reinforcement Learning

RVO Reciprocal Velocity Obstacle

SAC Soft Actor–Critic

SA Simulated Annealing

SSA Salp Swarm Algorithm

TA Traditional Algorithms

UAV Unmanned Aerial Vehicles

UKF Unscented Kalman Filter

VOM Velocity Obstacle Method

VO Velocity Obstacle

XRL Explainable Reinforcement Learning

Chapter 1

Introduction

1.1 Motivation and Scope of the Research

Research on unmanned aerial vehicles (UAVs) has made significant progress in recent years, primarily due to their autonomy, flexibility, and multi-agent collaboration capabilities. Initially, these systems were limited to remote-controlled or partially autonomous flights [27]. However, with the increasing complexity of missions, such as long-range surveillance, automated decision-making, and mission distribution research has shifted toward fully autonomous and collective intelligence-based systems.

A single UAV cannot perform large-scale missions due to limitations in energy, coverage, and computational resources. Consequently, the concept of multi-UAV systems emerged, in which overall performance can be enhanced through cooperation and distributed task execution. With the increasing number of UAVs, new challenges such as coordination, collision avoidance, and collective decision-making have arisen, forming the foundation of UAV swarm systems. These systems are based on the principles of Swarm Intelligence and play a key role in future autonomous aerial networks [4, 66].

However, this increasing complexity introduces a crucial question: How can UAV swarm flight speed and trajectory be effectively managed?

The accurate design of speed and trajectory is essential for the efficient operation of autonomous UAV swarms, ensuring system stability, energy efficiency, and timely mission completion [176]. The primary objective of trajectory design is to ensure that a UAV travels from its initial location to the target within defined constraints, completing the mission with minimal time and energy consumption.

This problem is considered NP-hard because, as the number of UAVs and mission objectives increases, the number of possible paths grows exponentially. Typically, this problem is solved in three stages: (1) creating a grid map based on environmental and target information, (2) updating the map with time-dependent features, and (3) selecting the optimal path from the updated map [154, 57].

Classical optimization techniques such as Newton's method [189, 99], gradient descent [168], interior point methods [136], and linear programming [144] have been applied to solve this problem. Although effective for small or well-defined problems, these methods often become trapped in local optima in large or uncertain environments.

To overcome these limitations, researchers introduced metaheuristic and bio-inspired algorithms modeled after natural phenomena, such as Simulated Annealing (SA) [237, 122], Tunicate Swarm Algorithm (TSA) [131], Harris Hawks Optimization (HHO) [49], Grey Wolf Optimizer (GWO) [156], Differential Evolution (DE) [17], Particle Swarm Optimization (PSO) [91], and Genetic Algorithm (GA) [198, 94, 217]. These algorithms provide global optimization in complex search spaces. However, their high computational cost limits their suitability for real-time applications.

As a further development, researchers have proposed multi-objective optimization models that simultaneously minimize multiple objectives such as path length and flight time [172]. For example, [26] introduced an adaptive genetic algorithm for UAV swarms that improved performance by optimizing crossover and mutation operators. Similarly, [223] combined the Multi-Travelling Salesman Problem (MTSP) with a two-dimensional coverage model to enhance target identification, while [193] presented a joint routing and trajectory optimization model that accounted for dynamic constraints. Although these methods are effective, they generally lack real-time updating and contextual learning capabilities.

Considering these challenges, we propose a Genetic Algorithm–Repulsion Force (GA–RF)-based approach for UAV swarm trajectory design. This approach minimizes distance and time while ensuring collision avoidance, velocity interference mitigation, and overlap prevention between UAVs operating in close proximity. Although GA–RF is suitable for global optimal estimation, its computational cost and lack of experiential learning restrict it mainly to offline optimization.

To address these limitations, modern approaches such as Model Predictive Control (MPC) [182] and game-theoretic frameworks [123] have been explored. MPC can predict local trajectories but suffers from rapidly increasing computational complexity as the number of UAVs grows. Similarly, game-theoretic methods achieve local convergence but often compromise global stability.

More recently, Artificial Intelligence (AI) particularly Deep Reinforcement Learning (DRL) [174]—has emerged as a promising paradigm for UAV trajectory design. DRL models such as DQN [195], DDPG [72], TD3 [56], and SAC [201] are capable of learning optimal policies in complex environments. However, they require large datasets, long training times, and frequent retraining, making them unsuitable for uncertain, real-time scenarios. Similarly, Multi-Agent Reinforcement Learning (MARL) [111] encourages cooperation but suffers from policy instability and scalability issues. Approaches such as Imitation Learning (IL) [221] and Inverse Reinforcement Learning (IRL) [151] can learn from expert demonstrations, yet fail to generalize under unpredictable conditions.

In recent years, Graph Neural Networks (GNNs) [13] have been employed to model relational dependencies among UAVs, improving distributed coordination. However, due to their lack of probabilistic reasoning, GNNs perform poorly in uncertain or noisy environments. Similarly, hybrid models, which combine methods such as GA [35] or PSO [120] with other optimization techniques, improve performance but remain unsuitable for real-time trajectory updates because of their high computational complexity.

All these approaches share a common limitation: they are either entirely data-driven but lack reasoning depth, or model-driven but weak in probabilistic adaptation. These models fail to clearly represent UAVs' beliefs, prediction errors, and decision-making mechanisms under uncertainty, leading to unstable and limited performance.

In this context, a recent study [89] attempted to address this gap by proposing an active and explainable model for a single UAV. The model combined a 2-OPT Traveling Salesman with Profits (TSPWP) optimizer with an Active Inference (AIn)-based reasoning framework. It introduced a concept for learning a comprehensive world model for UAVs, derived from offline-optimized demonstrations. The model is organized as a global dictionary representing the decision-making structure of the Traveling Salesman Problem with Profits (TSPWP) and included hotspots, local paths, and full flight trajectories. This world model enabled the UAV to analyze real-time situations, predict the consequences of actions, and classify policies based on expected surprise, facilitating adaptive online planning. However, the framework is limited to individual UAVs and did not consider collective decision-making, synchronous cooperation, or swarm-level coordination.

This limitation serves as the primary motivation for the present research, which extends the concept of active reasoning to UAV swarms. The goal is to enable multiple UAVs to work collaboratively to make collective, autonomous, and adaptive decisions in uncertain environments.

This intellectual continuum leads to Causal Probabilistic Graphical Modeling (CPGM), which provides a systematic and theoretical foundation for active learning and reasoning [207]. The CPGM framework uncovers latent dependencies in sensory data and organizes relationships between various inputs into a probabilistic structure. It enables reasoning under uncertainty through probabilistic inference, integrating information from multiple sources.

By organizing data systematically, CPGM supports reasoning at different levels of abstraction. This process is analogous to human language learning, where sounds form letters, letters form words, and words form sentences. Similarly, a UAV swarm can be trained at multiple levels, such as global planning, trajectory planning, and belief–observation learning, to make more logical, systematic, and flexible decisions in uncertain environments.

This intellectual foundation naturally evolves into the concept of Active Inference, which provides a unified Bayesian framework wherein a generative model establishes the relationships among observations, beliefs, and actions. Active Inference offers an explainable, self-adaptive, and real-time decision-making mechanism [98]. It provides a more comprehensive, theoretically coherent, and practically effective solution than both classical and modern AI techniques, establishing a new scientific direction for UAV swarm trajectory design.

Building on this foundation, the present study proposes an Active Inference–based, goal-oriented UAV swarm trajectory design framework, which is theoretically inspired by Causal Probabilistic Graphical Models [97] and practically supported by the GA–RF optimizer [19] and expert knowledge. This framework enables UAVs to learn probabilistic relationships between observations and beliefs, empowering them to make rational, self-adaptive, and collectively coordinated decisions in uncertain environments.

1.2 Objective and Research Contributions

The primary objective of this research is to develop a comprehensive, self-learning, and rational framework for swarming UAVs that can effectively, safely, and adaptively manage speed and trajectory in real-time. To this end, this research comprises three interconnected phases, with each chapter advancing a specific research goal and making a distinct scientific contribution.

Chapter II provides a comprehensive, systematic, and critical review of current approaches to UAV swarm trajectory planning. The purpose of this chapter is to clarify the comparative links and fundamental differences among various strategies, traditional, biological, and modern AI-based approaches. This section analyzes the structure of centralized and decentralized control architectures, their practical implications, and highlights the fundamental differences in trajectory design and path planning. It introduces the MTSP as a basic mathematical framework for designing UAV swarm trajectories. Furthermore, this chapter discusses online and offline training methods and explains how biologically inspired method based optimizer can be used for the training and real-time adaptation of artificial intelligence–based methods. This chapter highlights fundamental challenges, such as decision-making and collision avoidance. It provides a comparative analysis of the advantages and limitations of various geometric, physical, and APF-based techniques, laying the foundation for future directions in UAV swarm research.

Chapter III, continuing from the research gaps identified in Chapter II, presents a new GA–RF based trajectory optimisation framework that combines genetic algorithms with repulsion forces. The primary objective of this framework is to enhance UAV swarm trajectory design, minimize collisions, overlaps, and interference, and improve coordination among UAVs in multi-town scenarios. For this purpose, an improved GA–RF algorithm is developed that minimizes both travel time and distance while maintaining appropriate separation between UAVs. Furthermore, the GA–RF mechanism is used for efficient routing between nearby cities, which significantly reduces the overall mission distance. A systematic analysis is conducted to determine the optimal size and distribution of UAV swarms, thereby improving both efficiency and coverage, which significantly enhances the system's effectiveness. Comprehensive simulations based on a 2D environmental model are conducted to demonstrate the practical utility and feasibility of the framework. The results show that the GA–RF framework provides more effective navigation, minimizes time, distance, collisions, and interference, and achievies better coverage than traditional metaheuristic algorithms, such as PSO, ACO, SA, and 2-OPT.

Building upon the optimization results presented in Chapter III, Chapter IV introduces a novel Active Inference–based framework for trajectory design in UAV swarms. The proposed approach enables UAVs to autonomously perform mission allocation, path sequencing, and motion planning through a combination of probabilistic reasoning and self-learning. In the offline phase, expert trajectories are generated using the GA–RF optimizer proposed in Chapter III, which are used to train a World Model. This model learns and stores UAV swarm behavior at the abstract levels of mission, route order, and motion. During online operation, each UAV infers the best actions by minimizing the difference between its current beliefs and the reference states encoded in the World Model. This process allows the UAV swarm to adapt to new goals and environmental changes in real time. The results show that the proposed framework achieves faster convergence, better stability, and safer navigation than Modified Q-Learning. Furthermore, experiments were conducted using real drone flights to collect real-time data, which were later used in simulation testing. When the framework is tested on real-time simulated data obtained from these actual drone flights, it also demonstrate effective and generalized performance, which further demonstrates its potential for real-world deployment.

Overview and Research Implications

Overall, this research presents a coherent and progressive strategy that establishes a logical sequence from knowledge-driven optimization to self-learning active reasoning. The second chapter provides the theoretical foundations, the third chapter develops practical optimization models, and the fourth chapter transforms these models into self-learning intelligence through active inference. This unified framework presents a new theoretical and practical approach to UAV swarm control, addressing current limitations while proposing a sustainable, secure, and Bayesian-based direction for the development of future autonomous aerial networks.

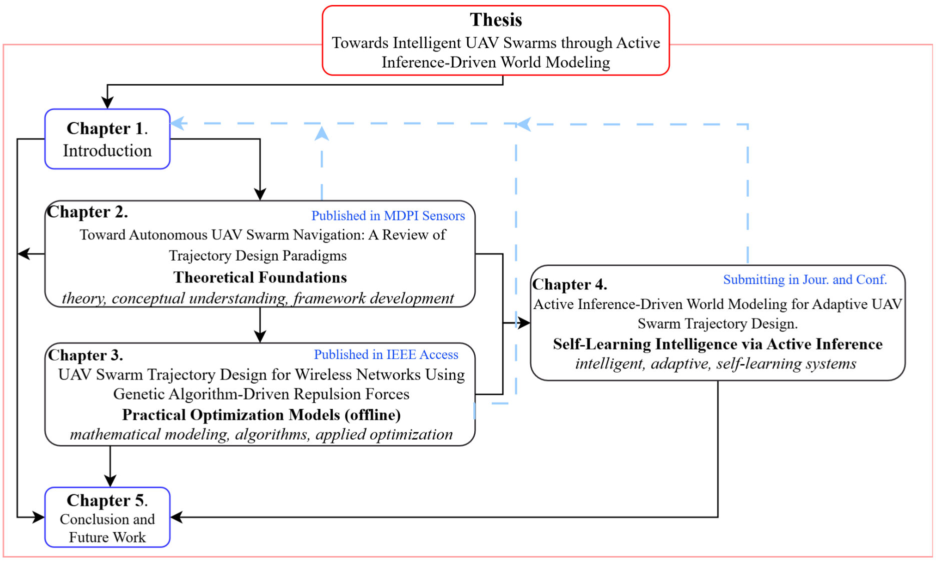


Fig. 1.1 Structure of the thesis

1.3 Outline of the Thesis

The overall structure of the thesis is depicted in Fig.1.1, and organized as follows:

- **Chapter 1** introduces the motivation, problem formulation, and overall objectives of this research. It explains the need for autonomous UAV swarm trajectory design under uncertain environments and highlights the limitations of existing optimization and AI-based approaches.

- **Chapter 2** provides a comprehensive and critical review of the literature related to UAV swarm trajectory planning. It categorizes existing approaches into three main groups, Traditional Algorithms, Biologically Inspired Metaheuristics, and AI-based Methods, and compares them in terms of computational efficiency, scalability, inter-UAV coordination, and robustness in uncertain environments. The chapter also identifies key challenges and research gaps that motivate the proposed framework.

- **Chapter 3** presents the proposed GA–RF optimization framework, designed to minimize time and distance while ensuring collision-free and energy-efficient UAV swarm operation. It integrates a RF mechanism into the GA to enhance spatial separation among UAVs. The chapter also provides extensive simulations and comparative analyses against standard algorithms such as PSO, ACO, and SA to validate the performance of the proposed method.

- **Chapter 4** This chapter presents a self-learning active inference-based framework for goal-directed UAV swarm trajectory design, which builds a causal probabilistic world model from GA–RF optimizer. The model is extended with bayesian decision making, kalman, and particle filters for real-time collision avoidance and adaptation. Experiments on simulated and real flight data show that this framework exhibits faster convergence, better stability, and generalizability than Q-Learning. Overall, this framework provides a scalable, secure, and knowledge-based robust solution for UAV swarms.

- **Chapter 5** summarizes the major findings and contributions of this thesis. It discusses the limitations of the current framework and suggests promising future research directions, including real-world deployment, swarm communication reliability, and reinforcement of Active Inference for distributed intelligence.

Chapter 2

Toward Autonomous UAV Swarm Navigation: A Review of Trajectory Design Paradigms

This chapter presents a comprehensive and critical review of effective and reliable path and trajectory planning strategies for swarms of UAVs. The research classifies existing approaches into three major categories: traditional algorithms, biologically inspired / meta-heuristics, and modern methods based on artificial intelligence (AI). The review provides a comparative analysis of these methods in terms of computational efficiency, scalability, energy consumption, inter-UAV coordination, and robustness in uncertain environments.

In addition, the chapter discusses the strengths and limitations of each algorithm, particularly with respect to collision avoidance, adaptive decision-making, and the balance between centralized and decentralized control. Furthermore, the study highlights hybrid frameworks that combine the global optimization capability of bio-inspired methods with the real-time adaptability of AI-based techniques to achieve a balanced exploration–exploitation trade-off in multi-agent systems.

Finally, the chapter outlines major challenges, such as nonlinear dynamics, multidimensional trajectory spaces, and real-time adaptation, and identifies promising directions for future research. This study serves as a valuable reference for researchers and engineers developing autonomous, intelligent, and integrated UAV swarm systems. The structure of the chapter is depicted in Fig. 2.1.

2.1 Introduction

The basic requirement for the safe and efficient operation of UAV swarms is that each UAV not only plans its trajectory autonomously but also flies in coordination with other UAVs to avoid collisions, resource wastage, and communication bottlenecks [11, 47]. This coordination can depend on both centralized and decentralized control architectures. Centralized systems rely on a central controller that manages the planning of all UAVs, while decentralized systems have each UAV relying on local information and communicating with neighbouring UAVs. Understanding this distinction is crucial for planning the trajectories of UAV swarms.

Moreover, it is essential to differentiate between trajectory planning/design and path planning: path planning primarily focuses on finding the shortest route, while trajectory planning incorporates time, velocity, acceleration, and the physical constraints of the UAV [176]. Trajectory planning in swarm missions is often modelled as a Multiple Travelling Salesman Problem (MTSP), where multiple UAVs must cover different targets while considering mission time, energy constraints, and inter-UAV safety. To address these challenges, the research community has proposed various approaches to trajectory planning. Three major paradigms stand out:

Traditional Algorithms (TA): Deterministic methods such as Dijkstra [43], A [234], and Dubins Curves [110], which rely on complete environmental information and provide optimal or near-optimal paths in well-structured scenarios [55, 100, 188].

Biologically Inspired Algorithms (BIA): Approaches inspired by natural phenomena, such as bird flocking or the pheromone trails of ants, including PSO [175], ACO [7], GA [198], and ABC [6], which provide global optimization in large and complex search spaces [21].

Modern AI-based Algorithms (AI-A): Machine learning [31], deep learning [148], reinforcement learning (RL) [133], multi-agent RL (MARL) [85], and graph neural networks enable UAV swarms to perform adaptive decision-making, collaborative coordination, and intelligent behavior in dynamic, uncertain environments [53, 142]. In particular, modern approaches such as Active Inference [96], based on Bayesian foundations, are introducing new directions in trajectory planning through predictive processing [98].

These approaches are interconnected and form a continuum. TAs provide a foundational structure, BIAs offer global exploration and diversity, and AI-based techniques enable real-time adaptability and intelligent decision-making. In modern research, these methods are being integrated into hybrid frameworks to simultaneously address complex aspects of trajectory design, such as scalability, collision avoidance, and mission-level optimisation.

The main objective of this chapter is to present a systematic, comprehensive, and analytical review of all the essential aspects of UAV swarm trajectory planning, highlighting the clear connections and differences between various approaches.

- This study outlines the fundamental concepts of centralized and decentralized control architectures and their practical applications.
- The fundamental difference between trajectory design and path planning is clarified, and MTSP is introduced as a central mathematical framework that has been effectively adopted in UAV swarm trajectory planning.
- The study discusses online and offline training/testing approaches, detailing how AI-based methods can be trained using an offline-generated BIA-based dataset and subsequently enhanced through online testing and minor adaptations in real-world missions.
- The study clarifies decision-making and collision avoidance as core challenges of UAV swarm trajectory planning and analyses various scientific approaches to solving these problems using geometric, physics-based, and AI-driven techniques.
- This investigation provides a comparative analysis and critically evaluates the strengths and limitations of each approach, ultimately outlining future directions for UAV swarm research.

2.2 Method

This study adopted a formal methodology for conducting systematic reviews following the PRISMA guidelines [146]. The methodology consists of several steps, which are detailed in Fig. 2.2 and explained below:

A systematic search for relevant research articles for this review was conducted in two reliable electronic databases: Web of Science and Scopus. The search process included keywords with "OR" and "AND" operators, incorporating terms such as: ("UAV swarm" OR "drone swarm" OR "multi-UAV") AND ("trajectory design" OR "path planning" OR "trajectory optimisation") AND ("algorithm" OR "control" OR "strategy"), to comprehensively identify all possible and relevant research articles.

A total of 1,743 research articles were retrieved during this phase of the search. The authors then independently screened and selected these articles. Using Zotero software, 832 articles were excluded as duplicate records, while 661 articles were excluded because they

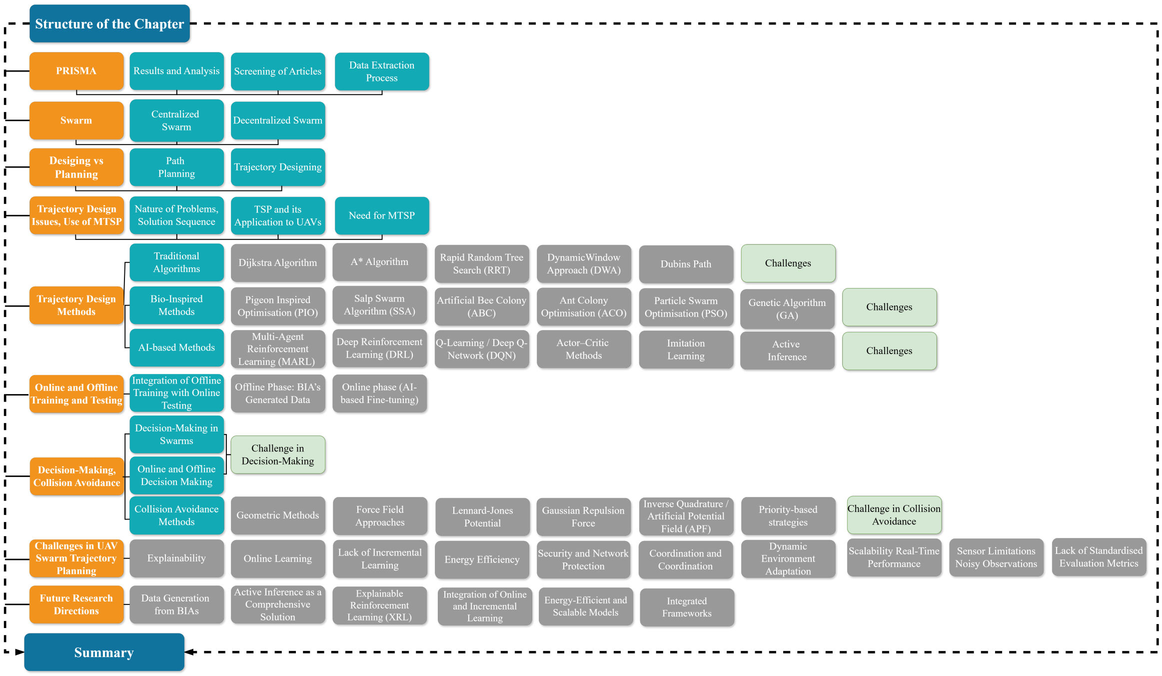


Fig. 2.1 Structure of the chapter.

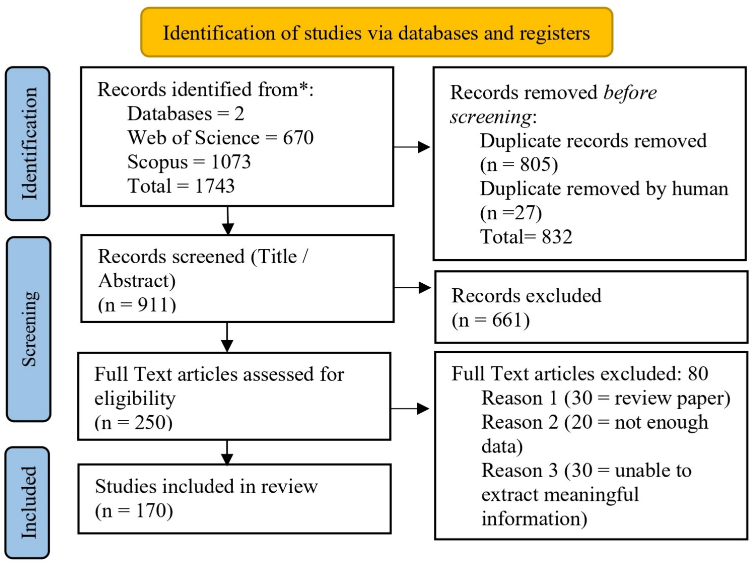


Fig. 2.2 Flowchart of the methodology adopted for selecting papers included in this work, following PRISMA guidelines.

provided only a general overview and did not meet the study's objectives. Therefore, only those articles that met the inclusion criteria were considered for review.

2.2.1 Screening of Articles

One author initially screened the research articles identified through the keyword search based on their titles and abstracts. A total of 911 studies were critically reviewed during this phase. All articles relevant to the topic of this study were included, while irrelevant studies were excluded.

If there was no consensus between the two authors regarding the selection or exclusion of a particular article, the entire article was carefully reviewed. If disagreement persisted, the final decision was made by a third, impartial reviewer to ensure transparency and objectivity.

2.2.2 Eligibility Criteria for Selection of Articles

This review included research articles that met the following criteria:

- The article used keywords such as "UAV swarm", "drone swarm" or "multi-UAV".
- The article included research related to "trajectory design", "path planning" or "trajectory optimisation".
- The article proposed a practical method or technique related to "algorithm", "control" or "strategy".
- The research focused on issues such as collision avoidance, path optimisation, overlapping and interference.
- The study should cover topics that are relevant to the practical application of UAV swarms.

This criterion is established to include only articles that focus on solving the problems of effective, safe, and practical UAV swarm trajectory design and control in real-world contexts.

2.2.3 Data Extraction Process

The extraction of information from the selected research articles is carried out in a systematic and standardised manner. For this purpose, a pre-prepared data extraction form is used, in which the following points are compiled from each study:

Name of the author(s), Year of publication, Objective of the study, Method or algorithm used, Platform or simulator used, Key findings and recommendations of the study, and Research limitations.

Data extraction is performed independently by two authors to minimise bias and ensure the accuracy of the results. In the event of any disagreement, the final decision is made after consulting with a third author. All the extracted information is compiled into a systematic table, which facilitates comparative analysis later.

2.2.4 Results and Analysis

A total of 250 research articles are ultimately included in this systematic review, as per our selection criteria of these 15 are review articles that helped us identify other research studies related to the topic [224, 157, 167, 202, 173, 187, 9, 75, 219, 83, 194, 3, 40, 118, 2]. Additionally, 75 articles are excluded because they do not meet the exclusion criteria.

The selected articles are divided into three main categories based on their research orientation: TA, BIAs and Modern AI-based approaches

The performance of the algorithms presented in each category is evaluated based on several standard metrics, including: Overlapping and Interference of Paths, Obstacle Avoidance, and Optimisation Quality.

The study utilises various tables to present the performance of each algorithm or hybrid approach visually. In these tables, the performance of each approach is presented, allowing for easy comparison of different techniques.

Fig. 2.3 depicts the time distribution of selected citations (2015–2025), showing the number of papers published per year. The graph shows that research on this topic was limited in the early years, but there has been a significant increase in publications since 2020. This trend indicates that UAV swarm trajectory planning has become a rapidly emerging and important research area in recent years.

2.3 Centralized Swarm vs Decentralized Swarm

A swarm is a concept derived from nature, such as a flock of birds, a school of fish, or a colony of ants. It involves several autonomous units (agents) working both in a coordinated and uncoordinated manner, without any central control, and using only local information [222].

In the field of UAVs, a swarm refers to multiple drones or UAVs working together, communicating with each other, and operating under a collective goal, such as surveillance, search and rescue, or enemy identification [84]. There are two basic methods of controlling a swarm:

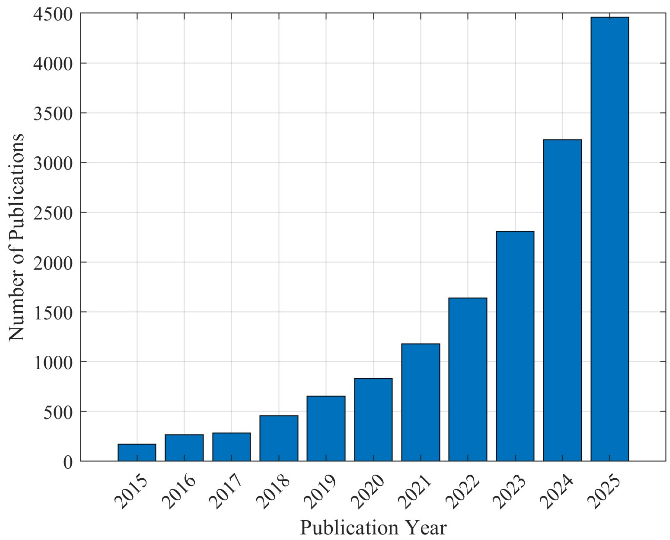


Fig. 2.3 The temporal distribution of selected studies (2015–2025), showing the number of papers published per year in the field of UAV swarm trajectory planning.

2.3.1 Centralized Swarm

A swarm of centrally controlled UAVs is a system in which all drones or UAV subunits are controlled by a single central system, such as a ground control station or a cloud server, as depicted in Fig. 2.4. This central station holds full responsibility for observation, control, and decision-making. Each UAV receives specific instructions, tasks are distributed from this central unit, and each drone follows its defined flight plan or direction. Equation 2.1 presents the states of the centralized swarm.

$$x_i(t+1) = x_i(t) + u_c(t), \tag{2.1}$$

where:

- $x_i(t)$ is the position of UAV i at time t.
- $u_c(t)$ is the control signal sent to all UAVs by the centralized controller.

Examples of controlled centralisation have also emerged in both research and practical applications. The authors in [78, 14] presented a comparative analysis of the performance of centralized control in a study, in which a cloud-based control system guided 12 drones cooperatively. The results showed that centralized control excels in decision-making; however, scalability remains a significant challenge. Similarly, [15] introduced a centralized control-based hybrid AI system for ground surveillance. This system utilises Proximal Policy

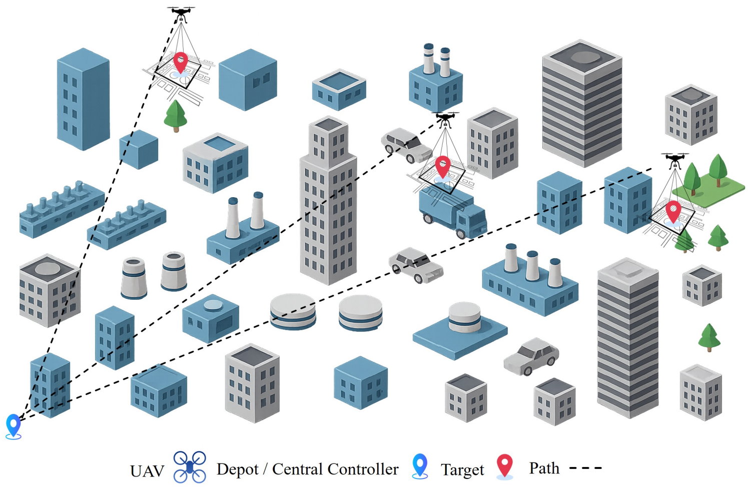


Fig. 2.4 Illustration of a centralized controller.

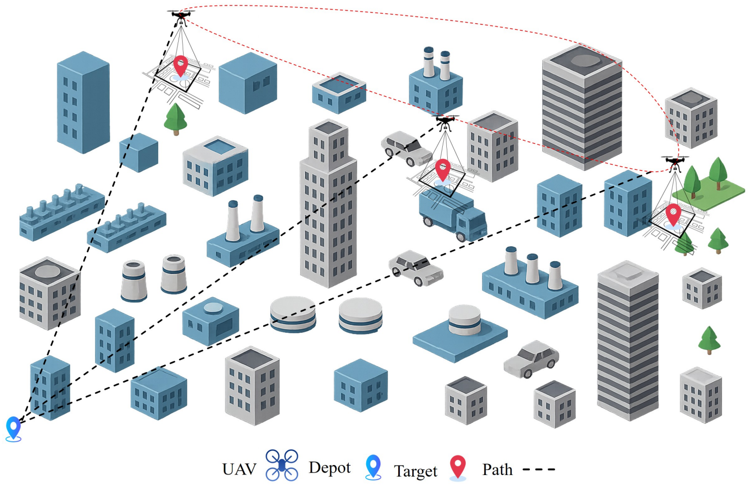


Fig. 2.5 Illustration of a decentralized controller.

Optimisation (PPO)-based reinforcement learning models, where the centralized controller assigns specific search and tracking tasks to different UAVs. The results demonstrate that this centralized structure is effective for both search and continuous tracking.

A study in [159] highlighted that the centralized task assignment mode is the most widely used, in which the Ground Control Station distributes tasks, and each UAV completes its flight. Although this improves the quality of decision-making and ensures that the system follows a coherent strategy, as the number of UAVs increases, challenges such as network communication, real-time response capability, and computational scalability arise. Ultimately, the researchers in [15, 78] agree that centralized control has its advantages, but its challenges cannot be ignored.

2.3.2 Decentralized Swarm

A decentralized UAV swarm is a model in which each UAV makes decisions based on its local information and signals received from neighbouring UAVs, as shown in Fig. 2.5. In this approach, there is no "single point of failure," meaning that if a single UAV fails, the rest of the system continues to function. Various studies have highlighted the robustness and resilience of decentralized models. For example, studies in [159, 69] present UAV coordination models based on decentralized algorithms, which demonstrate the advantages of efficient, low-latency control using local information. This proves that decentralized structures are more suitable for UAV swarms where the communication network is limited or uncertain [36]. Equation 2.2 presents the coordination of the decentralized swarm:

$$x_i(t+1) = x_i(t) + \sum_{j \in \mathcal{N}(i)} a_{ij}(x_j(t) - x_i(t)), \tag{2.2}$$

where:

- $x_i(t)$ denotes the position of UAV i at time t.
- $\mathcal{N}(i)$ is the set of UAVs neighboring UAVs of UAV i.
- a_{ij} is the magnitude of the influence that UAV j has on UAVi.

2.4 Trajectory Design Vs Path Planning

Although these two terms seem similar, there are several fundamental differences between them and their different uses have been repeatedly highlighted in research. For example, [102, 117, 105] define path planning as a method that focuses primarily on finding the shortest path from a starting point to a target, while [170, 238, 164, 16] define trajectory design as the planning of a complete and safe flight path with time, velocity, and acceleration.

2.4.1 Path Planning

The goal of path planning is to find a path from a starting point to a destination with the shortest distance, as shown in Fig. 2.6. This method is primarily used in static environments. It focuses on finding the shortest path based on local or global maps. Simple yet effective algorithms, such as Dijkstra [215] or A* [234], are used to obtain a path with the shortest distance.

The cost function, which is used to minimise the total length of the path, is given below:

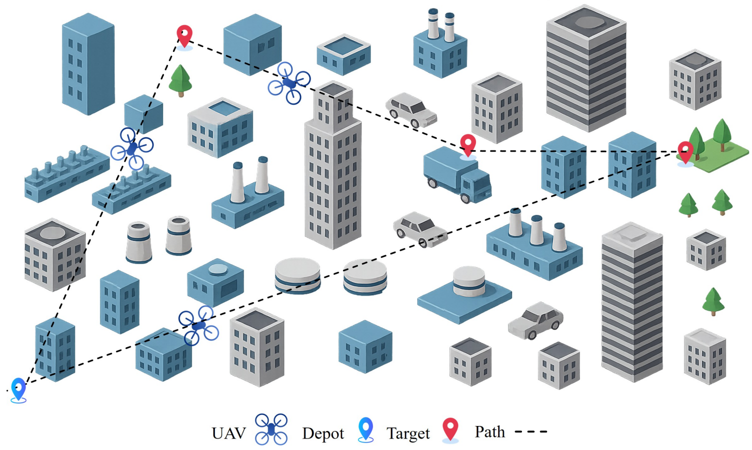


Fig. 2.6 Illustration of path planning.

$$\min \sum_{i=1}^{n-1} \|p_{i+1} - p_i\|, \tag{2.3}$$

where:

- p_i represents the waypoints of the path;
- $|p_{i+1} - p_i|$ is the distance between two consecutive points;
- n is the total number of points the path passes through.

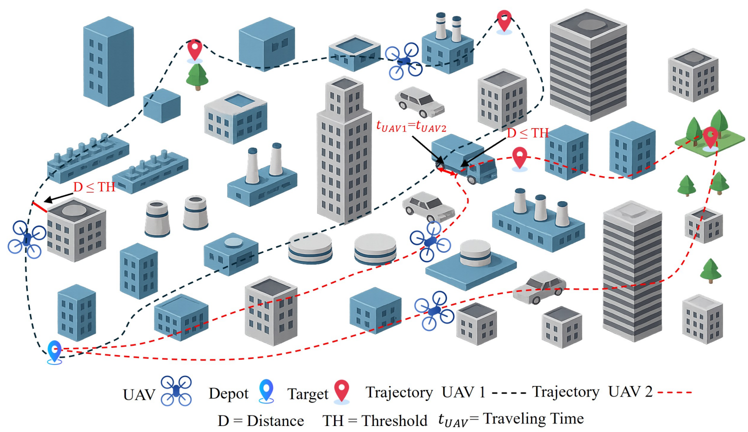


Fig. 2.7 Illustration of trajectory planning/designing.

Table 2.1 A comparative review of path planning and trajectory design.

Aspect	**Path Planning**	**Trajectory Designing**
Goal	Minimum safe path	Smooth and timely path
Time included?	No	Yes
Speed/acceleration included?	No	Yes
Place of use	Map navigation	Autonomous flight, drones, robotics
Nature of environment	Static	Dynamic/uncertain
Complexity	Low	High

2.4.2 Trajectory Designing

Trajectory design involves planning a fully dynamic flight path, including speed, time, angle, and acceleration as shown in Fig. 2.7. This method is commonly used in autonomous drones and robots, where the flight must be not only accurate but also smooth and energy-efficient. For this purpose, a cost function is commonly used to minimise the flight speed and its change (acceleration) [224]. The function given below is based on this principle:

$$J = \int_0^T \left(\|\dot{p}(t)\|^2 + \|\ddot{p}(t)\|^2 \right) dt, \tag{2.4}$$

where:

- $\dot{p}(t)$ is the velocity.
- $\ddot{p}(t)$ is the acceleration.
- T is the mission duration.

Table 2.1 provides a comparative overview of the main differences, application areas, and key technical aspects between path planning and trajectory designing. This comparison reveals that path planning is typically employed to find a safe path in a static environment. In contrast, trajectory design provides a smoother and more time-efficient path in a dynamic and uncertain environment, making it more flexible and better suited to modern UAV missions.

2.5 UAV Trajectory Design Issues and Use of MTSP

2.5.1 Nature of Problems and Solution Sequence

Trajectory design by UAV is a complex problem, especially when the target has to be reached at multiple locations, and the mission duration or energy is limited [161]. The following logical sequence is adopted to solve this problem:

1. Mission Definition: Target points, time limit, and objectives are specified.
2. Modelling: Targets are modelled as nodes, paths as edges, and distance/time as weights [44, 235].
3. Problem Classification:
   - If there is one UAV, $\rightarrow$ TSP [106, 125]
   - If there are multiple UAVs $\rightarrow$ MTSP [24, 141]
4. Trajectory Optimisation: A solution is derived using an appropriate heuristic or AI algorithm, which includes collision avoidance, energy limits, and other practical requirements [24, 141].
5. Simulation or Practical Testing: the performance of the obtained solution is tested.

2.5.2 TSP and its Application to UAVs

If there is only one UAV and it has to visit n destinations, the problem becomes the Travelling Salesman Problem (TSP) [106, 125].

The objective of TSP is to visit all the destinations in the shortest distance or time and finally return to the starting point. The UAV path shown in Fig. 2.6 is a practical example of solving the same TSP problem, where the UAV visits all the targets (waypoints) in a specific order to minimise the total distance.

$$\min \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} \cdot x_{ij}, \tag{2.5}$$

where:

- c_{ij} is the cost of travelling, i.e time and distance from location i to j.
- $x_{ij} = 1$ if the path is chosen.

2.5.3 When it Comes to Congestion: the Need for MTSP

TSP becomes inadequate in the presence of more than one UAV. Therefore, we use the MTSP, which assigns paths to multiple UAVs such that they collectively visit all the destinations in the shortest distance or time, and each UAV eventually returns to its starting point (Dpot) [24, 141]. The UAV trajectory shown in Fig. 2.7 is a practical example of solving the same MTSP problem, where each UAV visits a certain number of targets to minimise the total cost.

Definition and Mathematical Model of MTSP

MTSP is an extended model in which:

- m UAVs (salesmen)
- n targets (tasks or cities)
- Each target is assigned to only one UAV.
- All UAVs start and return from a depot.

Objective of MTSP:

$$\min \sum_{k=1}^{m} \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} \cdot x_{ijk}, \tag{2.6}$$

where:

- $x_{ijk} = 1$ if UAV k goes from location i to j.
- c_{ij} is the distance or time value.
- m is the total number of UAVs.

Application of MTSP to UAV Swarms

The use of MTSP in UAV swarms provides the following benefits:

- Parallelism: All UAVs perform separate missions simultaneously.
- Load Balancing: Fair distribution of targets is possible.
- Time Efficiency: The total mission time is reduced.
- Collision Avoidance: Obstacles are detected and avoided to ensure safe navigation.

While TSP is a suitable solution for UAVs, MTSP provides a very efficient, appropriate and workable framework for UAV swarms [44, 235]. It not only improves speed but also enables missions to be completed in less time and with greater efficiency.

2.6 Different Trajectory Design Methods

Trajectory design is a complex problem, especially when it comes to UAVs or multi-agent systems. There are different strategies to solve this problem, which can be divided into three basic types.

2.6.1 Traditional Algorithms Used in UAV Swarms (in the Context of MTSP)

These TAs are usually used in static or known environments.

Famous Algorithms:

- Dijkstra Algorithm [43]
- A* Algorithm [67]
- Rapid Random Tree Search (RRT) [104]
- Dynamic Window Approach (DWA) [216]
- Dubins Path [110]

UAV swarm-based trajectory design leverages TAs to identify optimal and safe paths to targets. In MTSP scenarios, these algorithms efficiently generate individual UAV trajectories, as illustrated in Algorithm 1, which depicts TAs' operations.

Dijkstra Algorithm and Its Role in UAV Swarms

Dijkstra's algorithm is a classic graph search technique that finds the least-cost or shortest path from one point to all other points. It is beneficial in UAV trajectory design once the MTSP has been solved, as it provides an efficient and shortest path for each UAV to reach its assigned targets [43]. Thus, this algorithm helps to reduce both the time and total cost of mission completion. This concept can be expressed mathematically as:

Algorithm 1 UAV Swarm Trajectory Design using MTSP

Input:
- $V = \{v_0, v_1, \ldots, v_n\}$ (locations, with v_0 as the base station)
- $U = \{u_1, u_2, \ldots, u_m\}$ (set of UAVs)
- $C(v_i, v_j)$ (cost between locations)

Output: Paths P_k for each UAV such that:

$$\min \sum_{k=1}^{m} \sum_{(i,j)\in P_k} C(v_i, v_j), \quad P_i \cap P_j = \{v_0\}, \forall i \neq j$$

Initial Step: Set starting location v_0 for each UAV, and mark all $v_i \in V$ as `unvisited` (except v_0).
while Unvisited nodes exist in V **do**
    for each UAV u_k **do**
        Select the nearest unvisited node:

$$v_{\text{next}} = \arg \min_{v_j \in V_{\text{unvisited}}} C(\text{current}(u_k), v_j)$$

        Add v_{next} to path P_k and mark it as `visited`
    end for
end while
Return: UAVs return to base v_0, with final paths P_k.

$$\min \sum_{(i,j)\in P_k} c_{ij}, \tag{2.7}$$

where:

- P_k: path of UAV k;
- c_{ij}: cost of traveling from node i to node j.

Research on UAV path planning has proposed basic algorithms that typically determine the optimal path from a cost map in a static 2D or 3D grid environment, yielding effective results in simple scenarios. However, these methods are generally limited to single UAV operations and cannot coordinate large-scale UAV swarms [200]. In the same vein, another study designed a pathfinding model for a group of 3–10 UAVs, taking into account battery limits, charging stations and coverage constraints, which provides more effective coverage and better mission completion. However, path overlap remains a key challenge [114]. In another study, the initial paths obtained from classical Dijkstra are improved by PSO to enhance collision avoidance and path selection, resulting in better performance in complex scenarios with reduced path overlap and outperforming classical Dijkstra [215]. Additionally, dynamic planning-based methods, which utilise local replanning with the Bresenham algorithm, have been proposed to avoid unknown obstacles in both static and dynamic environments. They are mainly effective for single UAVs and are capable of handling sudden changes and new obstacles [43]. Although Dijkstra-based method provides reliable routing for UAVs, classical Dijkstra has problems such as synchronization and lack of coordination of large-scale UAV swarms, which make it inadequate for large systems; however, its modern variants such as Multi-UAV Dijkstra and Dijkstra+PSO [215] overcome these weaknesses and provide more reliable solutions within UAV swarms with better coverage, effective collision avoidance, and less interference.

A* Algorithm

The A* algorithm is a heuristic-guided version of Dijkstra, which uses the heuristic function $h(n)$ to speed up the search process [45]. It considers the least-cost path, as well as the estimated remaining distance, in the graph-based search, making it more computationally efficient than the classical Dijkstra algorithm. The cost function in A* algorithm can be expressed as:

$$f(n) = g(n) + h(n), \tag{2.8}$$

where:

- $g(n)$: actual cost of reaching node n;
- $h(n)$: heuristic estimate of remaining distance.

The TA grid-based A* algorithm is utilised for UAV scheduling and routing, providing efficient coordination of 3–10 UAVs while minimising mission overlap through temporal offset batching. To improve upon this, Jump Point Search (JPS)-Enhanced A is introduced, which finds faster paths by skipping unnecessary nodes and gives better results in environments with static obstacles. However, some path overlap is reported during Moving Window Search [150]. As a further development, the 3D A algorithm provided efficient navigation in complex three-dimensional environments using octree-based space partitioning and reduced collisions through per-UAV deflection layers. Still, its performance remained relatively limited in unpredictable dynamic scenarios [108]. In the same sequence, Classification A implemented local A on each UAV by dividing the workspace into sectors, which reduced the computing time and achieved better results [134]. Overall, A* and its variants provide fast, reliable, and effective solutions for UAV trajectory design; however, challenges such as scalability and limited replanning capacity in large-scale UAV swarms and highly dynamic environments remain, which require more hybrid and adaptive approaches to overcome.

Rapidly-Exploring Random Trees (RRT)

RRT is a sampling-based path planning algorithm that rapidly grows new branches through random sampling in a given configuration space, to explore as much accessible space as possible [103, 109]. The following function is used to select the nearest node and extend it in a randomly chosen direction:

$$x_{\text{new}} = x_{\text{near}} + \varepsilon \cdot \frac{x_{\text{rand}} - x_{\text{near}}}{||x_{\text{rand}} - x_{\text{near}}||}, \tag{2.9}$$

where:

- x_{near}: current node in the tree that is closest to x_{rand};
- x_{rand}: randomly chosen point in the direction in which the tree is expanded;
- ε: step size that determines the extent of the expansion;
- $||x_{\text{rand}} - x_{\text{near}}||$: Euclidean distance between the two points, which normalises the direction.

The initial research utilises Multi-platform Space-Time RRT, which enables UAVs to operate in static and cluttered 3D environments with space and time constraints. This model provides smooth and flyable paths, where path overlap is significantly reduced by strictly enforcing the time and separation of each UAV. Another study [227] adopted Multi-RRT with kinodynamic constraints and Bézier curves, which not only provided smoother and shorter paths for 3–10 UAVs but also improved upon methods such as classical RRT and Theta-RRT [92], while ensuring collision avoidance. Meanwhile, RRT is utilised for single UAV scenarios in photogrammetry and aerial survey, where real-time obstacle avoidance is possible with the aid of stereo cameras, and safe navigation at speeds of 6 m/s is demonstrated in practical missions [33]. Furthermore, a hybrid method is introduced that combines iterative RRT with the Salp Swarm Algorithm (SSA), in which SSA intelligently guides the expansion of nodes. This approach reduces path length, decreases the number of iterations and nodes used, improves computational efficiency, and further minimises overlap between UAV paths [139]. Overall, RRT-based algorithms are highly effective in UAV trajectory planning, particularly in complex, dynamic, or partially known environments. Their main strength is fast search; however, the randomness and non-smooth nature of classical RRT often create limitations, which is why modern research is integrating these techniques with Bézier smoothing or SSA-guided approaches to enable smoother, collision-free, and computationally efficient trajectories for UAV swarms.

Dynamic Window Approach (DWA)

DWA is a real-time spatial planning algorithm that selects a safe and feasible path within the UAV's current velocity v and angular velocity (ω). This method is effective because it enables the UAV to avoid collisions even in rapidly changing conditions and complex or partially unknown environments. It analyses possible movements based on velocity and angle, assesses the safety and feasibility of each path, and instantly selects the path that provides the least risk and the most efficiency [50, 29]. In DWA, the objective function is used to select the optimal path, considering various factors such as target alignment, obstacle distance, and speed. Its mathematical expression is as follows:

$$G(v,\omega) = \alpha \cdot heading + \beta \cdot distance + \gamma \cdot speed, \tag{2.10}$$

where:

- v: velocity, ω: angular velocity;
- $heading$: target alignment;
- $distance$: distance from the obstacle;
- $speed$: current speed of the UAV;
- α, β, γ: weights that describe the relative influence of heading, clearance, and speed in decision-making.

This function (Equation 2.10) combines these parameters to produce a score for each possible move, based on which the most suitable move is selected.

DWA has been adopted in various scenarios in UAV swarms to enable quick response and collision avoidance. Several studies have shown that DWA-based approaches not only make the routes safer during missions but also significantly improve the overall efficiency of UAVs. For example, authors in [32] combined DWA with ORCA (Optimal Reciprocal Collision Avoidance), resulting in a 17% reduction in mission time and a 27.9% reduction in path length. A study [183] utilised DWA with gradient-field costs to enable UAVs to navigate more effectively around non-convex obstacles, although gradient sensitivity occasionally led to local minima. Similarly, work [230] combined DWA with global planners such as Jump Point Search (JPS), where the combination of local collision avoidance and global route guidance provided smoother paths. Overall, DWA is a reliable method for real-time local motion planning, enabling UAVs to make swift decisions in dynamic and partially known environments. It provides collision-free trajectories in a short time and improves mission duration. However, for large-scale coordination and nonlinear interactions in complex UAV swarms, DWA typically requires integration with global planners or AI-based intelligence to provide more scalable and adaptive solutions.

Dubins Path

Dubin's path is a classical geometric trajectory planning model designed for vehicles with limited turning radius, and is particularly suitable for fixed-wing UAVs where zero-radius turns are not possible [46, 110]. The model searches for a minimum path that consists of only three basic movements: straight ahead (S), left turn (L), or right turn (R). The combinations of these movements create different possible paths, which can be expressed mathematically as:

$$\text{Path} = \{LSL,\ LSR,\ RSL,\ RSR,\ RLR,\ LRL\}, \tag{2.11}$$

This set represents all the basic possible paths that the model evaluates for minimum distance or cost. In this way, the model compares the performance of each combination and selects the most efficient route, which saves both time and energy in UAV navigation and path planning.

The Dubins path model has been adopted in several studies in UAV trajectory planning. Authors in [137] developed a Dubins-based motion planning framework for fixed-wing UAVs, which is found to be effective for constrained turns and short-path planning. Another study in [208] designed minimum-turn paths for UAVs, which improve trajectory smoothness and mission efficiency in different environments. Furthermore, a work [220] integrated Dubins paths into cooperative UAV swarms, providing collision-free trajectories in a multi-agent path planning scenario despite the turning constraints.

Dubin's path model is a crucial technique for fixed-wing UAV swarms because it incorporates physical constraints, like turning radius, directly into the trajectory planning process. However, this model has limitations; it can only handle straight, constant-radius turns, making it less suitable for dynamic replanning. Therefore, it is often integrated with advanced methods or hybrid approaches in more complex scenarios.

Table 2.2 presents a comparative overview of different TAs in MTSP, showing that each algorithm plays a unique role in specific environments and scenarios. Observations indicate that a combination of different algorithms yields more effective, flexible, and situationally superior results in UAV swarm missions.

Table 2.2 Role of TAs in MTSP with comparison.

Algorithm	Advantages & Limitations	Role in MTSP	Time to complete a mission	Scalability	Energy Efficiency	Collision Rates
Dijkstra-based Algorithms	Efficient in static settings, though limited to a single UAV..	Path for each UAV.	Fast for static; moderate in dynamic.	Poor scalability.	Moderate.	Low in static; increases in dynamic.
A* Algorithm	Faster than Dijkstra, not for dynamic environments.	Heuristic guidance for movement.	Optimal for small swarms; moderate in large tasks.	Limited scalability.	High in static.	Low in controlled; moderate in dynamic.
RRT-based Algorithms	Fast search, effective in dynamic, non-smooth paths.	Obstacle avoidance in dynamic spaces.	Variable, slower in complex areas.	High scalability in a dynamic environment.	Low due to randomness.	Moderate, depends on smoothness.
DWA	Real-time planning, needs a global planner for large swarms.	Collision avoidance in cluttered spaces.	Quick for short-term plans.	Moderate with integration.	Moderate to high.	Low in real-time.
Dubin's Path	Effective for fixed-wing, not dynamic replanning.	Fulfilment of physical constraints.	Fast for fixed, not dynamic.	Low scalability.	High for fixed-wing.	Low if constraints met.

2.6.2 Bio-Inspired Methods Used in UAV Swarm

BIAs are inspired by simple yet effective behaviours found in nature. These heuristic-based methods are highly effective in solving NP-hard problems, such as the MTSP, particularly when designing trajectories for UAV swarms. As demonstrated in Algorithm 2, this illustrates the operational framework of TAs. Some famous algorithms are the following:

- Pigeon Inspired Optimisation (PIO) [199]
- Salp Swarm Algorithm (SSA) [132]
- Artificial Bee Colony (ABC) [6]
- Ant Colony Optimisation (ACO) [7]
- Particle Swarm Optimisation (PSO) [175]
- Genetic Algorithm (GA) [198]

Pigeon-Inspired Optimisation (PIO)

PIO is a BIA based on the navigation abilities, memory, and tendency of pigeons to use the Earth's magnetic field. PIO can be used to navigate UAVs in the right direction toward the global target, providing speed and accuracy in path planning. The algorithm was first introduced by the authors in [199], who described it in two main steps: the map and compass operator, inspired by pigeons' direction recognition and magnetic sensing, and the landmark operator, which reflects pigeons' memory and ability to fly to a target.

In recent research, PIO has been applied to various engineering and optimisation problems. A study in [58] applied PIO to UAV path planning and showed that it can derive paths to the target in less time than TAs. Similarly, Sharma and Panda [160] used PIO in multi-objective trajectory design, where PIO struck a balance between collision avoidance and energy efficiency. Furthermore, Authors in [121] adapted PIO for UAV swarms to provide effective navigation toward the global target even in dynamic and uncertain environments. In the UAV swarm MTSP scenario, the compass-based formula in PIO is used to guide each UAV to the global best position (x_g). This enables coordinated movement of UAVs and efficient multi-target allocation. This method minimises the total travel distance while maintaining swarm coordination and ensuring the avoidance of unnecessary or redundant paths. This compass-based update formula is mathematically expressed as:

$$x_i^{t+1} = x_i^t \cdot e^{-Rt} + x_g, \tag{2.12}$$

Algorithm 2 General Flow of BIAs for UAV Swarm and MTSP

Input:

- $V = \{v_0, v_1, \ldots, v_n\}$: Hotspots (where v_0 is the base station)
- $U = \{u_1, u_2, \ldots, u_m\}$: Set of UAVs
- $C(v_i, v_j)$: Cost (distance, time, or energy) between locations
- Algorithm-specific parameters (e.g., pheromone τ for ACO, velocity v for PSO, etc.)

Output: Optimal paths $\{P_1, P_2, \ldots, P_m\}$ that minimize the total cost:

$$\min \sum_{k=1}^{m} \sum_{(i,j)\in P_k} C(v_i, v_j),$$

with each city visited by only one UAV (except the base station).

Initial step:

Create an initial population/colony/cluster for each BIA:

$$Pop = \{Sol_1, Sol_2, \ldots, Sol_p\},$$

Where each solution is a set of possible paths for the UAVs.

Set initial algorithm parameters (pheromone level, inertia weight, learning coefficients, etc.).

while termination criterion is not met (e.g., max iterations or convergence) **do**

  for each solution $Sol_i \in Pop$ **do**

    Calculate fitness:

$$f(Sol_i) = \sum_{k=1}^{m} \sum_{(i,j)\in P_k} C(v_i, v_j)$$

    Update pheromone (for ACO):

$$\tau_{ij} \leftarrow (1-\rho)\tau_{ij} + \Delta\tau_{ij}$$

    Update velocity and position (for PSO):

$$v_i(t+1) = \omega v_i(t) + c_1 r_1 (pbest - x_i) + c_2 r_2 (gbest - x_i)$$

    Apply selection, crossover, mutation (for GA).

  end for

  Update best solution (global best or optimal).

end while

Output: Extract best solution $\{P_1, P_2, \ldots, P_m\}$, providing optimal or near-optimal MTSP paths for UAVs.

where:

- x_g: global best position;
- R: learning rate that reduces the intensity of the movement over time.

This Equation 2.12 ensures that over time, each UAV gradually moves from its current position to the global optimal position, allowing the entire swarm to complete the MTSP mission in a coordinated and efficient manner.

Salp Swarm Algorithm (SSA)

The salp swarm algorithm (SSA) is a bio-inspired optimisation method inspired by the movement of a swarm of salps in the ocean, where a leader salp moves towards a target and the rest of the salps follow it. SSA is first introduced by the [132], and consists of two stages: the movement of the leader salp that controls the exploration, and the movement of the follower salp that fine-tunes the exploitation.

SSA has demonstrated its effectiveness in various engineering applications over the past few years. For example, [12] utilised SSA for UAV path planning and showed that it can identify the most efficient paths even in complex and dynamic environments. Similarly, authors in [180] implemented SSA in multi-objective optimisation, where energy consumption and path length are optimised simultaneously. Furthermore, the study in [225] extended SSA to complex problems, such as UAV swarm coordination and MTSP, and demonstrated its flexibility.

Leader swarm update equation:

$$x_1^j = \begin{cases} F_j + c_1\big((ub_j - lb_j)c_2 + lb_j\big), & c_3 \geq 0.5 \\ F_j - c_1\big((ub_j - lb_j)c_2 + lb_j\big), & c_3 < 0.5 \end{cases} \tag{2.13}$$

Leader swarm update equation components:

- x_1^j: new position of the leader swarm in dimension j;
- F_j: position of the target (food source) in dimension;
- ub_j: upper bound in the given dimension;
- lb_j: lower bound in the given dimension;
- c_1: exploration coefficient, which decreases with time;

- c_2, c_3: random numbers between 0 and 1.
- If $c_3 \geq 0.5$, the swarm moves towards the target.
- If $c_3 < 0.5$, the swarm moves away from the target, which maintains diversity.

In SSA, the movement of the leader swarm controls the overall direction and behaviour of the entire swarm. In the context of a UAV swarm, the leader swarm can be a UAV that determines the general movement of the swarm towards the target, while the rest of the UAVs follow it. This mechanism is considered ideal for maintaining a balance between exploration and exploitation in complex path planning problems, such as MTSP.

Artificial Bee Colony (ABC)

ABC is a popular bio-inspired optimisation algorithm inspired by the natural foraging behaviour of honeybees. Authors in [86] introduced ABC, which consists of three types of bees employed: onlooker, scout, and worker bees. Each bee plays a role in the process of finding new food (solutions), exchanging information, and making better choices. The ABC algorithm has been successfully applied to various complex problems in engineering and robotics. For example, [6] uses it for numerical optimisation, while [112] shows in UAV path planning that crowd-based cooperation accelerates the search for better paths. In the same vein, [166] applied the ABC approach to multi-objective optimisation in UAV swarms, where the optimal speed and path are determined while considering constraints such as energy, time, and distance.

These studies present the current state of the problem and possible search paths, illustrating that each UAV requires both local and global information to determine the optimal direction. This concept is mathematically represented in the following equation, which is the basic formula for generating a new solution:

$$v_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{kj}), \tag{2.14}$$

where:

- x_{ij}: current solution (the current path or speed of the UAV),
- x_{kj}: neighbouring solutions (other UAVs or alternative paths),
- ϕ_{ij}: a random value that diversifies the search

This update mechanism, as explained in equation 2.14, describes how each UAV combines its current state with neighbouring information to generate a new solution. By applying this equation, improved paths and speeds are achieved, providing fast and effective solutions to complex problems, such as the MTSP. This enables each UAV to determine the optimal path or trajectory in a cooperative manner. The collective intelligence of the UAVs yields faster and more efficient solutions to complex problems.

Ant Colony Optimisation (ACO)

ACO is another important BIA inspired by the natural path-finding behaviour of ants, where ants leave pheromone trails and use them to find the best path. Authors in [38] founded ACO, and it remains a benchmark method for many optimisation problems today.

Study [140] utilised ACO for cooperative search and surveillance missions in UAVs, demonstrating that pheromone-based learning enables effective navigation for UAVs even in dynamic environments. Furthermore, the study in [203] modified ACO to solve UAV-based MTSP and observed that it provides better scalability in parallel UAV coordination.

In MTSP, each UAV is considered as an "ant" searching for the best possible path to reach its target. The initial state of the problem, including all possible paths, as each UAV explores different paths. In this search process, each UAV learns from its own and other UAVs' previous movements to choose the best path for the future. The following probability equation decides this selection:

$$P_{ij} = \frac{[\tau_{ij}]^{\alpha}[\eta_{ij}]^{\beta}}{\sum_{k\in\text{allowed}}[\tau_{ik}]^{\alpha}[\eta_{ik}]^{\beta}}, \tag{2.15}$$

where:

- τ_{ij}: pheromone level, which indicates the previous success of a path;
- η_{ij}: approximate information ($1/\text{distance}$), which gives the immediate availability of the route.

Equation 2.15 helps each UAV calculate which of the following cities or targets is most suitable to choose. The probability of selecting a route with a higher pheromone level and shorter distance increases, while the probability of choosing a path with a lower pheromone level and longer distance decreases.

ACO's pheromone trails provide UAVs with a "collective memory", which is updated after each iteration. This means that when a UAV passes a good route, it leaves pheromones along that route, which other UAVs sense and incorporate into their decisions. This collaboration results in the emergence of optimal routes in the final graph, where each UAV reaches its assigned targets in the shortest distance, time, and energy.

This sequence, initial state $\rightarrow$ decision-making through equations $\rightarrow$ pheromone update $\rightarrow$ optimised paths, helps solve complex problems like MTSP efficiently and consistently.

Particle Swarm Optimisation (PSO)

PSO is a popular bio-inspired meta-heuristic algorithm inspired by the collective behaviour of flocks of birds and schools of fish. Authors in [87, 41] introduced PSO, in which each possible solution is considered a "particle" that explores the solution space by continuously updating its velocity and position.

In recent years, PSO has been widely adopted in UAV path planning and swarm coordination problems. Study in [228] utilised PSO in the trajectory optimisation of UAVs and demonstrated that the algorithm quickly finds near-optimal paths, even in dynamic environments. Similarly, researchers in [218] implemented PSO in UAV-based multi-target assignment (MTSP) and observed that this approach provides better load balancing while maintaining a low computational cost. Furthermore, authors in [155] used an improved version of PSO in UAV swarm collision avoidance, and the results showed that PSO-based coordination is effective in both safety and efficiency.

In MTSP, each particle represents a possible path or velocity and learns from its personal best and the group's global best. The following equation controls the velocity update:

$$v_i^{t+1} = \omega v_i^t + c_1 r_1 (p_i - x_i^t) + c_2 r_2 (g - x_i^t), \tag{2.16}$$

where:

- ωv_i^t: inertial component — maintains the current direction and velocity;
- $c_1 r_1 (p_i - x_i^t)$: cognitive component — movement towards the personal best position p_i;
- $c_2 r_2 (g - x_i^t)$: social component — movement towards the collective best solution g;
- c_1, c_2: learning coefficients;
- r_1, r_2: random factors that diversify the search.

The following equation then updates the position:

$$x_i^{t+1} = x_i^t + v_i^{t+1}, \tag{2.17}$$

where:

- x_i^t: current position;
- v_i^{t+1}: newly updated velocity.

Together, these two equations 2.16 and 2.17 show a process in which each UAV continuously improves its position and velocity, first by taking advantage of its own experience and then by taking advantage of the collective experience of the group. Thus, the collective intelligence of PSO facilitates the identification of optimal paths in MTSP, utilising the minimum distance, time, and energy, and enables real-time swarm coordination.

Genetic Algorithm (GA)

GA are a popular evolutionary optimisation technique based on the principles of natural evolution, such as selection, crossover, and mutation. Goldberg [31, 198] introduced GA as a general framework for complex optimisation problems. Since then, GA has been widely used in various fields, including robotics and UAV path planning.

GA has repeatedly proven its usefulness in UAVs and swarm operations. The authors in [25, 35] utilised GA for UAV mission planning and demonstrated how chromosome-based encoding reduces the total cost (in terms of time and distance) by optimising multiple paths. Furthermore, the researchers in [209] utilised GA for UAV trajectory optimisation in the context of the MTSP, which demonstrated significant improvements in load balancing and mission completion time among UAVs. Similarly, the studies [42, 68] implemented GA in UAV-based collision-free path planning, and the results showed that GA-based approaches remain efficient and scalable even in large search spaces.

In MTSP, the GA represents each possible UAV path as a chromosome, where genes represent the sequence of cities or targets that the UAV can visit. The goal of the GA is to find the solution among these paths that provides the least cost (distance or time). The following fitness function is used to measure this performance:

$$Fitness(x) = \frac{1}{\text{Cost}(x)}, \tag{2.18}$$

where:

- $Cost(x)$: total cost of the UAV path, measured in terms of distance or time.

Equation 2.18 ensures that the lower the cost of the path, the higher its fitness. As a result, the GA naturally prefers low-cost and high-fitness paths.

The GA iteratively generates new solutions:

1. **Crossover**: creates a new path by combining two existing paths.

2. **Mutation**: creates diversity by making minor changes to the path.

3. **Selection**: selects paths with better fitness for the next generation.

With each iteration, weaker solutions are eliminated and stronger solutions become more dominant, until all UAVs agree on an optimal or closest solution. The final part presents the results of this evolutionary process, where non-conflicting and low-cost paths for the UAVs emerge. Thus, GA's evolutionary search enables the solution of complex problems, such as MTSP, quickly and efficiently, whether the problem involves trajectory planning, path allocation, or real-time swarm coordination.

Table 2.3 Role of BIAs in UAV swarm and comparison.

Algorithm	Advantages & Limitations	Role in MTSP	Time to complete a mission	Scalability	Energy Efficiency	Collision Rates
PIO	Simple, effective for basic navigation; limited adaptability.	GPS-like orientation.	Moderate, struggles with complexity.	Low in dynamic tasks.	Moderate.	Low in controlled settings.
SSA	Easy to implement; poor in dynamic environments.	Leader-follower coordination.	Moderate, slow in complex tasks.	Moderate for small swarms.	Low to moderate.	Low in simple environments, higher in dynamic.
ABC	Balanced exploration and exploitation; slow convergence.	Workers find solutions and share info.	Slow due to convergence.	Moderate in large spaces.	Moderate.	Moderate, varies with exploration.
ACO	Effective pathfinding; suffers from pheromone imbalance.	Collective memory via pheromones.	Moderate, slow in large environments.	High in dynamic tasks.	High due to pheromone updates.	Moderate, depends on pheromone strength.
PSO	Fast convergence; can stagnate in local optima.	Fast, optimal solutions.	Fast for continuous tasks.	High in continuous optimisation.	High in controlled settings.	Low, may increase with stagnation.
GA	High diversity; slow convergence, high computational cost.	Global solutions via mutation and crossover.	Moderate, slow in complex spaces.	High in large solution spaces.	Moderate.	Low with maintained diversity.

2.6.3 Challenges in Bio-Inspired Algorithms

In the context of UAV swarms, several BIAs have been effectively adopted to solve complex combinatorial problems such as the MTSP. A specific natural phenomenon or organism inspires each algorithm, which then performs in UAV swarms with its unique mechanisms and advantages. However, each algorithm also has some limitations, which subsequent methods aim to address and improve. Table 2.3 summarises these algorithms, describing the basic motivation of each algorithm, its role in UAV swarms/MTSP, and the main challenges.

This evolutionary sequence illustrates that each new algorithm overcomes the weaknesses of its predecessors to some extent. For example, PIO relies on basic GPS-like navigation behaviour; however, it often fails to reach the global optimum. This shortcoming is partially addressed by SSA, which introduced a simple leader–follower strategy; however, it also proved to be limited in more complex and dynamic environments.

Then ABC improved exploration by modelling the foraging activity of worker bees; however, it took longer in large search spaces due to slow convergence. ACO introduced collective learning through cooperative pheromone trails; however, it suffered from problems such as premature convergence and pheromone evaporation.

PSO provided an effective yet simple coordination mechanism by combining individual and collective best (personal best and global best). Still, it often got stuck in local minima due to the difficulty in maintaining diversity. Finally, GA emerged with an evolutionary mechanism that provides substantial diversity through crossover and mutation, offering a highly reliable and robust solution to complex combinatorial problems, such as MTSP [31, 209].

However, a fundamental limitation of GA is that it is primarily suited for offline scenarios, where all the data is already available. In online situations such as real-time UAV coordination, the computational complexity and latency of GA can limit quick decision-making. Therefore, while GA performs well in offline mission planning, either lightweight algorithms or hybrid approaches may be more effective for online decision-making [228, 42].

2.6.4 AI-based and Innovative Methods

In recent years, artificial intelligence-based methods have emerged as a crucial alternative for solving complex combinatorial problems, such as UAV swarm trajectory design and the MTSP. These innovative approaches have provided more adaptive, scalable, and data-driven solutions than TAs. AI-based algorithms enable UAVs to make autonomous decisions in changing environments and derive optimal routes in complex situations [191, 8].

Algorithm 3 AI Techniques for UAV Swarm and MTSP

Input:

- $V = \{v_0, v_1, \ldots, v_n\}$: Hotspots (with v_0 as the base station)
- $U = \{u_1, u_2, \ldots, u_m\}$: Set of UAVs
- $C(v_i, v_j)$: Cost (distance, time, or energy) between locations
- AI-specific parameters (e.g., learning rate, neural network structure, etc.)

Output: Optimal paths $\{P_1, P_2, \ldots, P_m\}$ that minimize the total cost:

$$\min \sum_{k=1}^{m} \sum_{(i,j) \in P_k} C(v_i, v_j),$$

with each city visited by only one UAV (except the base station).

Initial Step:

Initialise neural network weights, or reinforcement learning environment.

Set starting locations for each UAV v_0.

while Not converged (e.g., max epochs, acceptable error) **do**

    for each UAV u_k **do**

        Input current state (current location, previous path, etc.) into the AI model.

        Output next location for UAV:

$$v_{\text{next}} = \text{AI\_model}(state)$$

        Add v_{next} to UAV path P_k.

        Update model parameters based on the UAV's decision (Reinforcement Learning: update Q-value or loss function).

    end for

end while

Return: Extract best solution $\{P_1, P_2, \ldots, P_m\}$, providing optimal or near-optimal MTSP paths for UAVs.

Popular AI-based methods:

- Multi-Agent Reinforcement Learning (MARL) [231]
- Deep Reinforcement Learning (DRL) [229]
- Q-Learning / Deep Q-Network (DQN) [70]
- Actor–Critic Methods [8]
- Imitation Learning [145]
- Active Inference [51]

These AI-based approaches have ushered in a new era for UAV-based MTSP and trajectory planning, where UAVs not only operate according to pre-programmed rules but also adapt and perform effectively in complex, real-world scenarios, with the ability to learn and make autonomous decisions. As shown in Algorithm 3, this outlines the operational framework of AI-based methods for UAV swarm.

Multi-Agent Reinforcement Learning (MARL)

MARL is an extension of traditional reinforcement learning in which multiple agents learn and act together in the same environment [28, 231]. In MARL, each agent not only receives rewards and observations from the environment, but is also influenced by the presence and decisions of other agents. This feature is particularly suitable for UAV swarms because each UAV acts as an agent that determines its trajectory and decisions by taking into account the behaviour of other UAVs.

In recent years, MARL has been widely used for UAV swarm trajectory planning, MTSP, and cooperative decision-making. For example, the study [76] proposed a MARL-based framework for UAV swarms, which enables UAVs to jointly find optimal routes and share tasks (i.e., task allocation). Similarly, work [39] used a MARL model based on centralized training and decentralized execution (CTDE) for UAV collision avoidance, which provides better coordination in real-time decisions. Furthermore, authors in [211] demonstrated that MARL enables UAVs to be cooperative and adaptive in dynamic MTSP scenarios, particularly in environments where targets and routes change over time.

Fig. 2.8 illustrates the concept of MARL, where each UAV makes autonomous decisions based on its local observations and the rewards it receives. Each UAV learns not only from its own experience but also from the behaviour of other UAVs, allowing for better collective decision-making. This process can be described mathematically by the following objective function:

$$\pi_i^* = \arg\max_{\pi_i} \mathbb{E}\left[\sum_{t=0}^{T} \gamma^t r_{i,t} \,\middle|\, \pi_1, \pi_2, \ldots, \pi_n\right], \tag{2.19}$$

where:

- π_i: policy of agent i, which chooses an action based on current observations;
- $r_{i,t}$: reward received by agent i at time t;
- γ: discount factor, which maintains the importance of long-term rewards;
- $\pi_1, \pi_2, \ldots, \pi_n$: policies of all other agents, which influence the environment and decisions.

Equation 2.19 specifies that each UAV optimises its policy in such a way that the long-term total reward is maximised, while also taking into account the behaviour of other UAVs.

The MARL's frequently updated decisions enable UAVs to learn from each other, taking paths that avoid collisions, reduce time and distance, and successfully solve complex problems, such as MTSP, in dynamic and uncertain environments.

Thus, MARL provides an effective solution for UAV swarms, enabling them to adapt in real time and collectively adopt the best strategy [222, 212].

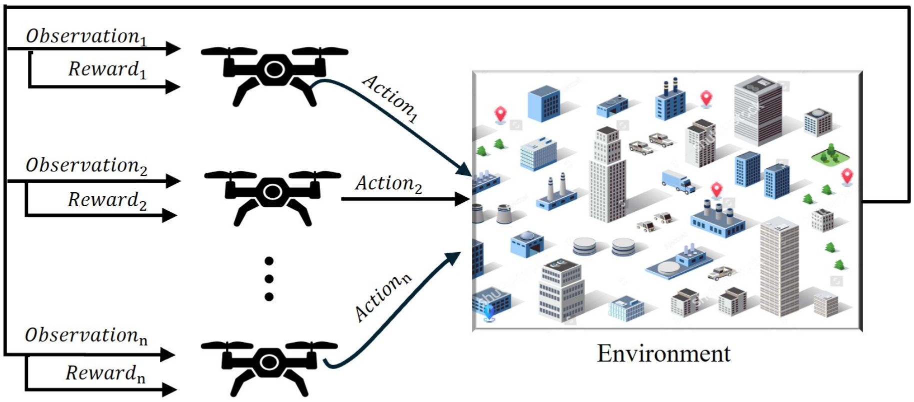


Fig. 2.8 Example of a MARL framework for UAV trajectory planning.

Deep Reinforcement Learning (DRL)

DRL is a modern learning method where an agent observes the environment, performs actions, and improves its policy based on rewards [133, 20]. DRL combines the principles of classical reinforcement learning with deep neural networks, allowing it to learn efficiently even on high-dimensional inputs such as images, sensor data, and complex state spaces.

DRL has been widely used in complex combinatorial optimisation problems such as UAV trajectory design and MTSP. For example, the article [236] proposes a policy framework based on DRL for UAV swarms, allowing UAVs to perform dynamic task allocation and real-time trajectory adjustments. Similarly, the study [192] demonstrated that DRL enables UAVs to make adaptive routing decisions in response to changing situations during mission execution. Furthermore, the research [53] achieved significant improvements in both load balancing and mission completion time by implementing DRL in an MTSP setting. This process can be described mathematically by the following objective function:

$$\pi^* = \arg\max_{\pi} \mathbb{E}\left[\sum_{t=0}^{T} \gamma^t r_t\right], \tag{2.20}$$

where:

- π: policy that describes the strategy for choosing the action;
- r_t: reward received at time t;
- γ: discount factor that balances long-term and short-term rewards.

Equation 2.20 explains that in DRL, the UAV optimises its policy π in such a way that the long-term total reward is maximised. After each observation, the UAV estimates which action in the current state will yield the most benefit in the future and updates its decisions accordingly.

The result of this iterative process is that the UAVs have learned from the environment and adopted better paths and target preferences for the MTSP. This has not only increased mission performance but also reduced execution time. Thus, DRL enables UAV swarms to operate effectively in dynamic and uncertain environments and automatically select the best paths [229, 8].

Q-Learning / Deep Q-Network (DQN)

Q-Learning is a classical value-based reinforcement learning technique that learns the expected reward for each state-action pair and ultimately produces an optimal policy [205].

Fig. 2.9 demonstrates the fundamental framework of Q-learning and DQN. After receiving State and Reward from the environment, agents update the Q-Table to learn which Action is best in which state, and this knowledge helps to improve subsequent decisions. The principle of this update is described in the following equation:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha \left[r_t + \gamma \max_{a'} Q(s_{t+1}, a') - Q(s_t, a_t) \right] \tag{2.21}$$

where:

- $Q(s_t, a_t)$: estimated value of action a_t in the current state s_t;
- r_t: reward received after executing the action at time t;
- s_{t+1}: next state;
- $\max_{a'} Q(s_{t+1}, a')$: highest-valued possible action in the next state;
- α: learning rate, which determines the weight of new and old Q-values;
- γ: discount factor, which determines the importance of future rewards.

Equation 2.21 describes how Q-learning updates the Q-value by combining new information with old information. The UAVs repeat this process repeatedly, learning which action will provide the highest reward in each situation. The result of this learning process is that the UAVs have adopted routes and task allocations that not only reduce distance and time but also avoid collisions in a dynamic and uncertain environment. Thus, Q-learning enables both real-time route optimisation and dynamic task allocation in MTSP, improving the overall performance of the swarm [204, 229].

Q-Learning enhances the discrete decision-making capabilities of UAVs, whereas DQN addresses large state spaces. The work in [204] proposes DQN-based trajectory planning for UAVs and observes better performance in complex urban settings. DQN combines the same principle with DQN to address high-dimensional state spaces, as demonstrated by the authors in [133] for human-level decision-making.

Recent research has used Q-Learning and DQN for complex combinatorial optimisation problems such as UAV trajectory planning and the MTSP. For example, the study [70] used DQN in UAV swarms to improve real-time path selection and reduce mission completion time in dynamic scenarios. Similarly, the work [53] presented a Q-learning-based task allocation approach for multi-UAV MTSP, which significantly improved load balancing among cooperative UAVs. Furthermore, the research [232] employed a DQN-based approach for UAV collision avoidance and adaptive navigation, yielding promising results in complex environments.

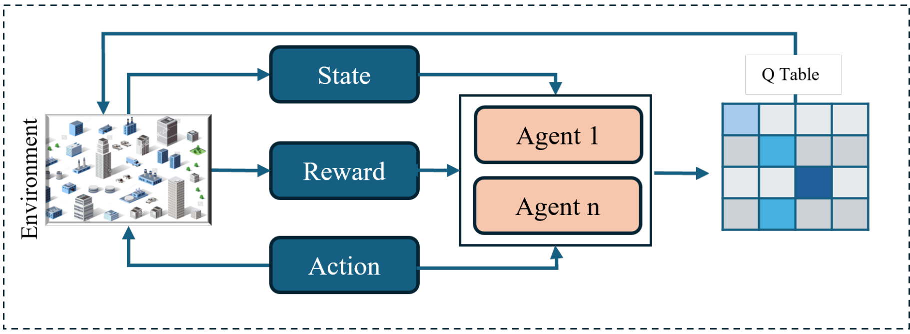


Fig. 2.9 Illustrative example of Q-Learning / DQN approach for UAV-based MTSP

Actor–Critic Methods

Actor–Critic is one of the primary reinforcement learning methods that combines policy-based and value-based approaches [95, 64]. These methods consist of two main parts, as illustrated in Fig. 2.10:

Actor: which chooses an action and learns a policy $\pi(a|s)$. Critic: which estimates the value of the selected action ($V(s)$ or $Q(s,a)$) and provides feedback to the actor.

These methods are particularly suitable for problems where the action space is continuous, such as speed, angle, or throttle control, because they require precise and smooth control at each step [65]. In the initial scenario, UAVs must not only decide which path to take but also make smooth adjustments to speed and angle while following that path, so that mission time is short and energy use is efficient. In such cases, the Policy Gradient update rule is used, which adjusts the policy parameters in such a way that the expected total reward is maximised:

$$\nabla_\theta J(\theta) = \mathbb{E}_{\pi_\theta}\left[\nabla_\theta \log \pi_\theta(a|s) \cdot A(s,a)\right], \tag{2.22}$$

where:

- θ: policy parameters;
- $\pi_\theta(a|s)$: probability of choosing action a in state s;
- $A(s,a)$: advantage function, which expresses the utility of an action relative to the average.

Actor–critic methods have been used in UAV swarm research through several advanced implementations:

- Proximal Policy Optimisation (PPO): The article [65] introduced PPO, which is a stable and sample-efficient Actor–Critic algorithm. For UAVs, PPO-based frameworks have been successfully adopted for dynamic mission planning and MTSP coordination [8].

- Deep Deterministic Policy Gradient (DDPG): The authors in [213] proposed DDPG for continuous control. In UAV swarms, DDPG is utilised to learn continuous parameters, such as velocity and angle, resulting in smoother trajectories.

- Soft Actor–Critic (SAC): It is an Actor–Critic variant based on maximum entropy RL, which provides a better balance between exploration and exploitation. SAC has shown promising results in UAV collision avoidance and coverage scenarios [34].

- Hybrid Multi-Agent Actor–Critic Approaches: Huang et al. [79] used the Actor–Critic architecture in the multi-agent counterfactual advantage (MACA) framework, which reduced collisions in UAV swarms by 90% and improved cooperative behaviour.

Actor–critic methods enable UAV swarms to make adaptive decisions in complex and continuous action domains. In the context of problems such as MTSP, these approaches would allow UAVs to manage the trade-off between local observations and global mission objectives; however, they also present challenges in terms of computational complexity and scalability in large-scale swarms [64, 229].

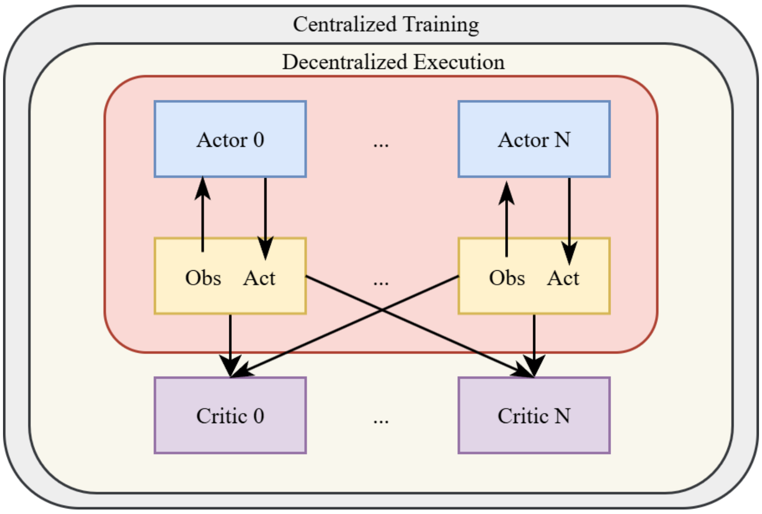


Fig. 2.10 Representation of the Actor–Critic framework for UAV swarm trajectory optimisation.

Imitation Learning

Imitation Learning is a learning method based on the principle that a model learns to make better decisions by following the demonstrations of experts [145, 80]. As depicted in Fig. 2.11, it uses data provided by human operators or expert agents to learn a new policy that performs the same actions as the expert. This method is more efficient than reinforcement learning, because it learns from expert demonstrations rather than "trial-and-error".

Imitation learning is particularly effective in the context of UAV trajectory planning and the MTSP. For example, Kim et al. [93] employed an imitation learning framework for UAV swarms, enabling UAVs to replicate expert trajectories and enhance cooperative formation flying. Similarly, Wan et al. [197] proposed the DAgger (Dataset Aggregation) algorithm, which enhances learning robustness through iterative expert corrections in UAV navigation and decision-making. Furthermore, Pan et al. [147] combined imitation learning with deep neural networks in UAV-based MTSP missions to significantly reduce planning time and increase mission efficiency. Imitation learning approaches have been combined in multi-agent setups for UAV swarms, as in Zhang et al. [233], who developed a hybrid imitation–reinforcement learning framework that initialises UAVs with expert data and then further improves performance through reinforcement learning.

In behaviour cloning, the goal of the model is to replicate the behaviour of the expert with maximum accuracy. To achieve this goal, a specialised loss function is used, which measures the difference between the predicted action and the actual action of the expert. The mathematical expression for this loss is as follows:

$$L(\theta) = \sum_{(s,a)\in D} \|a - \pi_\theta(s)\|^2, \tag{2.23}$$

where:

- D: training data set, consisting of pairs (s,a); where s is the state and a is the expert action;
- $\pi_\theta(s)$: action predicted by the policy network;
- a: actual action of the expert;
- $\|a - \pi_\theta(s)\|^2$: squared error between the prediction and the actual action.

This loss function teaches the policy to replicate the expert's actions as accurately as possible. The higher this error, the greater the loss, and the model will reduce this difference by updating its parameters θ.

Imitation learning not only enables UAVs to learn rapidly from expert demonstrations but also provides data-efficient and low-cost training for complex multi-target missions such as MTSP. However, expert data collection and domain shift can be a challenge in large-scale UAV swarms [145, 80].

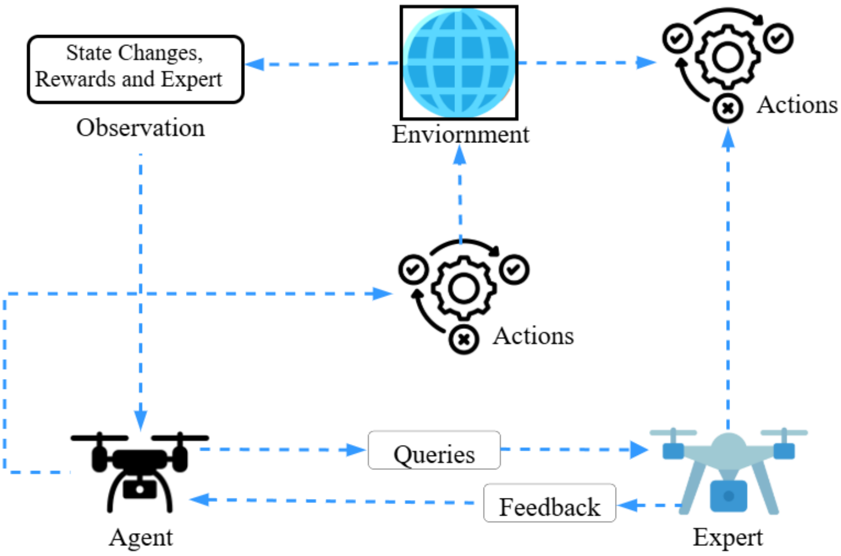


Fig. 2.11 Representation of imitation learning in UAV trajectory planning.

Active Inference

In active inference, decisions are based on the principles of free energy or surprise minimisation, which are inspired by theoretical models of the human brain and have been adapted to machines. It is an emerging probabilistic decision-making framework based on Bayesian theory, integrating prediction, planning, and action under a unified framework [51, 52]. In this approach, illustrated in Fig. 2.12, the agent constructs a generative model that captures the world's model. Through this model, the agent minimises the gap between the expectations of sensory input and the actual observations. This gap is called free energy, and minimising it allows the agent to make more adaptive decisions. This function measures the deviation between the agent's belief and the model, expressed mathematically as:

$$F = \mathbb{E}_{q(s)}[\log q(s) - \log p(s,o)], \tag{2.24}$$

where:

- $q(s)$: posterior belief of the agent about a state s;

- $p(s,o)$: generative model, which represents the joint probability of state s and observation o.

This function forces UAVs to learn in such a way that the difference between their belief and the actual model is minimised.

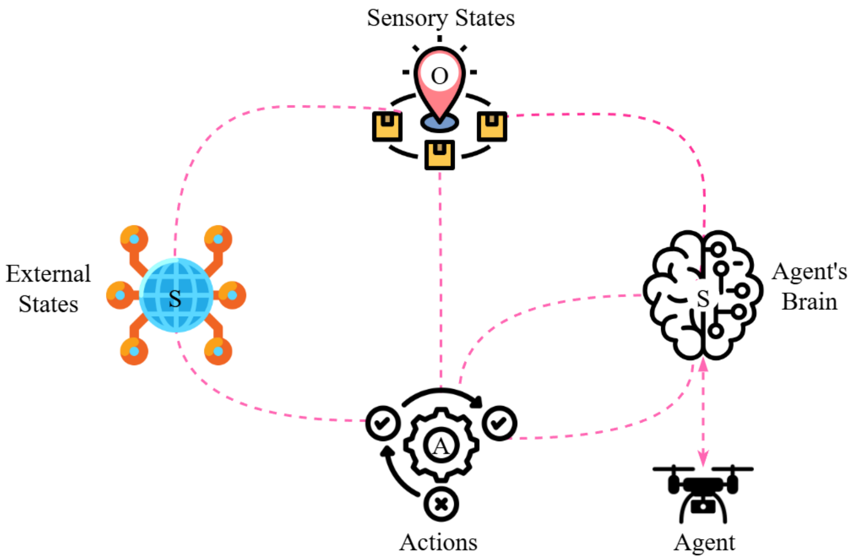


Fig. 2.12 Representation of the active inference framework for UAV trajectory planning.

In the context of MTSP, this method enables UAVs to design a trajectory and path based on predictions, thereby enhancing adaptation and facilitating real-time adjustments during the mission. As a result, UAV swarms reach their targets with greater precision and coordination, using minimal energy, regardless of the uncertain environment.

Applications of active inference have emerged in UAV research in recent years. For example, a goal-directed approach includes the TSPWP world model, which provides dictionary-based planning for effective flight by minimising surprises to a UAV in areas with wireless coverage. This model shows better results than Q-learning in terms of decision-making speed and stability, although further experiments are needed for full integration at the swarm level. [97]. In the same vein, Active-MGDBN (a hybrid of Gaussian Dynamic Bayesian Network) is introduced, which provides autonomous path planning and self-supervision, and increases flight flexibility and speed by suggesting optimal paths based on assumptions in an unknown network environment. At the same time, it does not require training on specific datasets, as it is capable of learning autonomously [98]. Another model is inspired by the decision-making style of human drivers, where decisions are made based on Bayesian cognition and free-energy minimisation. Although this model has not yet been directly applied to UAVs, its theoretical relevance makes it readily extensible to challenges such as UAV collision avoidance [171]. Smith et al. [181] showed that Active Inference enables UAVs to make successful decisions even in partially observable and dynamic environments. In contrast, Pezzulo et al. [153] provided predictive awareness to UAVs during missions through Bayesian active inference models. Furthermore, Millidge et al. [130] proposed a deep Active Inference framework that combines generative models with deep neural networks for UAVs, showing encouraging results in complex scenarios such as multi-target planning. Overall, Active Inference enable adaptive and prediction-driven decision-making capabilities to a UAV swarm. It provides a strong theoretical foundation through which UAVs can learn stable navigation in uncertain environments and effectively achieve speed, coordination, and continuously updated strategies during complex missions.

2.6.5 Challenges in AI-based Algorithms

In the context of UAV swarms, various AI-based methods are employed to solve complex problems, such as the MTSP, effectively. Each algorithm solves a problem more effectively based on its learning style and neural processing; however, it also has some weaknesses. In recent years, several research works have demonstrated how one method succeeds another and overcomes its shortcomings, ultimately leading to the emergence of a generative and explainable short language model as a robust and unified framework [229, 231, 181].

Table 2.4 presents a comparison of the basic concepts and roles of different AI-based methods in UAV swarms and MTSP. It shows that each method is effective in specific situations but has its limitations; therefore, a combination of different AI techniques can be more flexible, scalable, and provide better results in uncertain environments.

This study begins with MARL, which is designed for multi-agent coordination and cooperative task allocation in UAV swarms [28, 76, 211]. MARL gave UAVs the ability to learn and cooperate; however, it still had problems such as scalability, communication overhead, and multi-agent credit assignment.

Then came DRL, which is capable of learning whole mission-level policies [133, 236]. However, DRL requires large amounts of data, time, and computational resources. This limitation is alleviated by value-based methods such as Q-Learning/DQN, which are effective for small and discrete action spaces. However, they are not suitable for continuous UAV control [204].

Table 2.4 Using AI-based methods in UAV swarm and comparison.

Method	Advantages & Limitations	Role and Challenges in UAV Swarm	Time to complete a set mission	Scalability	Energy Efficiency	Collision Rates
MARL	Cooperative. Poor scalability.	Learns policies, but scalability is challenging.	Moderate, depends on coordination.	Moderate for large swarms.	Moderate.	Moderate, complexity impacts coordination.
Deep RL	Full learning. High data and training costs.	Learns complex tasks, but slow due to data needs.	Slow due to training.	Low scalability for dynamic tasks.	High computational cost.	Low, but increases in dynamic settings.
Q-learning / DQN	Fast for small tasks. Struggles with large spaces.	Effective in small spaces, but poor in complex ones.	Fast for small, discrete tasks.	Poor for continuous spaces.	Moderate.	High in small tasks, higher in complex ones.
Actor–Critic	Continuous control. Unstable without tuning.	Effective for continuous tasks but needs tuning.	Moderate, depends on tuning.	Moderate for complex tasks.	Moderate.	Moderate, instability increases risk.
Imitation learning	Fast learning. Poor generalisation.	Learns fast, struggles with new scenarios.	Fast, but weak in new environments.	Low scalability in dynamic tasks.	Low energy.	High in controlled, low in new environments.
Active Inference	Adapts with limited data. Needs robust models.	Adapts well with minimal data, but needs efficient models.	Fast with minimal data.	High scalability with tuning.	High due to computational cost.	Low, adaptive nature helps avoid collisions.

The next is Actor–Critic methods, which combine policy and value learning and are effective for continuous actions, such as speed and angle [65, 213]. However, these methods can be unstable without hyperparameter tuning. Imitation Learning took a step further, enabling UAVs to learn rapidly based on expert data [93, 197]. However, when new or unforeseen situations arise, it demonstrates limited adaptability.

After addressing these problems, Active Inference emerged as a promising solution, based on Bayesian generative models that combine observation, prediction, and action into a unified framework [51, 181]. Active Inference works effectively even with limited data, providing UAVs with adaptive decision-making capabilities in uncertain environments and enabling real-time mission execution.

Overall, this progressive evolution demonstrates how each approach addresses the weaknesses of the previous one, and ultimately, Active Inference emerges as a state-of-the-art, adaptable, and computationally efficient method for complex multi-agent problems, such as UAV swarm trajectory planning and MTSP.

Table 2.5 Comparison of different methods for trajectory planning review.

Method	**Examples**	**Scope of Use**	**Skill Requirement**
TA	Dijkstra, A*, RRT, DWA, Dubin's Path	Known or static environments, pathfinding, and trajectory planning for individual UAVs or fixed-wing UAVs	Beginner to Intermediate
BIA	PIO, SSA, ABC, ACO, PSO, GA	Complex or large search spaces, swarm coordination, and optimisation in UAV swarms	Intermediate
AI-A	MARL, Deep RL, Q-learning, Actor–Critic, Imitation Learning, Active Inference	Dynamic, stochastic, and uncertain environments, multi-agent cooperation, learning from experience	Advanced

Table 2.5 illustrates when and where different approaches are used to solve complex problems such as UAV trajectory planning and MTSP. While TAs are simple and computationally efficient, they are limited to static situations. BIAs are helpful for more complex and large-scale optimisation; however, they require parameter tuning and computational resources. AI-based approaches, particularly DRL and Active Inference, are most promising

in high-dimensional and uncertain scenarios; however, they require specialised expertise, advanced computational setups, and often large training datasets.

2.6.6 Hybrid Methods

Efficient trajectory design for UAV networks can be achieved using hybrid techniques such as 2-OPT, GA, and active inference. Initially, the 2-OPT algorithm is employed to generate offline training examples, where UAV paths are optimised based on minimum distance and time. This data is then used to train a world model, enabling the UAV to self-supervise its environment and select an online policy through active inference [98]. Another study proposed a GA-based hybrid approach to generate repulsion forces in UAV swarm paths, thereby reducing collisions, overlaps, and interference among UAVs while producing optimal paths under the challenges of MTSP [19]. The data generated by 2-OPT is fed into an active inference model, allowing UAVs to analyse online situations, adapt their policies accordingly, and perform fast, stable, and reliable path planning. This hybrid framework enables UAVs not only to learn from offline training but also to make optimal decisions in real time through online active inference, resulting in significant improvements in network performance, overall capacity, and the sustainability of route planning [97].

2.7 Online and Offline Training and Testing: In the Context of UAV Swarms

Offline Training

The UAV swarm trajectory planning model is trained on previously collected data (trajectory sets, mission requirements, obstacle maps). This process is often conducted in a simulator or controlled environment to enable UAVs to learn effective policies before they are deployed on a mission. Once the model has completed training, it is deployed in the field [71, 97].

Online Training

The model receives new observations in real-time and continuously updates its policy. This method is essential in dynamic and uncertain environments because it enables UAVs to make adaptive decisions during the mission [240, 97].

Table 2.6 illustrates a comparative overview of key aspects of offline and online learning in UAV swarms. The comparison reveals that offline training offers a safer and less complex approach, while also having room for improvement in terms of flexibility. In contrast, online training provides real-time adaptation, albeit at the expense of requiring more computational resources and increasing the risk of field errors.

Table 2.6 Comparative aspects of offline and online learning in UAV swarm.

Aspects	**Offline training**	**Online training**
Learning time	Before mission	During mission
Data source	Pre-existing data	Real-time field data
Computational complexity	Low	High
Model flexibility	Limited	High (adaptive)
Risk	Low (safe environment)	High (potential for error in field)

2.7.1 Integration of Offline Training with Online Testing

An effective strategy for UAV swarm missions is to utilise offline training for initial learning, followed by validation and fine-tuning of the model in the field through online testing.

Offline Phase: BIA's Generated Data with Supervised / Unsupervised Learning

In the offline phase, the goal is to train an AI policy using data generated from BIAs (such as GA, PSO, or ACO) to learn expert-level performance [60]. To do this, a dataset definition is first defined, which consists of states and their corresponding actions:

$$D = \{(s_i, a_i)\}_{i=1}^{N}, \tag{2.25}$$

where D is the trajectory plan generated by the BIAs, and each pair (s_i, a_i) represents a particular state and its corresponding expert action.

Based on this dataset, a loss function is defined so that the AI policy $\pi_\theta(s)$ can accurately replicate the expert's actions. The following optimisation problem is solved to minimise this loss:

$$\theta^* = \arg\min_{\theta} \sum_{(s,a)\in D} \|a - \pi_\theta(s)\|^2, \tag{2.26}$$

where:

- D: dataset generated by algorithms such as GA, PSO, or ACO;

- s: state of the environment (e.g., location of the UAV or remaining targets);
- a: action (trajectory segment or assignment) recommended by the BIAs;
- $\pi_\theta(s)$: AI-based policy that is learning to predict actions for these states.

Through this process, the AI policy can learn from the expert algorithm's decisions to enhance path planning and task allocation in the MTSP, enabling effective and autonomous decision-making without requiring expert assistance in the future.

Online phase (AI-based Fine-tuning)

During the online phase, the model learns from the environment in real-time to further refine the policy it has previously learned. This process is achieved through RL-based fine-tuning, where the UAV updates its policy based on its observations and the rewards it receives, thereby improving mission performance [10]. The following policy update equation is used for this purpose:

$$\theta_{t+1} = \theta_t + \alpha \cdot \nabla_\theta \log \pi_\theta(a_t|s_t) \cdot r_t, \tag{2.27}$$

where:

- θ: parameters of the AI model, which are updated during the learning process;
- s_t, a_t: current state and currently selected action;
- r_t: reward received after the action, which reflects the effectiveness of the action;
- α: learning rate, which determines how much impact each update will have

Equation 2.27 ensures that the UAV updates its policy toward actions with higher expected rewards.

In MTSP scenarios, this update mechanism enables real-time path and speed optimisation, rapid adaptation to new targets, and improved inter-UAV coordination, enhancing overall mission success.

In this approach, BIAs (e.g., GA, PSO, ACO) are employed to generate initial datasets and trajectories, which subsequently serve as training inputs for AI-based models. This dual strategy not only provides UAV swarms with a robust initial policy but also enables them to perform adaptive decision-making in real-time, significantly increasing both mission success and safety [60, 10, 61].

2.8 Decision-Making and Collision Avoidance in UAV Swarms

2.8.1 Decision-Making in Swarms

Decision-making is a fundamental challenge in UAV swarm systems, as each UAV must not only focus on its mission (such as task execution or trajectory following) but also make real-time decisions while cooperating with other UAVs. The accuracy of these decisions is critical for mission success, collision avoidance, efficient energy use, and overall system stability [165, 239].

Decision-making is typically described at two levels: Local Decision-Making, where each UAV makes decisions based on its local information (such as sensor data and the positions of nearby UAVs). Collective Decision-Making, where UAVs share data and act according to a global strategy [177].

2.8.2 Online and Offline Decision Making

Offline decision-making: In offline decision making, UAVs rely on pre-trained policies or role-based models, which are often trained on simulations or historical data. This approach is computationally lightweight and suitable for predictable missions (such as mapping or fixed survey paths) [184].

Online decision-making: Online decision-making is more dynamic, where UAVs continuously observe the environment, share information, and make decisions in real-time based on the current situation (such as sudden obstacles, changing weather conditions, or a new mission target). This approach makes UAV swarms more adaptive and resilient, but it requires more computational power and a robust communication structure [239].

Modern research is moving in the direction of using both methods in a hybrid manner, that is, first providing UAVs with a basic decision policy through offline learning and then continuously improving it through online decision-making during the mission [61].

Table 2.7 illustrates a comparison of offline and online decision-making in UAV swarms, showing that offline methods have low computational demands and rely on pre-trained policies. In contrast, online decision-making offers greater adaptability and flexibility in real-time, but requires more computational resources.

Table 2.7 Comparison of offline and online decision making in UAV swarms.

Aspect	**Offline decision making**	**Online decision making**
Decision time during mission	Pre-trained	Real-time
Adaptation	Static	Dynamic
Example	Pre-trained policy, static path	Real-time obstacle avoidance, task redistribution
Computational demand	Low	High

2.8.3 A Challenge in Decision-Making: Collision Avoidance

When multiple UAVs fly together on close or shared paths, the risk of collision increases. This is a fundamental challenge for UAV swarms, as a minor collision can not only damage one UAV but also fail the entire mission. Therefore, collision avoidance strategies are considered an integral part of decision-making. Modern research has shown that various approaches are used to improve collision avoidance in UAV swarms, including geometric, potential field-based, optimisation-driven and AI-assisted methods [162, 179].

Collision avoidance: Techniques by which UAVs avoid each other or obstacles to maintain mission safety. Decision-making in UAV swarms is not limited to path selection, but is a continuous, informative and protective process, involving real-time perception and mutual coordination. Especially in dynamic and uncertain environments, online decision-making and collision avoidance are inseparable [169].

2.9 Filters — Role in Goal-Directed Trajectory Design

directed trajectory designing in UAV swarms requires that each UAV accurately estimates its position, velocity, and direction so that it can choose the optimal path to reach the target [190, 97]. Since real observations are often noisy and uncertain, Filtering Techniques such as KF [74, 127] and PF [48, 128] are used to improve the observations and achieve more stable trajectories.

2.9.1 Kalman Filter (KF)

KF is a probabilistic estimation method used to find the hidden states of a system. It assumes that the system is linear and the noise is Gaussian in nature [63]. It consists of two stages:

Prediction and Update.

Suppose the state of the UAV at time k is described as follows:

$$x_k = F_{k-1}x_{k-1} + B_{k-1}u_{k-1} + w_{k-1} \tag{2.28}$$

$$z_k = H_k x_k + v_k \tag{2.29}$$

Where: - x_k : State of the UAV (e.g. position, velocity, angle) - F_{k-1} : State Transition Matrix - $B_{k-1}u_{k-1}$: Control Input - w_{k-1} : Process Noise - z_k : Measurement - H_k : Observation Matrix - v_k : Measurement Noise

Prediction Step:

$$\hat{x}_{k|k-1} = F_{k-1}\hat{x}_{k-1|k-1} + B_{k-1}u_{k-1} \tag{2.30}$$

$$P_{k|k-1} = F_{k-1}P_{k-1|k-1}F_{k-1}^T + Q_{k-1} \tag{2.31}$$

Update Step:

$$K_k = P_{k|k-1}H_k^T (H_k P_{k|k-1} H_k^T + R_k)^{-1} \tag{2.32}$$

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k(z_k - H_k\hat{x}_{k|k-1}) \tag{2.33}$$

$$P_{k|k} = (I - K_k H_k)P_{k|k-1} \tag{2.34}$$

These equations enable continuous prediction and correction of the UAV trajectory. While the UAV is moving towards the target, the KF updates its estimate every moment, which makes the trajectory smooth and accurate. The following are the main types of KFs used in Goal-Directed Trajectory Designing of UAVs.

Extended Kalman Filter (EKF): EKF is designed for nonlinear systems. Since the KF only works on linear systems, the EKF linearizes the nonlinear functions of the system using a Taylor Series Approximation [163]. The system is expressed as:

$$x_k = f(x_{k-1}, u_{k-1}) + w_{k-1}, \quad z_k = h(x_k) + v_k \tag{2.35}$$

Where $f(\cdot)$ and $h(\cdot)$ are nonlinear functions. The EKF linearizes these functions by a first-order Taylor series and updates them using the Jacobian matrix. In UAV swarms, this filter is suitable for attitude estimation, sensor fusion, and GPS/INS integration. However, if the system is highly nonlinear, the error in the EKF can increase.

Unscented Kalman Filter (UKF): UKF is introduced to overcome the limitations of EKF. This filter better represents nonlinear distributions rather than linearizing the system. In UKF, "sigma points" are selected that represent the spread of the state distribution. These points are passed through a nonlinear function to obtain a more accurate estimate [196, 81]. Mathematically, it updates the mean and covariance of the estimates in a more realistic way:

$$\hat{x}_k = \sum_{i=1}^{L} W_m^{(i)} \chi_k^{(i)}, \quad P_k = \sum_{i=1}^{L} W_c^{(i)} (\chi_k^{(i)} - \hat{x}_k)(\chi_k^{(i)} - \hat{x}_k)^T + Q_k \tag{2.36}$$

Where $\chi_k^{(i)}$ are the sigma points and $W_m^{(i)}$, $W_c^{(i)}$ are the weights. UKF for UAVs provides more accurate sensor fusion, attitude estimation, and excellent results in flight control, especially when the system is nonlinear but continuous.

Ensemble Kalman Filter (EnKF): EnKF is designed for large and complex systems with high state dimension. It uses a statistical approach based on Monte Carlo simulations in which particles (samples) or "ensemble members" are generated for the state of the system [77, 101]. Each member represents a possible state, and the average estimate of all of them is obtained as follows:

$$\bar{x}_k = \frac{1}{N} \sum_{i=1}^{N} x_k^{(i)} \tag{2.37}$$

$$P_k = \frac{1}{N-1} \sum_{i=1}^{N} (x_k^{(i)} - \bar{x}_k)(x_k^{(i)} - \bar{x}_k)^T \tag{2.38}$$

The advantage of EnKF is that it remains computationally feasible even in large-scale nonlinear systems, and reduces noise sensitivity. In UAV swarms, this filter is used for cooperative trajectory estimation and dynamic environment adaptation.

All types play a central role in Goal-Directed Trajectory Designing of UAVs, where they enable accurate and smooth flight towards the target by estimating the true state of the system.

2.9.2 Particle Filter (PF)

When the system is nonlinear or the noise is non-Gaussian, the PF is used. It is based on Monte Carlo simulation and describes the state of the system in terms of a number of particles, where each particle represents a possible state [48].

The general equation of the system is as follows:

$$x_k = f(x_{k-1}, u_{k-1}) + w_{k-1} \tag{2.39}$$

$$z_k = h(x_k) + v_k \tag{2.40}$$

Here $f(\cdot)$ and $h(\cdot)$ are nonlinear functions.

For each particle i, the following steps are performed at time k:

Prediction:

$$x_k^{(i)} \sim p(x_k|x_{k-1}^{(i)}, u_{k-1}) \tag{2.41}$$

Weighting:

$$w_k^{(i)} \propto w_{k-1}^{(i)}\, p(z_k|x_k^{(i)}) \tag{2.42}$$

Normalization:

$$w_k^{(i)} = \frac{w_k^{(i)}}{\sum_{j=1}^{N} w_k^{(j)}} \tag{2.43}$$

Resampling: The particles with lower weights are removed and the more suitable particles are retained to increase the accuracy of the estimation.

Trajectory Estimation: Finally, the estimated state of the UAV is obtained as follows:

$$\hat{x}_k = \sum_{i=1}^{N} w_k^{(i)} x_k^{(i)} \tag{2.44}$$

This equation shows that the PF evaluates multiple paths based on probabilities and selects the most suitable path that is, it provides the ability to accurately design the UAV's goal-directed trajectory even in uncertain environments. Below are some of the main types of PFs that are commonly used in the context of Goal-Directed Trajectory Designing and UAV swarms.

Auxiliary Particle Filter (APF): APF aims to improve the performance of the SIR filter. This filter uses information from future observations during resampling to prioritize more suitable particles for the next step [158, 206]. Mathematically, it introduces an additional weighting function:

$$w_k^{(i)} \propto w_{k-1}^{(i)} p(z_k|\hat{x}_k^{(i)}) q(\hat{x}_k^{(i)}|x_{k-1}^{(i)}, z_k) \tag{2.45}$$

Where $q(\cdot)$ is a proposed distribution. This method reduces the influence of less reliable particles and increases the accuracy of resampling.

Rao-Blackwellized Particle Filter (RBPF): RBPF is a hybrid method that combines both KF and PF [22]. In this, some states of the system, which are linear and Gaussian in nature, are estimated by the KF, while the PF is used for nonlinear and uncertain states. In this way, the overall accuracy and efficiency of the estimation are improved. The basic equation of RBPF is as follows:

$$p(x_k|z_{1:k}) = \int p(x_k|x_k^{(i)}) p(x_k^{(i)}|z_{1:k}) dx_k^{(i)} \tag{2.46}$$

This method is particularly useful in complex problems such as Simultaneous Localization and Mapping (SLAM) and Cooperative UAV Path Estimation.

Gaussian Particle Filter (GPF): In GPF, each particle is not just a point but represents a complete Gaussian distribution [119]. That is, each particle represents both an estimate and its covariance:

$$x_k^{(i)} \sim \mathcal{N}(\mu_k^{(i)}, \Sigma_k^{(i)}) \tag{2.47}$$

This method produces smoothness in the estimates and reduces the excessive variation of the weights. GPF is particularly suitable for UAV routing systems where there is minor noise in the observations and the accuracy of the estimate is important.

Markov Chain Monte Carlo Particle Filter (MCMC-PF): MCMC-PF select particles from a better predicted distribution [107]. This method is effective when the observation model is complex or resampling reduces the diversity of particles. MCMC filters can be used for cooperative path optimization in UAV swarms, where each UAV updates its state based on the estimates of the others.

All of these filters can be used for Goal-Directed Trajectory Designing in UAV swarms in such a way that the system analyzes the feasibility of possible paths, continually updates the prediction and observation, and progresses towards the goal with greater confidence and accuracy.

Both filters play a fundamental role in routing in UAV swarms. The KF provides fast and accurate estimation in linear systems, while the PF provides a more flexible and effective solution in nonlinear, uncertain, and variable environments. These filters reduce the gap between prediction and observation by continuously updating the UAV's beliefs within an Active Inference framework, allowing the autonomous system to move towards the target with greater confidence and accuracy.

2.9.3 Modern and Scientific Methods for Collision Avoidance

Several approaches have been developed for collision avoidance in UAV swarms, which can be categorised into the following groups.

Geometric Methods:

These methods are based on the geometry of the velocity and position of UAVs. For example, in the velocity obstacle method (VOM), each UAV predicts its future position based on the current position and velocity of other UAVs, and adjusts its velocity to avoid potential

Table 2.8 Comparative summary of KF and PF types for UAV trajectory designing

Filter Type	**Key Features**	**Advantages**	**Applications in UAV**
EKF	Linearizes nonlinear systems via Taylor series.	Accurate estimation in constrained nonlinear environments.	UAV attitude estimation, localization.
UKF	Better representation of nonlinear distributions through sigma points.	Higher accuracy than EKF; lower linearity error.	Sensor fusion, GPS/INS integration.
EnKF	Based on Monte Carlo simulation; suitable for large systems.	Better stability in complex systems.	Supported routing in multiple UAVs.
APF	Adds future observations to the weighting.	More accurate and stable estimation.	Complex navigation and cooperative path planning.
RBPF	Combination of KF + PF.	More accurate estimation in hybrid systems.	UAV SLAM (Simultaneous Localization and Mapping).
GPF	Represents each particle as a Gaussian distribution.	Smooth and noise-resilient estimation.	UAV flight path smoothing and uncertainty modeling.
MCMC-PF	Uses Markov Chain Monte Carlo Sampling.	Maintains particle diversity; suitable for complex environments.	Supported UAV trajectory optimization.

collisions [115]. The basic concept in this method is to define a velocity obstacle set, which is the set of all velocity vectors that could lead to a collision in the future. This set can be expressed mathematically as:

$$VO_{i|j} = \left\{ v_i \;\middle|\; \exists t > 0 : \; p_i + v_i t = p_j + v_j t \right\}, \tag{2.48}$$

where:

- $VO_{i|j}$ is the set of all possible velocities of UAV i that can cause a collision with UAV j.
- p_i, p_j are the current positions of UAV i and UAV j.
- v_i, v_j are the current velocities of UAV i and UAV j.
- t is the time in the future when the collision can occur.

If v_i is part of this set, UAV i can collide with UAV j in the future. In this case, UAV i should change its velocity and adopt a safe alternative vector. In the context of MTSP, equation 2.48 enables UAVs to not only optimise their routes in real time but also complete missions at a safe distance from each other, regardless of the proximity of their routes. This helps to avoid collisions, reduce mission completion time, and improve team coordination.

Force Field Approaches

Potential field-based approaches to UAV navigation are based on the concept that the mission target and obstacles in the environment produce attractive and repulsive forces, respectively. These forces can be expressed mathematically as a total force function, which determines the direction and magnitude of the UAV's motion.

$$F = F_{attract} + F_{repel}, \tag{2.49}$$

where:

- $F_{attract}$: force that attracts the UAV towards the target;
- F_{repel}: force that repels the UAV from the obstacles.

These two forces together enable the UAV to take a smooth and safe path, where the attraction force encourages it to reach the target while the repulsion force ensures collision avoidance [116, 37, 113].

In the context of MTSP, this function not only provides UAVs with an effective path to the target but also helps to avoid collisions and reduce mission completion time in multi-UAV operations.

Optimization-Based Methods

Optimisation-based collision avoidance approaches are based on the principle that each UAV should choose its path in such a way that the overall mission cost is minimised, while also meeting the requirements for collision avoidance. For this, a cost function is defined that incorporates both mission performance and safety conditions.

$$\min_{x} J(x) \quad \text{s.t. } |x_i - x_j| \geq d_{safe}, \ \forall i \neq j, \tag{2.50}$$

where:

- $J(x)$: overall mission cost (e.g., time, distance, or energy);

- x_i, x_j: positions of UAV i and UAV j;
- d_{safe}: minimum safe distance that must be maintained between UAVs.

This optimisation problem ensures that each UAV updates its path in a way that not only completes the mission at the lowest cost but also stays at a safe distance from other UAVs.

In the context of MTSP, this method is particularly effective in multi-objective scenarios, as UAVs can simultaneously improve both mission efficiency and flight safety [5].

Lennard-Jones Potential:

The Lennard-Jones Potential is a physical model that describes the balance of attractive and repulsive forces between two UAVs [138]. This concept is utilised in collision avoidance algorithms to prevent UAVs from getting too close or too far apart. The following potential function mathematically represents this model:

$$U(d) = \varepsilon \left[\left(\frac{\sigma}{d} \right)^{12} - 2 \left(\frac{\sigma}{r} \right)^{6} \right], \tag{2.51}$$

where:

- d: distance between the two UAVs;
- ε: parameter controlling the magnitude of the potential;
- σ: distance at which the potential is at its minimum value.

The Lennard-Jones model generates strong repulsion at close range and weak attraction at intermediate range, allowing UAVs to maintain a safe distance and avoid collisions [214].

In the context of MTSP, the Lennard-Jones potential enables UAVs to adopt a balanced behaviour, efficiently completing their paths while maintaining coordination within the swarm, especially in narrow or complex mission areas.

Harmonic Potential

Harmonic potential is an effective mathematical model that imposes a penalty for deviations from the desired distance between two UAVs or between a UAV and a target. The basic concept relies on a quadratic function, where energy or potential increases with deviation [126]. This function ensures that the UAVs remain within the desired distance d_0. Mathematically, it is expressed as:

$$U(d) = \frac{1}{2} k (d - d_0)^2, \tag{2.52}$$

where:

- d: current distance;
- d_0: desired or target distance;
- k: spring constant, which controls the magnitude of the correction.

This provides a soft corrective mechanism, as minor deviations incur a small penalty, while large deviations incur a significantly larger penalty.

In the context of MTSP, the harmonic potential is beneficial for formation-based missions, where UAVs must reach targets while maintaining a certain distance. The harmonic potential method not only ensures collision avoidance but also improves swarm coordination and mission performance.

Gaussian Repulsion Force

The Gaussian repulsion force is designed to generate a repulsive force as the distance decreases. Still, this force increases or decreases smoothly so that there are no sudden changes in the movement. This has the advantage that the movement of the UAVs remains more natural and stable, especially when the swarm formation is dense [135, 186]. Mathematically, this potential function is expressed as:

$$U(d) = A \cdot \exp\left(-\frac{(d-\mu)^2}{2\sigma^2}\right), \tag{2.53}$$

where:

- d: current distance between the two UAVs;
- A: maximum amplitude of repulsion;
- μ: distance around which repulsion is most effective;
- σ: spread parameter, which determines the extent of the repulsion effect.

This Gaussian repulsion force method prevents sudden changes in motion or direction, allowing UAVs to move in a smooth and coordinated manner. In the context of MTSP, the Gaussian repulsion force protects UAVs from collisions in dense aerial scenarios, while also ensuring stable swarm alignment and improved mission performance.

Inverse Quadrature / Artificial Potential Field (APF)

In the APF method, the mission target generates an attractive force while obstacles generate a repulsive force. This repulsive force is designed to keep the UAV away from obstacles, thereby avoiding collisions. This model utilises a potential function based on the inverse square of the distance from the obstacle, which increases rapidly as the obstacle is approached. Mathematically, the repulsive potential can be described as:

$$U_{rep}(d) = \frac{1}{(d-d_0)^2}, \tag{2.54}$$

where:

- d: current distance between the UAV and the obstacle;
- d_0: safe or minimum allowed distance.

As d approaches d_0, the collision potential increases significantly, forcing the UAV to change direction and ensuring collision avoidance [90].

In the context of MTSP, the APF method enables UAVs to navigate towards the target while avoiding obstacles, thereby accelerating mission completion and maintaining swarm coordination.

Priority-based strategies

In certain scenarios, UAVs employ simple rule-based strategies to avoid collisions or manage air traffic. For example, a UAV may slow down or stop to let another UAV pass first. This strategy does not rely on complex mathematical models or heavy computational processing, making it a low-computational heuristic that is particularly effective in congested airspace [23].

In the context of MTSP, priority-based strategies enhance inter-UAV route coordination, minimise unnecessary interference, and reduce mission completion time. It is especially beneficial when airspace is limited or swarms have to operate nearby.

2.9.4 Challenges in Collision Avoidance Methods

Many of these approaches, geometric, force field, optimisation-based, and heuristic, offer distinct advantages; however, they also have some drawbacks.

- Geometric methods: These are mathematically simple and fast in real time. However, they can be less flexible in dynamic and uncertain environments, and their accuracy may suffer in complex scenarios.

- Force field methods: These provide smooth and safe paths. However, they can become stuck in local minima and are ineffective in environments with complex constraints.

- Optimisation-based methods: improve performance and safety simultaneously. Nevertheless, they have high computational cost and can be slow in large swarms or real-time applications.

- Heuristic methods: are simple, fast, and require fewer computational resources. However, they do not always provide the best solution and may fail in complex or unpredictable situations.

The current trend is towards developing hybrid systems that integrate physics-based and AI-driven collision avoidance techniques. This combination can significantly improve the reliability, adaptability, and mission performance of UAV swarms by combining the strengths of each method[169, 10, 82].

Table 2.9 illustrates various collision avoidance methods. Each method has distinct advantages and limitations, and hybrid approaches are often employed for more effective results in practical scenarios.

Table 2.9 Different collision avoidance methods in UAV swarms and their applications.

Method	Explanation	Role in UAV swarms
Geometric	Geometry-based analysis of velocity and path, such as velocity obstacle (VO) or reciprocal velocity obstacle (RVO).	Fast and computationally light; effective in low-density swarms and predictable environments [115].
Force field	Combination of attractive and repulsive forces; target pulls and obstacle pushes.	Generates intuitive and smooth paths, but can get stuck in local minima [116, 37].
Optimisation	Minimises the objective function with collision avoidance constraints.	Very effective in multi-UAV coordination, but computationally demanding [5].
Lennard–Jones	Physics-inspired potential that provides short-range repulsion and medium-range attraction.	Useful for formation flights and maintaining safe separation [214].
Harmonic potential	Quadratic potential that provides a penalty on deviation.	Helpful in information-keeping and smooth trajectory generation [126].
Gaussian repulsion	Gaussian-based repulsive field that increases in intensity with distance.	Provides soft yet strong repulsion and reduces sudden manoeuvres [135].
Inverted quadrant / APF	Classical artificial potential field model: target is attractive, and obstacles have repulsive potential.	Easy implementation; but the problem of local minima remains [90].
Waiting or yielding rules	Priority-based heuristics: UAV stops or slows down to let other UAVs go first.	Simple and effective in decentralized systems; basic safety measure in congested airspaces [23].

2.10 Challenges in UAV Swarm Trajectory Planning

Trajectory planning for UAV swarms is a complex, multifaceted problem with numerous challenges. The most fundamental challenge is to determine safe, energy-efficient, and collision-free trajectories in real-time in a dynamic and unpredictable environment. Collision avoidance and effective coordination among multiple UAVs, particularly in the presence of limited communication resources and latency, are key issues. In addition, the presence of obstacles, deceptive signals, and weather uncertainties also affects the accuracy of trajectory planning. A summary of the challenges is presented in Fig. 2.13.

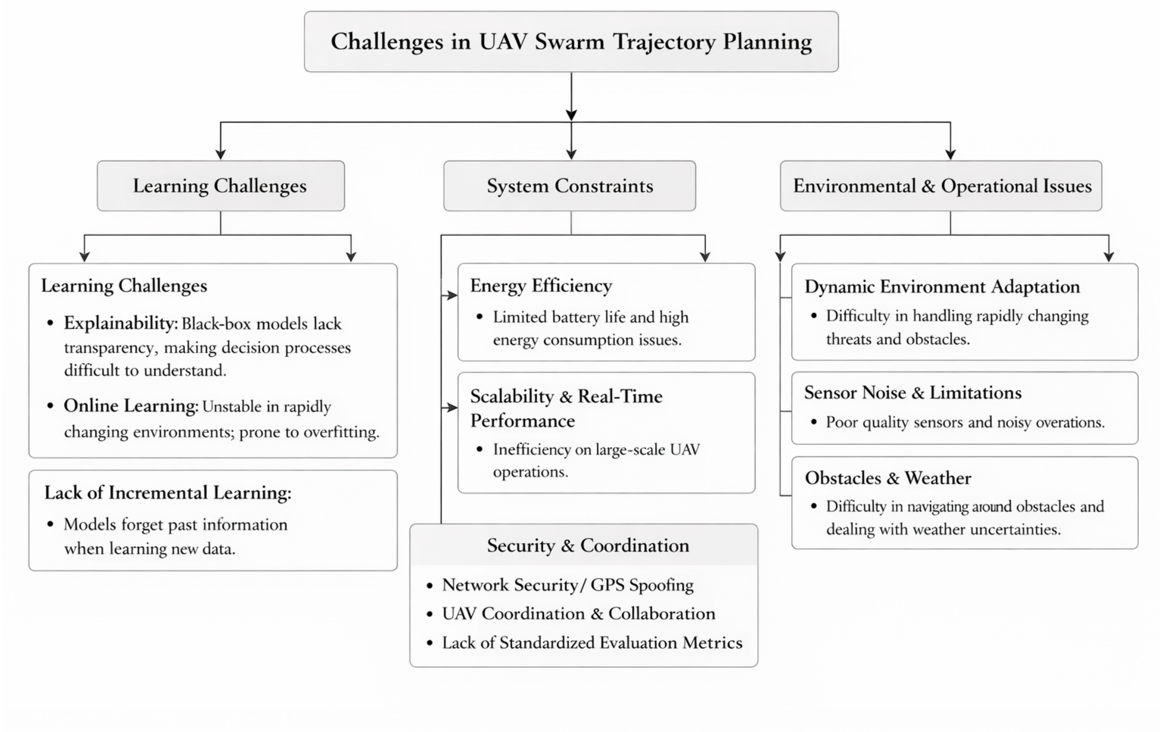


Fig. 2.13 Summary of the key challenges faced in UAV swarm trajectory planning.

There is still no internationally recognised uniform metric for trajectory planning, making it difficult to compare different models scientifically. This deficiency is hindering research progress.

Trajectory planning for UAV swarms is another multifaceted problem that involves a number of technical and practical tasks. The most fundamental problem is to determine safe, energy-efficient, and collision-free paths in real time in dynamic and non-environmental environments. Effective coordination and collision avoidance between UAVs becomes particularly challenging when more communication resources are limited or network options are available. Furthermore, planning constraints, such as deceptive signals and weather uncertainties, can affect the accuracy and reliability of the trajectory.

Modern algorithms, especially those based on machine learning and reinforcement learning, often operate as black-box designs whose full decision-making is not easily understood. Clarity and transparency of the adaptive process in a UAV swarm system are important for system reliability, validation, and safe testing. Similarly, online learning capability is essential for effective decision-making in a rapidly changing environment, but it is currently limited or non-existent, and protecting it from further phoning during online updates is a major issue.

Another important aspect is the lack of incremental learning, as many new data are learned while learning from previously acquired information. Such a mechanism to control this continuously improves the information it provides by advancing it. Energy efficiency is also an important aspect because UAVs have a limited power life, and changing algorithms or frequent Rahul changes can change energy consumption.

Furthermore, UAV systems are network-based, sensitive to GPS spoofing, data points, or communication connectivity, which can affect the trajectory planning process. Similarly, ensuring effective cooperation and distributed decision-making among multiple UAVs is also a constant, especially when each UAV is operating autonomously and there is no central control system. Moreover, the current conditions are designed for a typical jam or semi-dynamic environment, while in the real world, there is a relaxed, and constantly changing preferences, which requires fast and accurate adaptation.

In addition to the scalability and real-time performance of algorithms, a major issue is that algorithms that are effective in small systems often do not prove effective over a large number of UAVs. The limitation of the sensors and the observational noise can also affect decision-making, as UAVs often operate on low-quality or noisy sensor data. Finally, it has been scientifically difficult to develop standardized and globally recognized evaluation metrics for trajectory planning performance, which is a key issue for progress in its research.

2.11 Future Research Directions

Given the current challenges in UAV swarm trajectory planning, future research should prioritise technical directions that deliver effective and flexible solutions while incorporating explainability, online learning, and principles inspired by biological motion. A summary of the future research directions is depicted in Fig. 2.14.

The lack of training datasets poses a significant challenge, particularly for RL or Active Inference models. To address this issue, diverse and realistic synthetic trajectories can be generated using BIAs such as GA, PSO, or ACO. This data can not only be used for pre-training but can also be used to power other models through transfer learning.

Active Inference is a modern and emerging framework based on Bayesian brain theory. This model unifies perception, decision-making, and learning simultaneously. In the context of UAVs, Active Inference not only enables decision-making under uncertain conditions but also maintains the internal transparency of the system. The most prominent feature of

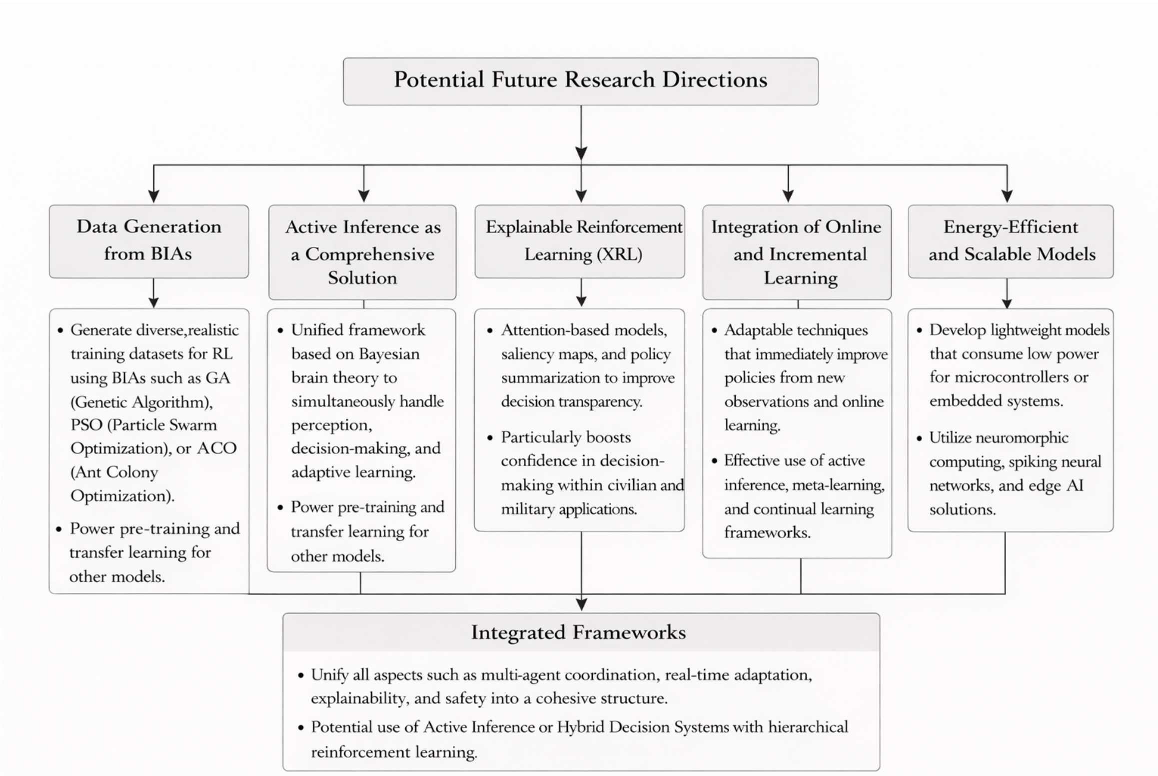


Fig. 2.14 Summary of potential future research trends and key research directions for UAV swarm trajectory planning.

this model is that it self-learns policies to minimise prediction error, making it particularly suitable for dynamic and partially observable environments.

Although traditional RL models possess strong learning abilities, their decisions are often opaque and difficult to understand. Explainable RL techniques such as attention-based models, saliency maps, or policy summarisation will be crucial for the reliability and human-in-the-loop validation of UAV systems in the future. It will enhance confidence in decision-making, particularly in both civilian and military applications.

TAs often rely on offline training, which can be ineffective in practical applications where the environment changes over time. In the future, lightweight and real-time adaptable models are needed that can immediately learn from new observations and improve their policy based on online learning. Approaches such as active inference, meta-learning and continual learning frameworks can be effective in this context.

Given the limited battery and computational resources in UAVs, there is a need for computationally lightweight models that consume low power and can efficiently run on microcontrollers or embedded systems. Neuromorphic computing, spiking neural networks, and edge AI solutions can play a crucial role in this direction.

A comprehensive framework is needed in the future that integrates all aspects, such as multi-agent coordination, real-time adaptation, explainability, and safety into a unified structure. For this purpose, approaches such as Active Inference or Hybrid Decision Systems with hierarchical reinforcement learning are promising.

2.12 Summary

The findings of this chapter highlights that hybrid frameworks, which combine the reliability and global search capability of BIAs with the adaptability and learning efficiency of AI, represent a promising direction for future UAV swarm trajectory planning. These hybrid approaches establish an effective exploration–exploitation balance in multi-agent missions and have the potential to address fundamental challenges such as collision avoidance, scalability, and mission performance.

Finally, this chapter provides a strong theoretical foundation for UAV swarm trajectory design. In the subsequent chapters, particularly Chapter 3, this theoretical foundation will be applied to develop a World Model based on the Genetic Algorithm–Repulsion Force (GA–RF) framework, which will play a central role in advancing UAV swarm trajectory design. This chapter can therefore be regarded as the first stage of a hybrid approach, establishing the basis for generating World Model Knowledge through an empirical, expert-driven optimization process.

Chapter 3

UAV Swarm Trajectory Design for Wireless Networks Using Genetic Algorithm-Driven Repulsion Forces

Based on the results obtained in Chapter 2 and the effective methods highlighted, this chapter presents a new and improved method based on the integration of GA and RF. The proposed method introduces dynamic repulsive forces between UAVs that not only maintain an appropriate distance between them but also ensure better coordination and orderly movement within the swarm. Thus, this approach effectively alleviates some limitations of traditional MTSP-based models, such as overlaps between UAVs, unbalanced paths, and potential collision issues, and improves the overall energy consumption and performance of UAV swarms.

This study compares the performance of the proposed method with several well-known optimization algorithms, including MTSP-GA, Particle Swarm Optimization (PSO), 2-OPT, Ant Colony Optimization (ACO), and Simulated Annealing (SA) to evaluate its performance. Various simulation experiments and performance index analyses show that the proposed method effectively reduces the total travel distance and time, limits the possibility of collisions and path overlaps between UAVs, and improves the overall organizational and operational efficiency of the swarm.

Overall, the obtained results demonstrate that the proposed GA-RF-based method provides more effective and stable performance than traditional strategies. At the same time, the method offers a scalable, reliable, and practical solution for large-scale UAV swarms, which can be effectively used in future advanced autonomous UAV-based communication and surveillance systems.

3.1 Introduction

The design of trajectories in autonomous control systems such as UAV swarms is crucial for their successful operation and for achieving their objectives, especially when dealing with a large number of UAVs [176]. The trajectory design problem entails selecting the best flight path for UAV swarms from a starting point to a target point while meeting specific conditions. It involves establishing a flight path that can quickly reach the target area at minimal cost. Trajectory design for UAV swarms is considered an NP-hard problem, and several solutions have been proposed to address this challenge. Typically, three steps are employed to solve this issue. The first step involves creating a grid search map with targets and environmental information and reformulating the trajectory design task as a map search problem. The second step involves updating the search map using criteria that incorporate time-dependent characteristics into the UAV swarm's search process. Finally, the enhanced search map is used to determine optimal or near-optimal paths as quickly as possible to locate the target [154, 57].

Two main categories of search algorithms are used in trajectory planning to find the best path: traditional optimization algorithms and metaheuristic algorithms. Traditional optimization methods include the Newton's method [189, 99], the gradient descent method [168], the interior point method [136], and linear programming [144]. These traditional algorithms are well-suited for handling large-scale problems. However, they often struggle with getting stuck in local optima, making the optimisation results highly dependent on the initial values. Numerous novel metaheuristic algorithms have been developed in recent years to address this issue. These algorithms draw inspiration from various biological behaviours and natural phenomena. Some of them are named after notable individuals or entities, such as Simulated Annealing (SA) [237, 122], Tunicate Swarm Algorithm (TSA) [131], Aquila Optimization (AO) [1], Harris Hawks Optimization (HHO) [49], Grey Wolf Optimizer (GWO) [156], Differential Evolution (DE) [17], PSO [54], and Genetic Algorithm (GA) [198, 94, 217]. These bioinspired algorithms are primarily used for optimization problems related to task allocation and target identification in UAV swarms. However, a major limitation of these approaches is their high time complexity at each time step, which hinders them from converging to a reasonable approximation.

In addressing this issue, the work in [241] initially proposed a GA-based strategy to optimize the MTSP. However, this approach has a limitation as it focuses on multi-UAV path planning without considering the necessary collective dynamic coordination or communication for managing UAV swarms. In contrast, another researcher proposed a multi-UAV concept using the ACO algorithm. ACO is primarily effective for single-objective optimization and requires significant modifications to handle multi-objective problems effectively

[124]. Similarly, another study utilized the PSO algorithm to optimize the distance of a UAV swarm. Still, PSO may be suitable for specific problems rather than global ones, particularly in complex multidisciplinary environments [185]. Additionally, a multi-UAV task planning strategy using improved simulated annealing and GA are proposed to optimize the size of a single UAV [35]. Furthermore, an attracting trajectory design solution based on active inference has been presented in [97, 98]. However, this approach is tailored for a single UAV, necessitating fundamental steps for its adaptation to accommodate multiple UAVs collaborating in a swarm.

Several studies have focused on multi-objective optimization design to achieve the dual goals of minimizing route length and drone flight time upon mission completion [172]. For instance, one study introduced a multi-objective optimization model for UAV swarms, aiming to reduce path length and time spent in the air while enhancing genetic and crossover operators within an adaptive GA [26]. Another study proposed a model that combines MTSP and a two-dimensional in-plane point circle coverage model to address UAV identification [223]. This study utilized the AC and Monte Carlo algorithms to find the optimal path in the minimum amount of time while also addressing symmetric and asymmetric solutions to the MTSP. However, a closed path challenge remains to be resolved [210].

Optimization techniques, such as GA, are used to control UAV swarms in a 3D environment to manage UAV formations effectively [129]. A new method for joint optimal routing and trajectory optimization for UAVs is proposed in [193]. This approach extends the traveling salesman problem by considering operational constraints arising from the motion characteristics of drones. Another strategy incorporates a GA to avoid manipulators by determining the control and generating a repulsive force based on distance calculations between the robot's links and obstacles. This method improves real-time collision avoidance capability by adjusting the safety parameters according to the severity of potential collisions, thus adapting to the dynamic environment [62]. Additionally, a new method has been developed to improve aircraft trajectories in uncertain environments. This method solves the local minimization problems and improves the overall quality using Model-Based Control (MPC). The algorithm optimizes the pilot's ability to follow a planned trajectory, avoid fixed obstacles, and uses simulation and real-world testing [68].

In summary, while some researchers have adequately addressed optimization challenges in UAVs, the specific hurdles associated with UAV swarms have received limited attention. Certain studies have concentrated on optimizing distance for individual or multiple UAVs rather than for swarms as a whole. Although solutions for UAV swarms have been proposed, critical aspects such as collision avoidance have often been neglected. Even when collision avoidance is considered, issues such as multiple UAVs targeting closely situated points or trajectory interference are not sufficiently resolved. Furthermore, in cases where these issues are partially addressed, the optimization of swarm efficiency is still deficient.

In light of the above discussion, this study proposes a method for designing UAV swarm trajectories for wireless networks using an improved GA with RF. The method aims to minimize both distance and time, optimize the size of the UAV swarm, avoid interference in trajectories, prevent overlap in close proximity of multiple towns, and avoid collisions between UAVs.

The primary contributions of proposed work are summarized as follows:

- Developed a method for optimizing the trajectory of a swarm of UAVs to minimize time and distance. This method, called GA-RF, is based on a modified MTSP-GA that considers Repulsion Force (RF). GA-RF minimizes interference and overlap between UAVs, thereby reducing the risk of collisions and efficiently serving the designated areas.
- Optimize these forces using GA-RF to ensure optimal spacing between UAVs in a swarm, especially when two towns are close to each other. This RF mechanism guarantees that both towns are served by a single UAV, thereby optimizing the overall travel distance.
- Conducted extensive analysis to determine the optimal size of the UAV swarm needed to serve each respective town assigned to each UAV in the swarm.
- The approach is improved by utilizing a 2D environmental model and a comprehensive distance and time evaluation function to optimize UAV swarm coverage. Simulation results are examined to assess the algorithm's effectiveness and feasibility in planning UAV swarm trajectories within the 2D model, ultimately enhancing swarm performance in UAV-assisted wireless scenarios.

The chapter is organized as follows: Section II introduces the system model. Section III details the proposed GA-RF based framework for trajectory design. Section IV discusses the results and provides an analysis. Lastly, Section V includes the conclusion and outlines future work.

3.2 System Model

Consider multiple UAVs, depicted in Fig. 3.1, where numerous UAVs form a swarm and act as an Aerial Base Station (ABS) to provide uplink data service to randomly distributed

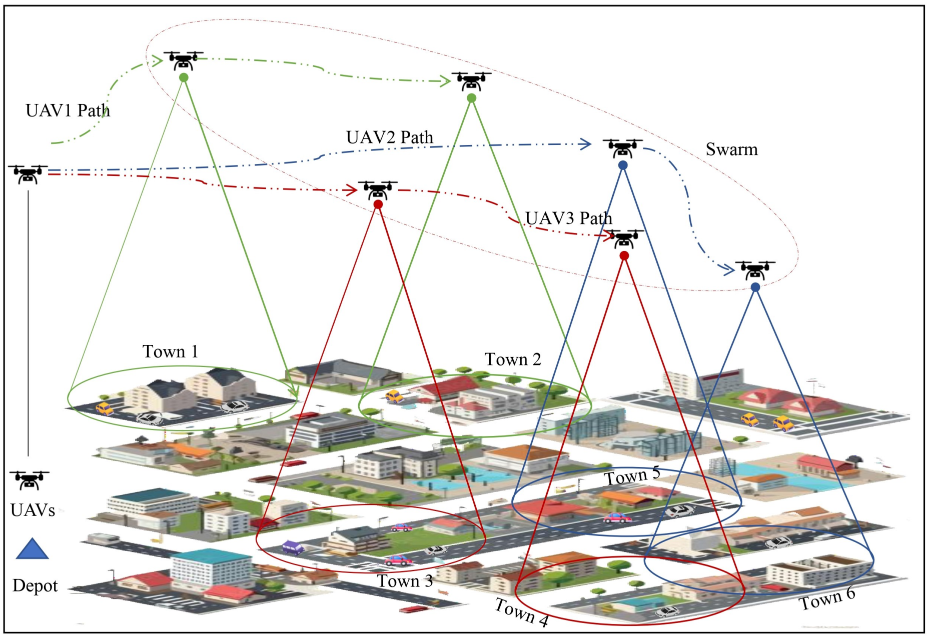


Fig. 3.1 Illustration of system model.

Ground Users (GUs) in a given geographical area. The coordinates C of each GU are given by $C_n = [x_n, y_n]$. The GUs are presumed to be divided into N distinct groups, each representing a specific town or hotspot T. The set of towns available is represented as $\mathscr{T} \triangleq \{T_n = T_1, T_2, ..., T_N\}$. These different groups will be further subdivided into UAVs. The objective of the UAV swarm is to travel from its starting point to the towns with significant demands for data services and then return to its original location in the minimum time. The UAV's starting and ending positions are represented by the coordinates $p_0 = [x_0, y_0]$.

Study consider a UAV swarm consisting of m UAVs ($m \geq 2$), denoted as $U = \{u_1, u_2, \ldots, u_m\}$. The UAV swarm adjusts its deployment position for each flight slot based on the user's perception. As a result, the trajectory of a UAV u_i within the swarm is represented as $q_{u_i}(t) = [x_{u_i}(t), y_{u_i}(t), z_{u_i}(t)]$. The distance between two UAVs is denoted by $d(q_{u_i}(t), q_{u_j}(t))$, where $i, j = 1, 2, \ldots, m$. The flying speed of a UAV u_i is denoted as v_i, its flight acceleration as a_i, and its maximum rotation angle as ϕ_i. Let r_{u_k} represent the radius of the UAV's collision danger area. The annular region between r_{u_i} and $r_{u_i} + s$ serves as the warning area, where s represents the threshold for the.

The flight path of the UAV swarm is represented by the sequence $q_{u_i} = [R_1, ..., R_{N_{u_i}}]$, where $T_n \in \mathscr{T}$ denotes the nth town visited by the UAV swarm, and $T_{N_{u_i}}$ indicates the total number of towns visited by u_i during the flight.

3.2.1 Problem Formulation

The MTSP formulation focuses on determining tour groups for all m UAVs, with each UAV starting and ending at the same source position. Each town must be visited by exactly one UAV, ensuring that all towns are covered while minimizing the total distance, time, and cost incurred by each UAV. This problem can be effectively modeled using graph theory as a weighted graph $G = (T, L, W)$, where T represents the set of towns (vertices), L represents the paths between towns (edges), and W represents the weights (distances) associated with each pair of towns.

A closed cycle in which all towns are visited exactly once, with each visit being distinct, is known as a Hamiltonian cycle. To minimize the total travel distance in a graph, it is essential to identify a set of H Hamiltonian cycles that incur the lowest travel distance together. The distances between towns are represented by a distance matrix $\mathbb{D}$, an $N \times N$ matrix. Each entry $d_{i,j}$ indicates the distance between towns i and j. The matrix $\mathbb{D}$ is termed symmetric if $d_{i,j} = d_{j,i}$ for all pairs (i, j); otherwise, it is referred to as asymmetric. In the integer linear programming framework, the objective is to minimize the total travel distance. Thus, the formulation of MTSP as an optimization problem can be expressed as follows:

$$\min \sum_{k=1}^{m} \sum_{i \in \mathscr{T}} \sum_{j \in \mathscr{T}} d_{ij} X_{ij}^{k} \tag{1a}$$

$$\text{s.t.} \quad \sum_{k=1}^{m} \sum_{i \in T} X_{ij}^{k} = 1 \quad \forall k \in U, \tag{1b}$$

$$\sum_{k=1}^{m} \sum_{j \in T} X_{ij}^{k} = 1 \quad \forall k \in U, \tag{1c}$$

$$\sum_{k=1}^{m} \sum_{j \in T} X_{ij}^{k} - \sum_{k=1}^{m} \sum_{i \in T} X_{ij}^{k} = 0, \tag{1d}$$

$$\forall k \in U, \forall i \in T\{0\},$$

$$\sum_{k=1}^{m} \sum_{i \in T} \sum_{j \in S} X_{ij}^{k} \leq |S| - 1, \tag{1e}$$

$$\forall k \in U,\ \forall S \subset T,\ S \neq \emptyset,\ S \neq T.$$

In equation (1a), the decision variables $X_{ij}^{k} \in \{0, 1\}$ are defined as binary variables, indicating whether the UAV will visit the next town, as defined below:

$$X_{ij}^{k} = \begin{cases} 1, & \text{if UAV } k \text{ travels from } i \text{ to } j \text{ and } i \neq j, \\ 0, & \text{otherwise.} \end{cases} \tag{3.2}$$

The constraint in (1b) specifies that each town must be visited exactly once, except for the starting point. Constraint (1c) ensures that each UAV departs from the depot and visits one town (other than the depot) exactly once. To prevent towns from being skipped or visited multiple times, constraint (1d) is applied. Additionally, constraint (1e) ensures that towns are divided into distinct areas for each UAV.

The likelihood of collisions among multiple UAVs in swarms is significantly high. Therefore, monitoring the distances between UAVs, obstacles, and among the UAVs themselves is essential. To tackle this issue, study integrating the objective function defined in (1a) with additional constraints (i.e., *UAV Assignment Constraints*) to enhance the allocation of towns. This is defined as follows:

$$\sum_{k=1}^{m} \sum_{i \in T} X_{ij}^{k} = 1 \quad \forall k \in U, \quad \text{if } F_{\mathrm{T}(i,j)} > 0, \tag{1f}$$

$$\sum_{k=1}^{m} \sum_{j \in T} X_{ij}^{k} = 1 \quad \forall k \in U, \quad \text{if } F_{\mathrm{T}(i,j)} > 0. \tag{1g}$$

The constraints (1f) and (1g) illustrate that if two towns are in close proximity to each other, they can be served by a single UAV. Additionally, if the force $F_{\mathrm{T}(i,j)}$ is greater than zero, a UAV will be assigned to serve both towns i and j.

Additionally, a new constraint, referred to as the *Collision Avoidance Constraint*, has been added to the objective function in (1a). This constraint is implemented to prevent collisions, preventing two UAVs from getting dangerously close to each other. The specific distance considered dangerous is defined as follows:

$$d_{\mathrm{U}(i,j)} \geq D_{\mathrm{s}} \quad \forall i, j \in U, \quad \text{if } F_{\mathrm{U}(i,j)} > 0. \tag{1h}$$

Constraint (1h) ensures a safe distance between two UAVs to avoid collisions. Here, D_{s} represents the safe distance threshold needed to prevent collisions within a swarm whenever $F_{\mathrm{U}(i,j)} > 0$ is positive.

These conditions are intended to collectively optimize the total distance travelled while also ensuring complete coverage and efficient allocation of towns among the UAVs. When designing the trajectory for the UAV swarm, it is essential to determine the optimal number of UAVs needed. In the formation flight for MTSP, each UAV in the swarm is assigned specific towns to cover.

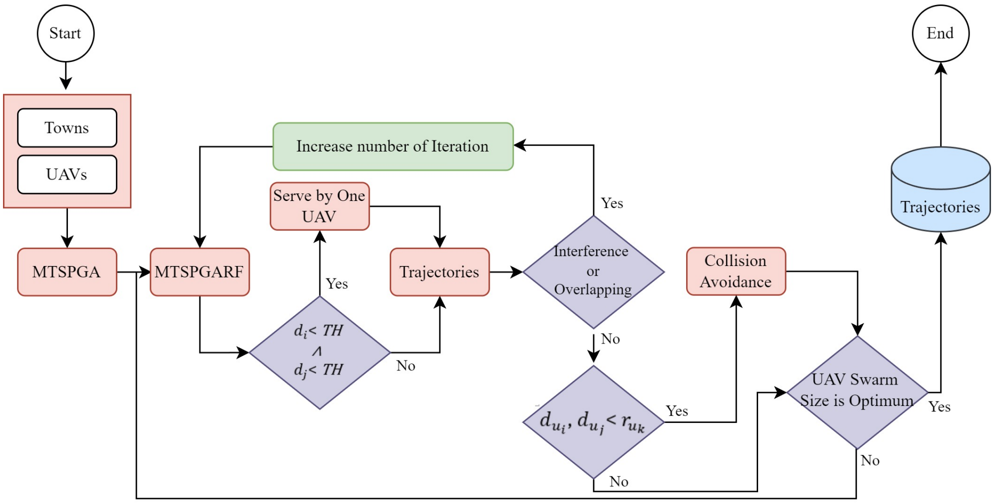


Fig. 3.2 Proposed GA-RF-based framework.

3.3 Proposed GA-RF based Framework for Trajectory Designing

An GA-RF model has been proposed to optimize the traveling distance, traveling time, and size of a UAV swarm while minimizing interference, overlapping, and avoiding collisions during trajectory design. The model features include:

1. The trajectory design begins with multiple UAVs and a fixed number of towns.

2. An MTSPGA is adapted to initiate the trajectory design process.

3. A RF is used in the MTSPGA to meet the desired objectives. The GA-RF checks the distance (d_{ij}) between towns i and j. One UAV will serve both nearby towns if this distance is less than a specified threshold (TH), known as the buffer zone.

4. Next, trajectories are generated to assess any interference or overlapping. If such issues are detected, the number of iterations in the GA-RF is increased, and the trajectories are re-evaluated. Once no interference or overlapping is found, the distance (d) for UAV (u_i) is checked. A collision avoidance method is implemented if this distance is less than the buffer zone D_s.

5. Finally, the optimization of UAV size is evaluated. Subsequently, trajectories are generated with optimized distances and UAV swarm sizes while ensuring minimal interference and overlapping.

Fig. 3.2 illustrates the proposed GA-RF framework for trajectory design. The model is organized into the following sections:

3.3.1 Problem Solution Using GA Driven RF

GAs are highly efficient heuristics used to obtain results for the MTSP and similar problems. While GAs do not guarantee that the solutions they produce are optimal, they often find solutions that are significantly closer to optimality in a shorter amount of time.

The underlying technology of GAs is based on principles of natural selection and genetics, which are used to improve results from previous populations (i.e., solutions). The process begins with an initial population generated randomly. A fitness function is then designed to evaluate the performance of each solution within the population at every iteration.

Following this evaluation, the algorithm selects two parent solutions and applies three key operations: crossover, offspring generation, and mutation. These operations generate two new populations, which are then included in the next generation. If the newly created generation has improved fitness values, it will replace the original parent population. This cycle of selecting the initial population, performing crossover, generating offspring, and mutation continues until the new generation reaches the designated population size, completing one iteration (or generation).

The algorithm continues to produce new generations until a specified number of generations is reached, as determined by the user. In GA methods, the initial populations are depicted as chromosomes, and various techniques exist to represent these chromosomes.

The pseudocode of MTSPGA is illustrated in **Algorithm 4** and the following details the procedures involved:

1. *Initial Population*:

   In practice, examining all possible search spaces can be quite costly. Therefore, it is essential to effectively investigate a portion of the research space known as the population to identify better solutions. The population is a subset of all potential solutions to a specific problem and serves as the initial process for all heuristic searches. Each individual within this gene-like population represents a viable solution to the problem.

   As depicted in Fig. 3.3, step 1 involves generating the initial population. The variable $A1 = n$ indicates the size of the chromosome C, which corresponds to the number of towns, with $n = 6$ representing six towns and $n = 1$ representing one gene (town) of the chromosome. $A2$ illustrates the first random selection of towns (genes), referred to as a chromosome.

Algorithm 4 UAVs Travel Distance Optimization Using MTSPGA

Input:
- Number of towns N
- Number of UAVs m
- Cost per meter c
- Maximum generations G

Output:
- Optimal routes for UAVs: $\{R_1^*, R_2^*, \ldots, R_{m^*}^*\}$

Define the set of towns: $\mathscr{T} \triangleq \{T_1, T_2, \ldots, T_N\}$

Initialize a random population of chromosomes $\{A_1, A_2, \ldots, A_k\}$, each representing a candidate solution (a sequence of towns)

Define the fitness function:

$$\text{Fitness}(A_i) = \frac{1}{c\sum_{j=1}^{N} d(T_j, T_{j+1})}, \quad \text{where } T_{N+1} = T_1$$

where $d(T_j, T_{j+1})$ is the Euclidean distance between consecutive towns in the route represented by A_i.

for $gen = 1$ to G **do**

  Select parents $\{A_a, A_b\}$ based on fitness probabilities

  Perform crossover at point cp to produce offspring n_1, n_2:

$$n_1[:cp] = A_a[:cp], \quad n_1[cp:] = A_b[cp:]$$

$$n_2[:cp] = A_b[:cp], \quad n_2[cp:] = A_a[cp:]$$

  Apply mutation with probability μ to offspring:

$$\text{Mutate}(n_i) = \text{swap}(n_i[j], n_i[k]), \quad \text{for random indices } j, k$$

  Evaluate fitness for n_1 and n_2

  Select the best chromosomes from $\{A_a, A_b, n_1, n_2\}$ to form the next generation

end for

Assign the final best chromosome A^* to UAV routes

Break A^* into m segments, ensuring minimal total distance for each UAV

return Optimal routes $\{R_i^*\}$

After several iterations, the algorithm produces additional chromosomes (subsequent populations), leading to $A3$, $A4$, and so on, depending on the number of iterations completed. The top chromosome with the minimum distance or cost is identified upon completing all iterations.

The objective function aims to minimize the distance travelled by a swarm, exemplified by n. The cost (c) traveled from town i to town j is set to $c = 2$ per meter. For example, the distance calculated in meters using the Euclidean distance formula is 20 meters, resulting in a cost of 40. Meanwhile, $A3$ records a distance of 21 meters, and this pattern continues up to n.

2. *Crossover*:

   The crossover process involves combining two different chromosomes to create a new one. This system is gradually refined through random selection to produce better chromosomes. In this process, parent chromosomes are selected, and one or more offspring are generated using their genetic material. A crossover point cp is established, indicating that the genes to the right of this point will be exchanged between the chromosomes $A2$ and $A3$. If this gene exchange results in a shorter distance compared to the distances in the previous configurations of $A2$ and $A3$, it is considered advantageous.

3. *Offspring*:

   The offspring refers to new solutions generated from the genetic material of individual (parent) solutions. This process is designed to mimic biological reproduction, aiming to produce new candidate solutions that may perform better.

   In Step 3, which focuses on generating offspring, the crossover shifting process is repeated, but this time on the left side of the crossover point (cp). For instance, if gene $n_{2,3}$ has a distance of 10 meters, and $n_{2,1}$ has a distance of 9 meters, study advances one step. However, $n_{1,1}$ does not satisfy the condition because i should not equal j. Therefore, study shuffle to $n_{1,3}$ where the distance is 8 meters. Next, the study sums the distances of the genes on the right side and evaluates the overall distance.

4. *Mutation*:

   The mutation process introduces random changes to an individual's genes (solution) as a way to maintain diversity and explore new solutions in a GA. If the resulting distance after mutation is minimal, the mutation step is executed as the final step of the algorithm. During mutation, $A2$ shuffles its genes from the second gene to the second-to-last gene. If this new arrangement produces a minimum distance, it replaces $A2$ and is deemed the optimal solution; otherwise, the process repeats.

5. *Fitness Function*:

   In this study, the fitness function is designed to minimize the total travel cost of UAVs. Each chromosome A_i represents a possible route, and the distances between all the towns traveled by UAVs on that route are collected. This total distance is multiplied by the cost per meter c, and the fitness function is defined as the inverse of this cost:

$$\text{Fitness}(A_i) = \frac{1}{c\sum_{j=1}^{N} d(T_j, T_{j+1})} \tag{3.3}$$

   This ensures that paths that produce shorter distances and lower costs have higher fitness, and those chromosomes are more likely to be selected for the next generation. As a result, over time, the GA evolves towards a solution that provides the least cost path for the UAVs.

After optimizing the solution to the traditional traveling salesman problem (TSP), the GA partitions the chromosome $A2$ into u parts and distributes these parts among the available UAVs to efficiently solve the MTSP. In Fig. 3.3 the study shows a scenario involving 3 UAVs and $n = 6$ hotspots.

This partitioning is based on a data structure that includes two elements:

RTE (route): The list of hotspots that are associated with a particular trajectory or path. BP (breakpoints): The points where the RTE is broken and distributed among different UAVs.

When BPs are applied, get three separate chromosomes:

$C = 1$ for the first UAV, $C = 2$ for the second UAV, and $C = 3$ for the third UAV.

3.3.2 Collision Avoidance between UAVs in Swarm Using Repulsion Force

Each UAV checks the distance to other UAVs around it, and if a UAV comes within a safe distance (buffer zone), it is identified as a potential collision risk. Suppose that for each UAV u_k the radius of the collision risk zone is r_{u_k}, which represents the specific buffer zone for that UAV. The distance between two UAVs u_i and u_j is represented by $d(d_{u_i}, d_{u_j})$. If this distance becomes less than r_{u_k}, a collision risk arises. If a collision risk exists, the trajectory of the respective UAV is updated to deviate from the collision path and follow a new, safe path. Otherwise, the UAV continues to head towards its assigned target town. The proposed collision avoidance mechanism is shown in Fig. 3.4 and its details are given in **Algorithm 5**.

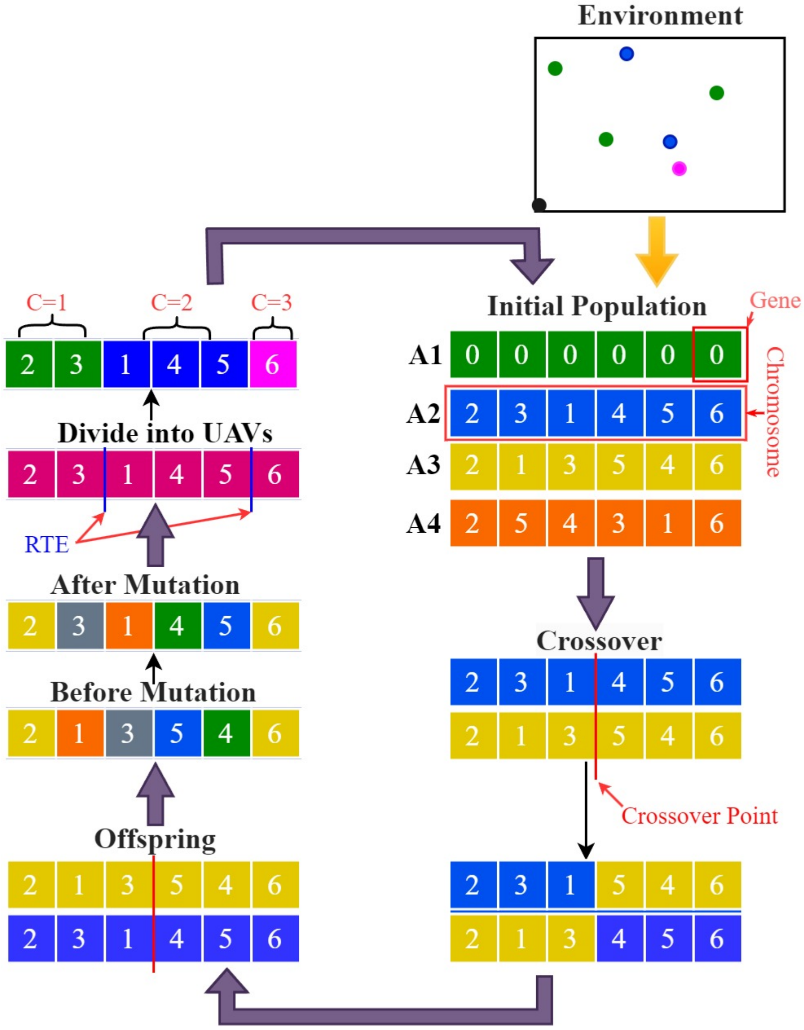


Fig. 3.3 Genetic Algorithm working steps.

This strategy helps maintain a safe distance between UAVs, especially when multiple UAVs operate simultaneously in a confined airspace.

To prevent collisions between UAVs and towns, the study utilize RF. A simplified version of Coulomb's Law is employed to model the RF between towns and UAVs. The pseudocode for calculating the RF between towns is detailed in **Algorithm 6**, while **Algorithm 7** explains the RF between UAVs [226, 178]. The total force f_i acting on UAV $u_i \in U$ is defined as follows:

$$\sum_{i,j(i \neq j)} F_{\mathrm{T}(i,j)} + \sum_{i,j(i \neq j)} F_{\mathrm{U}(i,j)}, \tag{2a}$$

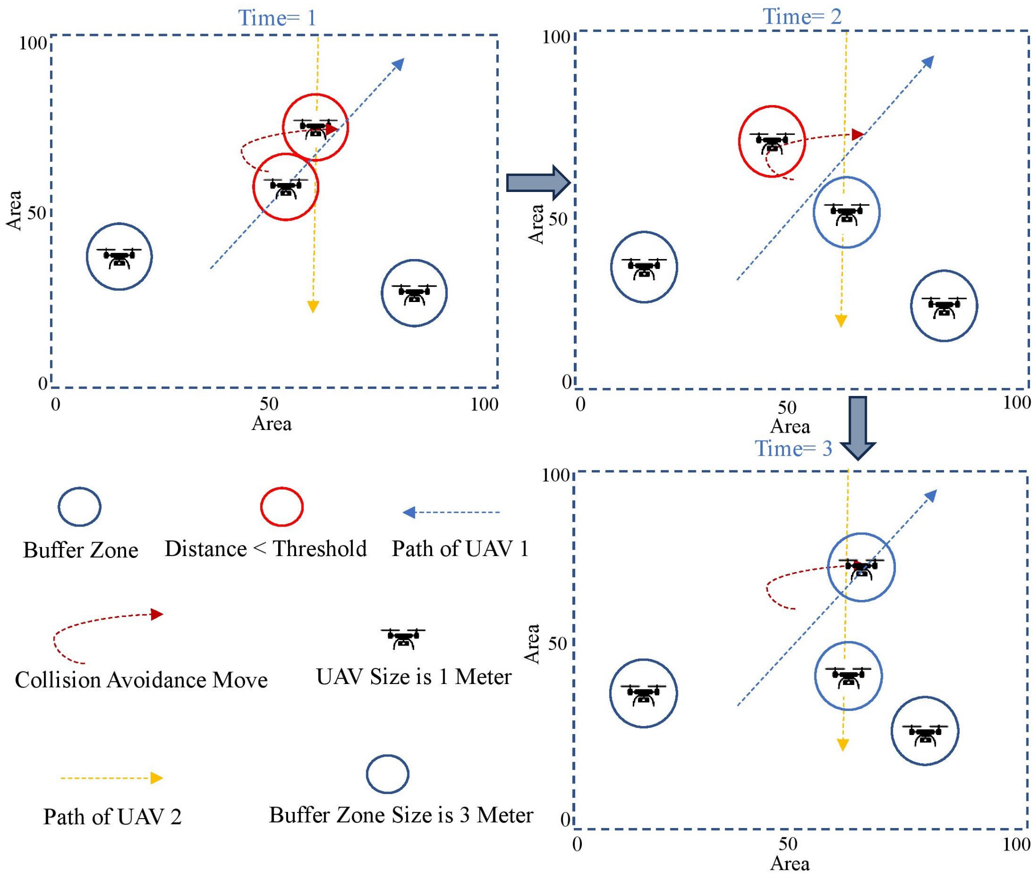


Fig. 3.4 Proposed method for collision avoidance.

where,

$$F_{\mathrm{T}(i,j)} = \frac{k_{\mathrm{T}(i,j)}}{d^2_{\mathrm{T}(i,j)}}, \tag{2b}$$

$$F_{\mathrm{U}(i,j)} = \frac{k_{\mathrm{U}(i,j)}}{d^2_{\mathrm{U}(i,j)}}. \tag{2c}$$

In this context, $F_{\mathrm{T}(i,j)}$ represents the repulsive forces between all pairs of towns, with indices i and j iterating through all possible combinations. Similarly, $F_{\mathrm{U}(i,j)}$ denotes the repulsive forces between pairs of UAVs, where l and m represent specific UAV indices. The constants $k_{\mathrm{T}(i,j)}$ and $k_{\mathrm{U}(i,j)}$ represent the strength of the repulsion forces for towns and UAVs, respectively The terms $d^2_{\mathrm{T}(i,j)}$ and $d^2_{\mathrm{U}(i,j)}$ indicate the squared distances between the respective entities (towns or UAVs).

Algorithm 5 Collision Avoidance for UAV Swarm

Input:
Set of UAVs $U = \{u_1, u_2, \ldots, u_m\}$ with positions (x_k, y_k)
Safe distance (buffer zone) r_{u_k} for each UAV
for each UAV $u_i \in U$ **do**
  for each UAV $u_j \in U, j \neq i$ **do**
    Compute distance:

$$d(u_i, u_j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$$

    if $d(u_i, u_j) < r_{u_k}$ **then**
      Compute repulsive force:

$$F_{\mathrm{U}(i,j)}$$

      Update trajectory of u_i to avoid u_j
    end if
  end for
  Continue moving toward assigned hotspot T_i
end for

3.3.3 UAV Swarm Size Optimization

The proposed GA-RF method systematically optimizes the size of a UAV swarm as detailed in **Algorithm 8**. This is achieved by applying the rules of MTSP to determine the optimal number of UAVs in the swarm. The optimization process examines swarm sizes S ranging from 2 to $|S|$ UAVs. The set T contains n target towns, which are the locations the UAV swarm must visit. The Euclidean distance methodology is used to calculate the distances between these locations, where $d(s)$ denotes the total distance traveled by the UAV swarm for a specific size $s \in S$.

Each UAV in the swarm is assigned a route represented as $R_i = \{R_{i1}, R_{i2}, \ldots, R_{im}\}$, where m is the number of targets assigned to the i-th UAV, and each $R_{i,j}$ corresponds to an element of T. This indicates the j-th target visited by the i-th UAV. The set U represents the UAVs in the swarm, and s indicates the size of the swarm. The objective is to determine the optimal swarm size that minimizes the total distance, which can be expressed as follows:

$$\min_{s \in S} d(m) = \sum_{i=1}^{m} \left(\sum_{j=1}^{|R_i|-1} d\left(r_{ij}, r_{i(j+1)}\right) \right) \tag{3}$$

Algorithm 6 Repulsion Force Between Towns

1: **Input:**
   Positions of two towns, $t_i = (x_i, y_i)$ and $t_j = (x_j, y_j)$
   Repulsion constant $k_{\mathrm{T}(i,j)}$
   Maximum repulsion range threshold θ_{T}
   Close proximity threshold D_t (town clustering threshold)
2: **Step 1: Compute Euclidean distance between towns**

$$\Delta x = x_i - x_j, \quad \Delta y = y_i - y_j$$

$$d_{\mathrm{T}(i,j)} = \sqrt{(\Delta x)^2 + (\Delta y)^2}$$

This quantifies the spatial separation between towns t_i and t_j.
3: **Step 2: Check maximum range for force**
4: **if** $d_{\mathrm{T}(i,j)} > \theta_{\mathrm{T}}$ **then**
5:     **return** 0 ▷ No repulsion force beyond range
6: **end if**
7: **Step 3: Check proximity for same UAV allocation**
8: **if** $d_{\mathrm{T}(i,j)} < D_t$ **then**
9:     **return** 1 ▷ Towns are assigned to same UAV if $F_{\mathrm{T}(i,j)} > 0$
10: **end if**
11: **Step 4: Compute repulsion force**

$$F_{\mathrm{T}(i,j)} = \frac{k_{\mathrm{T}(i,j)}}{d^2_{\mathrm{T}(i,j)}}$$

This ensures separation between towns that are too close.
12: **return** $F_{\mathrm{T}(i,j)}$

where $d\left(r_{ij}, r_{ij+1}\right)$ represents the distance between consecutive towns (r_{ij}) and (r_{ij+1}) in the route of the i-th UAV. Every town must be visited by the swarm exactly once, according to:

$$\bigcup_{i=1}^{m} R_i = T, \quad \text{and} \quad r_i \cap r_k = \emptyset, \forall i \neq k. \tag{4}$$

The swarm size must fall within a predefined range, denoted as $m \in M$. The optimization process starts with a minimum swarm size of $m = 2$. For each $m \in M$, the towns are distributed among the UAVs. A GARF is then employed to optimize the route R_i for each UAV, aiming to minimize the distance $d(m)$. The system computes $d(m)$ for each swarm size and identifies the optimal swarm size m^* that minimizes the overall distance $D(m)$ as

Algorithm 7 Repulsion Force Between UAVs

Input:
 Positions of two UAVs, $u_i = (x_i, y_i)$ and $u_j = (x_j, y_j)$
 Repulsion constant $k_{\mathrm{U}(i,j)}$
 Distance threshold θ_{U} (maximum effective range)
 Safe distance threshold D_s (minimum allowed separation)
Step 1: Compute Euclidean distance between UAVs

$$\Delta x = x_i - x_j, \quad \Delta y = y_i - y_j$$

$$d_{\mathrm{U}(i,j)} = \sqrt{(\Delta x)^2 + (\Delta y)^2}$$

This measures the spatial proximity between UAV u_i and UAV u_j.
Step 2: Enforce Collision Avoidance Constraint
if $d_{\mathrm{U}(i,j)} < D_s$ **then**
 return ∞ ▷ Violation of safe distance, penalize heavily
end if
Step 3: Check repulsion range
if $d_{\mathrm{U}(i,j)} > \theta_{\mathrm{U}}$ **then**
 return 0 ▷ No repulsion force beyond effective range
end if
Step 4: Compute repulsion force

$$F_{\mathrm{U}(i,j)} = \frac{k_{\mathrm{U}(i,j)}}{d^2_{\mathrm{U}(i,j)}}$$

This ensures that UAVs close to each other experience stronger repulsion.
return $F_{\mathrm{U}(i,j)}$

follows:

$$[m^* = \arg\min_{m \in M} d(m)]. \tag{5}$$

The output provides the optimal swarm size m^*, the configuration of routes $\{R_1^*, R_2^*, \dots, R_{m^*}^*\}$ for the optimal swarm, and the minimum total distance $D(m^*)$.

3.3.4 Constraint Alignment between MTSP Model and GA-RF Implementation

To explicitly demonstrate how the GA-RF framework enforces the mathematical constraints introduced in the problem formulation, the study provides a comprehensive mapping between

Algorithm 8 UAV Swarm Size Optimization using GA-RF

Input:
  Set of target towns $T = \{T_1, T_2, ..., T_n\}$,
  Possible swarm sizes $m = \{2, 3, ..., 10\}$
Output:
  Optimal swarm size m^*,
  UAV routes $\{R_1^*, R_2^*, ..., R_{s^*}^*\}$,
  Minimal total distance $d(s^*)$
Initialize $d_{\min} = \infty$
Initialize $m^* = 2$
for $m \in M$ **do**
  Distribute targets T among m UAVs to form initial routes $\{R_1, R_2, ..., R_m\}$
  Apply a GA to optimize each route R_i
  Calculate the total distance $d(m)$:

$$d(m) = \sum_{i=1}^{m} \left(\sum_{j=1}^{|R_i|-1} d(r_{ij}, r_{i(j+1)}) \right)$$

  if $d(m) < d_{\min}$ **then**
    $d_{\min} = d(m)$
    $m^* = m$
    Save optimal routes $\{R_1^*, R_2^*, \ldots, R_m^*\}$
  end if
end for
return Optimal swarm size m^*,

the theoretical constraints and their practical enforcement. This ensures methodological coherence between the model and the implementation.

- **Objective Function (Eq.** (1a)**, Eq.** (3)**):** The core fitness function minimizes the total distance traversed by all UAVs. This directly corresponds to minimizing the travel cost across all UAV routes as formulated in the MTSP objective.

- **Assignment Constraints (Eq.** (1b)**, Eq.** (1c)**, Eq.** (4)**):** Chromosome encoding ensures each town is assigned exactly once. Violations are penalized through:

$$\mathscr{P}_{\text{cov}} = |T| - \left|\bigcup_{i=1}^{m} R_i\right| + \sum_{i \neq j} \left|R_i \cap R_j\right| \tag{3.4}$$

  which captures both missing and redundant assignments.

- **Depot and Flow Constraints (Eq.** (1d)**):** All UAV paths are anchored to the same depot. This is enforced via initial population encoding and enforced closure of UAV trajectories.

- **Subtour Elimination (Eq.** (1e)**):** Infeasible subtours are avoided by penalizing disconnected segments through chromosome repair and fitness penalization mechanisms.

- **Repulsion-Based Assignment (Eq.** (1f)**, Eq.** (1g)**):** The repulsion force between towns is used to guide assignment. The interference penalty:

$$\mathscr{P}_{\text{int}} = \sum_{(t_i,t_j):d(t_i,t_j)<D_t} \mathbb{I}[t_i \in R_p, t_j \in R_q,\, p \neq q] \tag{3.5}$$

  discourages different UAVs from serving tightly clustered towns.

- **Collision Avoidance (Eq.** (1h)**):** If the distance between any two UAVs is less than the buffer zone D_s, a penalty is added:

$$\mathscr{P}_{\text{col}} = \sum_{u_i \neq u_j} \mathbb{I}[d(u_i,u_j) < D_s] \tag{3.6}$$

  and trajectory correction is performed using the repulsion mechanism in **Algorithms 5** and 4.

- **Swarm Size Optimization (Eq.** (5)**):** The algorithm iterates over candidate swarm sizes $m \in M$, and selects the optimal size m^* minimizing the global objective:

$$m^* = \arg\min_{m \in M} d(m) \tag{3.7}$$

The complete alignment is summarized in Table 3.1.

Table 3.1 Mapping of MTSP constraints to GA-RF implementation

Eq.	Model Constraint	Implementation in GA-RF
(1a), (3)	Total travel cost minimization	Fitness function minimizes total distance per UAV
(1b), (1c), (4)	Towns assigned exactly once, route coverage	Enforced via unique gene representation, $\mathscr{P}_{\text{cov}}$
(1d)	Balanced flow constraints	Routes begin/end at depot, route balancing
(1e)	Subtour elimination	Chromosome repair + penalty for disconnected subtours
(1f), (1g)	Repulsive force town assignment	Town proximity clustering via $F_{\text{T}(i,j)}$, see Algorithm 6
(1h)	Collision avoidance	Collision penalty $\mathscr{P}_{\text{col}}$ + trajectory correction (Algorithm 5 & 4)
(5)	Optimal swarm size	Loop over m, minimize total distance to find m^*

3.4 Results and Discussion

In this section, this study systematically evaluate the performance of proposed method for designing trajectories for the UAV swarm. This method effectively optimizes both distance and time, avoids collisions, and minimizes interference and overlapping. This study was conducted in several stages. Firstly, examined a scenario in which a UAV swarm serves multiple towns fixed in a geographical area denoted as g. Fixed the number of towns, T, and applied a GA to optimize the paths. Secondly, implemented the UAV swarm U to serve various towns. Thirdly, an RF was introduced between the towns and UAVs to further avoid collisions and reduce interference and overlapping. Fourthly, Optimized the size of the UAV swarm with the proposed method. Finally, to verify the effectiveness of the proposed approach in the UAV swarm path optimization process, experiments comparing the proposed method with modified versions of existing methods: 2-OPT [149], PSO [91], AC [73], SA [237], and MTSPGA [88].

Table 3.2 Simulation parameters

Parameter	Value	Description
g	1000×1000 m^2	Total geographical area where towns are located
U	3, 5, 10	UAVs in Swarm used in the simulation
T	20, 50, 100	Hotspots (Towns) excluding depot (start and end) position
Depot Location	(0, 0)	Fixed depot position for UAVs
Location Coordinates	Random values (1-100)	Coordinates for each town's location
Population Size	100 to 1000	In GA
Iteration	10^7	Maximum iteration in GA
Crossover Fraction	0.8	In GA fraction of population
Mutation Rate	0.1	In GA probability of mutation
Step Size	3 feet	Motion distance of each UAV in swarm
Time Step	1	Time used to reach motion distance of UAV
Buffer Zone	5 feet	RF threshold between UAVs

The parameters listed in Table 3.2 are used in all the following experiments. The operating system is 64-bit Windows 11 Pro, running on an Intel(R) Core(TM) i9-14900KF processor with a clock speed of 3.20 GHz. The system is equipped with a GeForce RTX 4090 graphics card and has 24 GB of RAM. The code for the experiments is written and implemented using MATLAB 2023b.

Fig. 3.5 illustrates an example of a possible scenario with 50 towns (or hotspots) represented by blue dots randomly distributed in a defined geographical area, along with a fixed depot position (black dot). Each town requires certain resources. The objective is to initiate the journey of a swarm of UAVs from the depot position to visit each town, with only one UAV serving each town once, deliver the required resources, before returning to the depot. This is an example of swarm intelligence and cooperative UAV trajectory planning, providing an ideal framework for autonomous drone logistics, disaster relief, and smart delivery networks.

The scenario illustrated in Fig. 3.5 serves as the example to be addressed using the proposed MTSPGA method, which generates the desired trajectory for the UAV swarm. In Fig. 3.6, it can be seen that the five trajectories corresponding to the five UAVs within the swarm, produced by the MTSPGA. Each UAV's path is represented by a different colored line, indicating that each town is served by a single UAV only once. After completing its assignment, each UAV returns to the depot to conclude the tour. Furthermore, Fig. 3.6 illustrates the close proximity of two towns within a red dotted circle, which indicates a

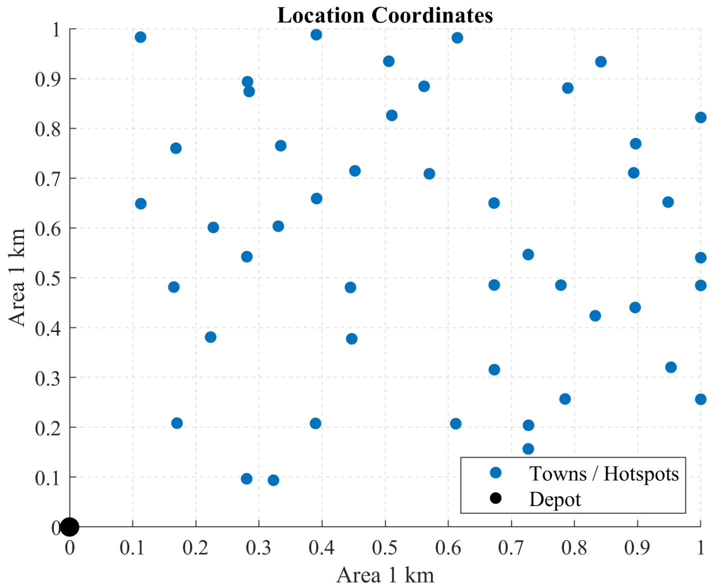


Fig. 3.5 Example of a realization consisting of 50 towns to be served by a UAV swarm of 5 UAVs.

potential risk for collisions and overlaps that could interfere with the UAVs' operational areas. This interference may disrupt the coordination of UAV swarms and reduce mission efficiency. Therefore, the objective of a single UAV is to intelligently serve both towns while minimizing overlap and lowering the risk of collisions. By integrating the proposed RF mechanism into the MTSPGA method, as depicted in Fig. 3.7, the UAVs are able to design their trajectories to avoid interference and minimize the overlap between their operational areas. Fig. 3.7 demonstrates that one UAV effectively serves both towns despite their close proximity, successfully managing the proximity threshold and avoiding interference.

The results presented in Fig. 3.6 and Fig. 3.7 is obtained using the same number of iterations, specifically 10^3, as applied in the proposed GA-RF method. This ensures a fair evaluation of swarm coordination and trajectory optimization. In Fig. 3.7, it becomes apparent that the issue of two close towns has been addressed; however, problems such as interference and potential collisions between UAVs remain. This highlights the necessity for further optimization and enhanced swarm coordination strategies. To tackle these challenges, the number of iterations was increazed to 10^7. After making this adjustment, the proposed GA-RF effectively reduced the interference problem, eliminating any risk of collisions in this scenario, as shown in Fig. 3.8.

Fig. 3.9 illustrates the movements of each UAV. These movements are considered as sub-steps taken by the UAVs on their way to their respective goals, which is crucial for trajectory design. The sub-step size is set to 3 feet, as indicated in Table 3.2. This parameter is vital for

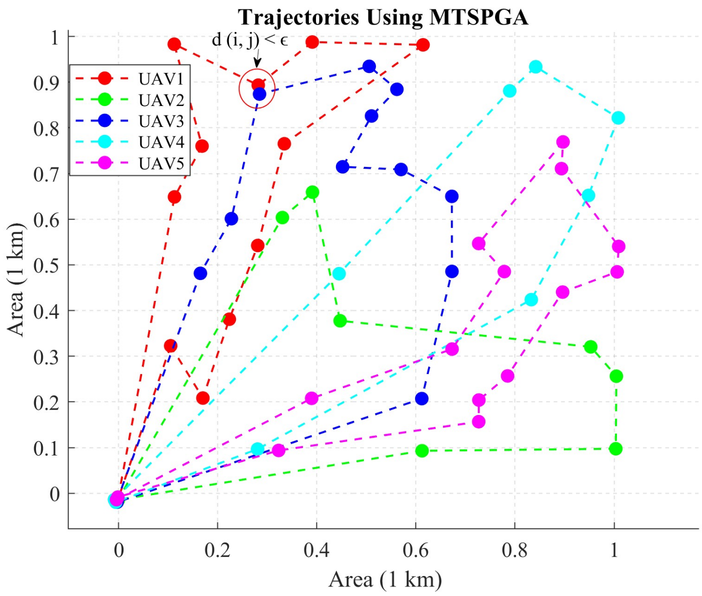


Fig. 3.6 Generated trajectories using MTSPGA.

ensuring precise movement control of the UAVs and for generating smooth trajectories. It plays a significant role in facilitating collision-free navigation and optimizing energy-efficient flight planning.

In Fig. 3.10-(a), the sub-steps are clearly illustrated. However, the red circle indicates that there are still potential collisions. By using the proposed GA-RF method, Fig. 3.10-(b) shows the UAV swarm trajectory animation with collision avoidance between the UAVs. The zigzags in the trajectory indicate that the UAVs are actively avoiding collisions with others that are moving close to their thresholds. The actual path is depicted in red, while the collision avoidance trajectory, also known as the ad-hoc trajectory, is shown in green.

The results presented in Fig. 3.6 to Fig. 3.10 successfully achieved the objectives of removing interference, avoiding collisions, and preventing overlap. To further assess the performance of the proposed GA-RF method, this study will conduct a comprehensive analysis of the total distance covered and the completion time using 100 towns and 5 UAVs operating in the same geographical area and under similar conditions. This study will compare the proposed method against several heuristic algorithms, including MTSPGA, PSO, 2-OPT, SA, and AC.

The analysis in Fig. 3.12-(a and b) illustrates how the results of a GA vary with different mutation rates when the Population is tested at varying levels. Where, Pop.1 denotes a population of 100, and Pop.10 denotes a population of 1000. The mutation rate is used from 0.01 to 1.0, and the total distance and execution time of the algorithm are shown for each rate.

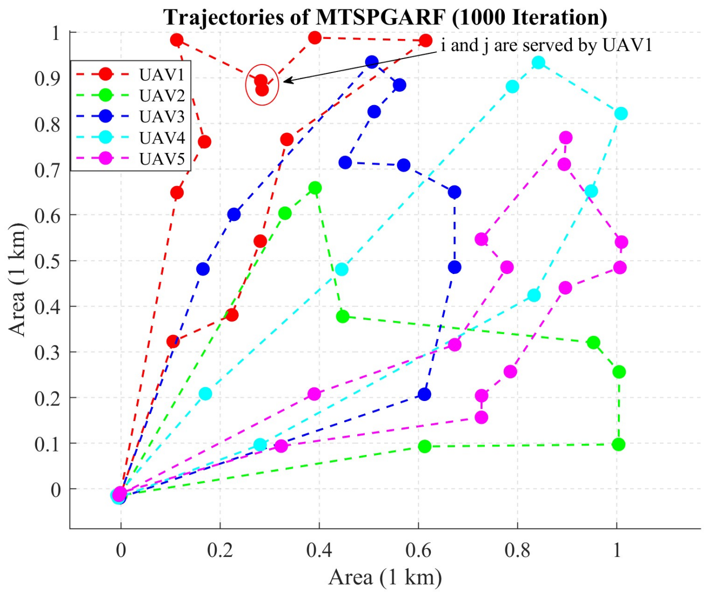


Fig. 3.7 Generated trajectories using GA-RF with 10^3 iterations.

As the mutation rate increases, the probability of chromosome mutations increases. At low mutation rates (0.01–0.07), the algorithm reaches a solution quickly, but that solution often stops at a local optimum. At mutation rates (0.08–0.1), an equilibrium is observed where the algorithm discovers new paths and reaches a better solution. At high mutation rates (0.2–1.0), the improvement in distance stops due to excessive randomness, and the solution becomes unstable.

Furthermore, increasing population size also had a significant effect on distance. For example, in the initial 100 populations (Pop.1), the algorithm usually found a path that provided the minimum distance; however, at large populations (Pop.10 = 1000), due to the high diversity, longer paths are sometimes chosen. Meanwhile, the optimal results are obtained at populations (e.g. Pop.. 10 = 1000), where the distance is minimal; however, the time increases as the population grows.

The timing analysis in Fig. 3.12-(c and d) illustrates that as the mutation rate and population size increase, the processing time also increases, due to the higher computational complexity resulting from more individuals (Chromosomes) and mutations. The time is shorter at Pop.1, while it is longer at Pop.10, so with a larger population, the chances of improving the quality of the solution increase; however, the time also increases.

Furthermore, 100 iterations are used in this analysis, which reveals that the results obtained with a mutation rate of 0.1 and a population size of 1000 are optimal. As a result, the study adopted this combination in its analysis and experiments, and further tested it

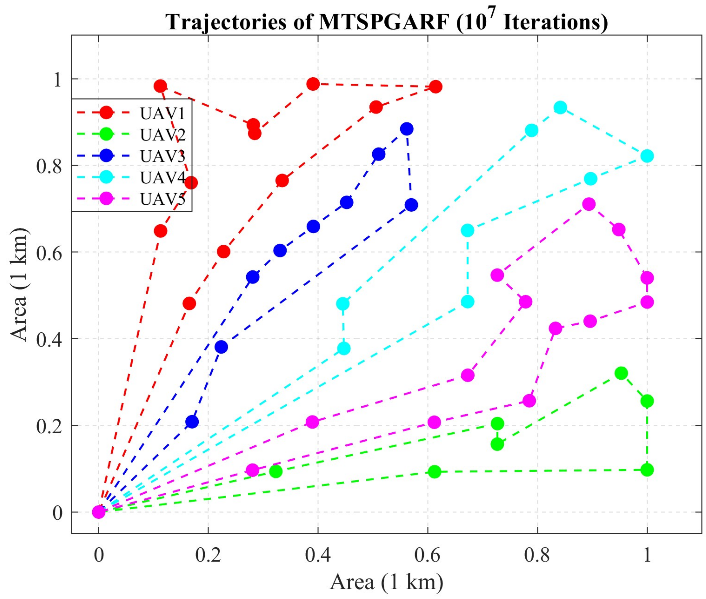


Fig. 3.8 Generated trajectories using GA-RF with 10^7 iterations.

with different iteration rates to improve the performance of GA. This analysis provides an important guide for GA in complex problems, such as UAV swarm trajectory design, by showing how to balance mutation rate, population size, and iteration order to achieve optimal results.

In Fig. 3.11-(a), it can be observed that three UAVs are in a swarm. The distance decreases with an increasing number of iterations, resulting in the GA-RF method achieving a minimized distance compared to other algorithms. Notably, the distance of 2-OPT remains minimal compared to the others up to 10^6 iterations, but it does not continue to decrease after that point. As shown in Fig. 3.11-(b) to Fig. 3.11-(h), it can be seen that as the number of UAVs increases, the performance of the proposed method significantly improves relative to the other algorithms. This improvement is mainly due to the use of the GA, which benefits from an increased number of iterations. More iterations lead to a greater number of generations in the GA, enhancing the likelihood of selecting the minimal distance. In Fig. 3.11-(h), when the study considers a swarm of 10 UAVs, the performance of 2-OTP is comparable to the proposed method after 100 iterations. However, as the number of iterations increases, the performance of the proposed GA-RF method surpasses that of all other approaches, demonstrating its superior efficiency.

Fig. 3.13 illustrates the performance of proposed GA-RF method, specifically in terms of the time taken by the UAV swarm to complete its tour with varying swarm sizes. The completion time for 2-OPT is minimal at 100 iterations. However, as shown in Fig. 3.13(a-h),

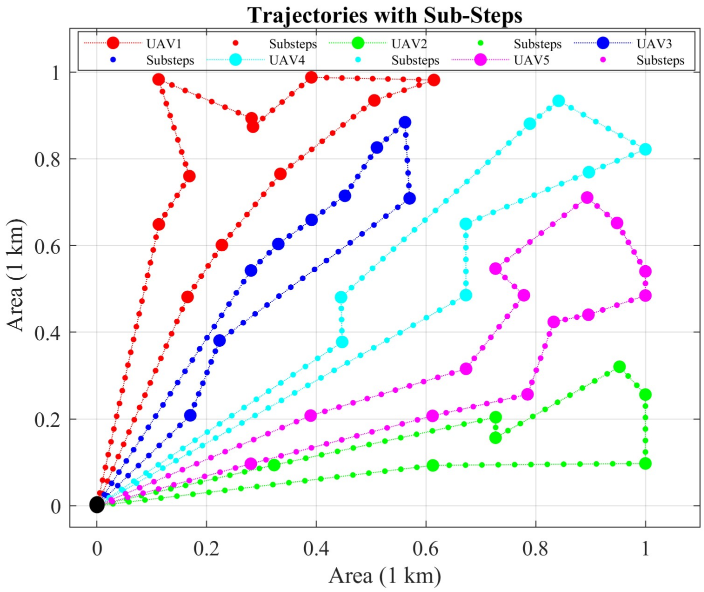
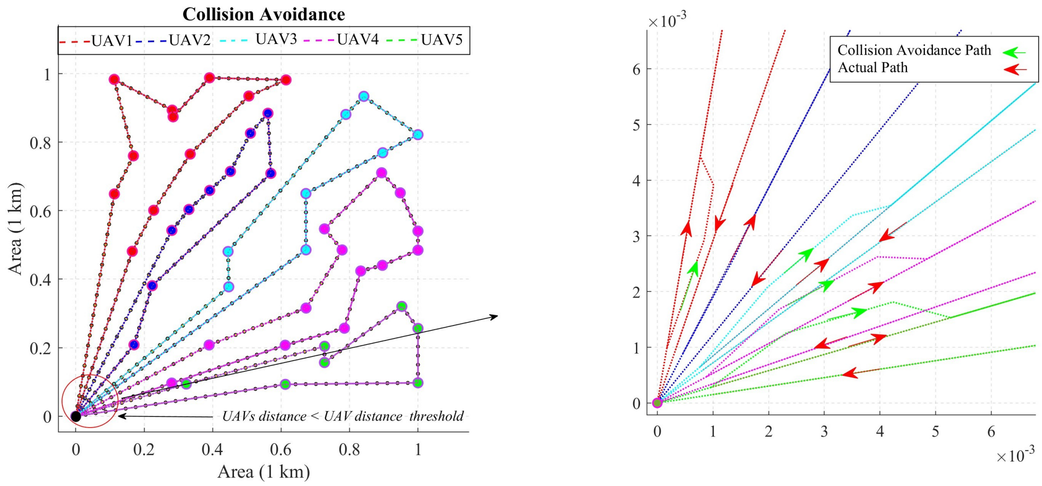


Fig. 3.9 Swarm trajectories produced by the GA-RF method in conjunction with UAV movements.



(a) (b)

Fig. 3.10 (a) UAVs collision avoidance. (b) UAVs actual and collision avoidance paths.

the time decreases as the number of iterations increases, even when the number of UAVs in the swarm and the hotspots remain constant. Notably, in Fig. 3.13(a), when the number of iterations reaches 10^7, the time taken by the proposed GA-RF method to complete the task is the lowest compared to all other methods. This analysis confirms that the GA-RF method is not only efficient in reducing completion time but also highly effective for large-scale UAV swarms.

Optimizing the size of a UAV swarm is crucial during the deployment process, as it directly relates to the number of towns to be served and the time available to complete a specific mission. Fig. 3.14 illustrates the relationship between swarm size and its impact on travel distance (or completion time), ultimately enhancing overall performance efficiency. In Fig. 3.14(a), study evaluated the effectiveness of proposed method for selecting the optimal size of a UAV swarm tasked with serving 100 towns, starting with 10 UAVs stationed at the depot. The analysis indicates that a swarm size of 7 UAVs is optimal, as it achieves the minimum distance over 100 iterations, as shown in Fig. 3.14(b). Furthermore, when considering 10^3 and 10^4 iterations, deploying 8 UAVs proves to be more effective for the swarm. In contrast, at 10^5 and 10^6 iterations, using 9 UAVs emerges as the best choice. This evidence demonstrates that the proposed method can successfully identify the ideal number of UAVs in a swarm, in various UAV applications.

3.5 Summary

This chapter presents the MTSPGA–RF framework, a novel offline optimization model developed for UAV swarm trajectory planning. The framework enhances the classical MTSP by incorporating repulsion forces inspired by Coulomb's law. The proposed method employs a GA that generates efficient and safe trajectories for multiple UAVs by addressing UAV-specific challenges such as collision avoidance, path interference reduction, and fair path allocation.

Simulation results demonstrate that GA–RF achieves significant improvements over classical metaheuristic algorithms, such as MTSP–GA, PSO, SA, and ACO, particularly in terms of travel distance, mission time, and energy consumption. Additionally, the framework incorporates a Swarm Size Optimization Mechanism that determines the minimum number of UAVs required to completely cover the target area, thereby enabling effective and resource-efficient mission planning.

An important contribution of this research is that the proposed framework generates high-quality trajectory datasets for future intelligent UAV swarm systems. These datasets serve as the foundation for learning-based decision-making models, enabling autonomous and real-time coordination in complex and dynamic environments.

Finally, this chapter develops collision-free and optimized paths that serve as expert knowledge for constructing the World Model. This model will provide the UAV swarm with intelligent and autonomous decision-making capabilities through the Active Inference framework discussed in the next Chapter (4).

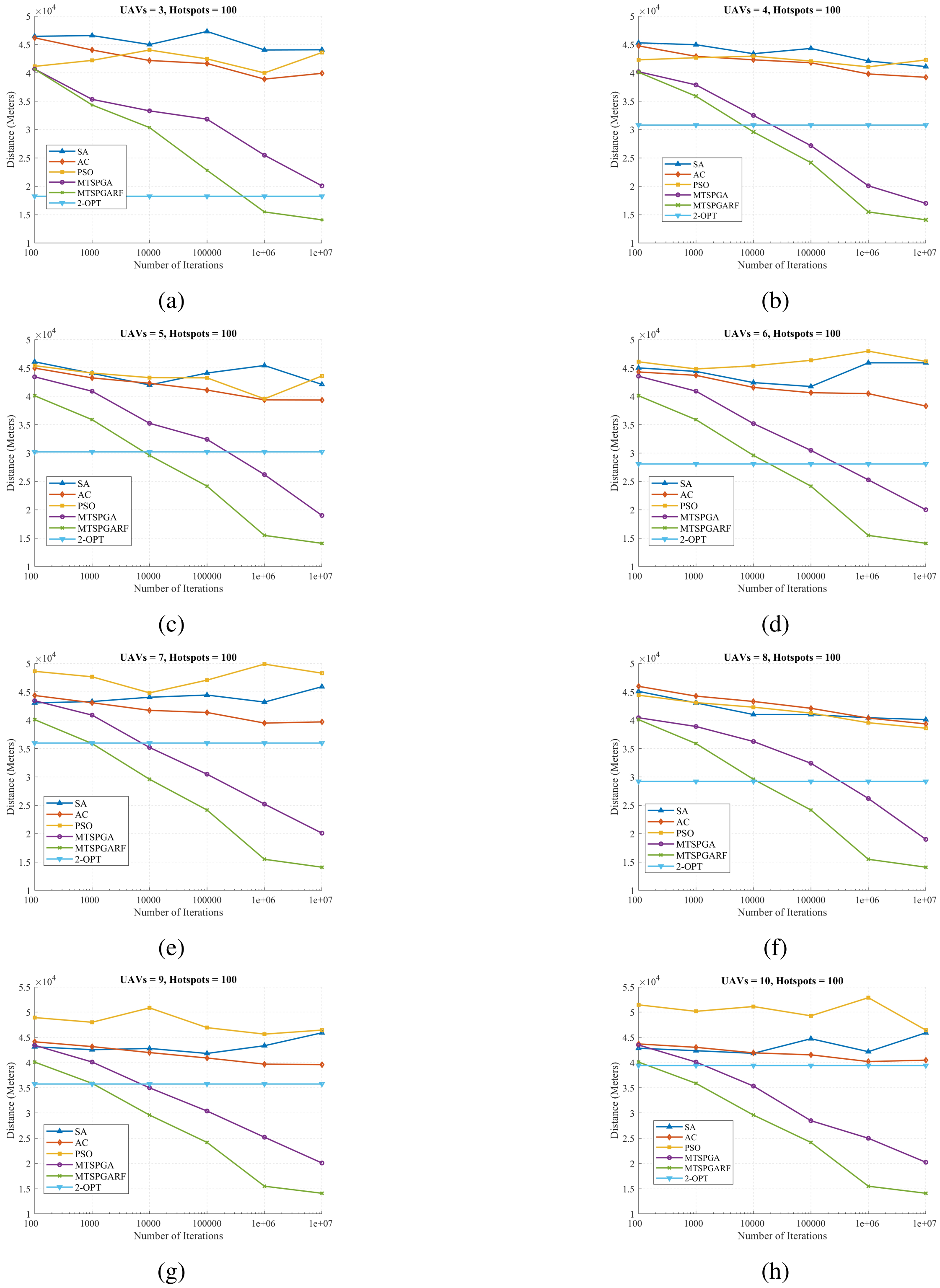


Fig. 3.11 The performance of the proposed GA-RF in terms of distance, compared with MTSPGA, PSO, 2-OPT, SA, and AC for different numbers of UAVs in a swarm: (a) The reference graph is represented for 3 UAVs in a swarm. (b) The updated graph is represented for 4 UAVs in a swarm. (c) The updated graph is represented for 5 UAVs in a swarm. (d) The updated graph is represented for 6 UAVs in a swarm. (e) The updated graph is represented for 7 UAVs in a swarm. (f) The updated graph is represented for 8 UAVs in a swarm. (g) The updated graph is represented for 9 UAVs in a swarm. (h) The updated graph is represented for 10 UAVs in a swarm.

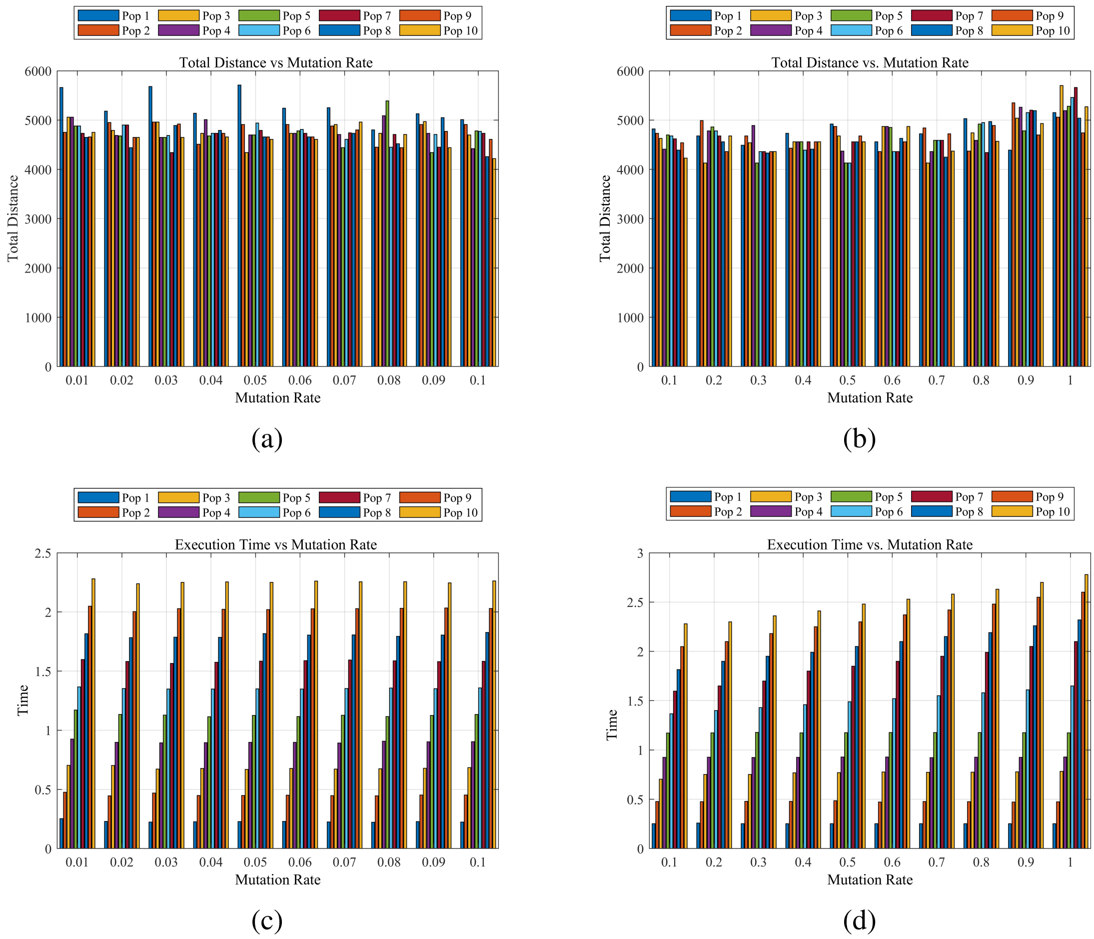


Fig. 3.12 Analysis of distance travelled, execution time vs different mutation rates with respect to population size: (a). Distance travelled vs mutation rate (0.01-0.1) with respect to population size (100-1000). (b) Distance travelled vs mutation rate (0.1-1.0) with respect to population size (100-1000). (c) Execution time vs mutation rates (0.01-0.1) with respect to Population size (100-1000), and (d) Execution time vs mutation rates (0.1-1.0) with respect to population size (100-1000),

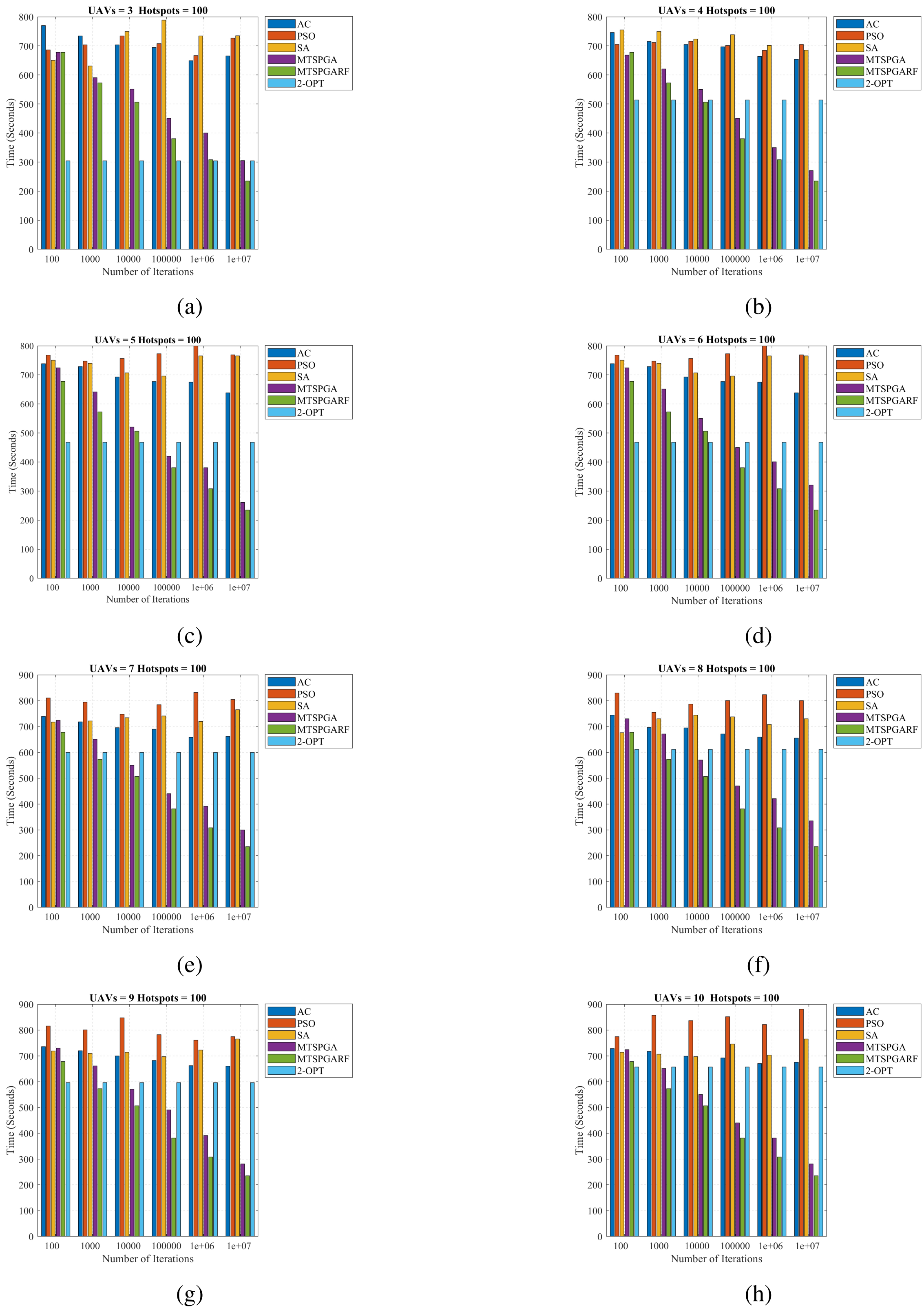
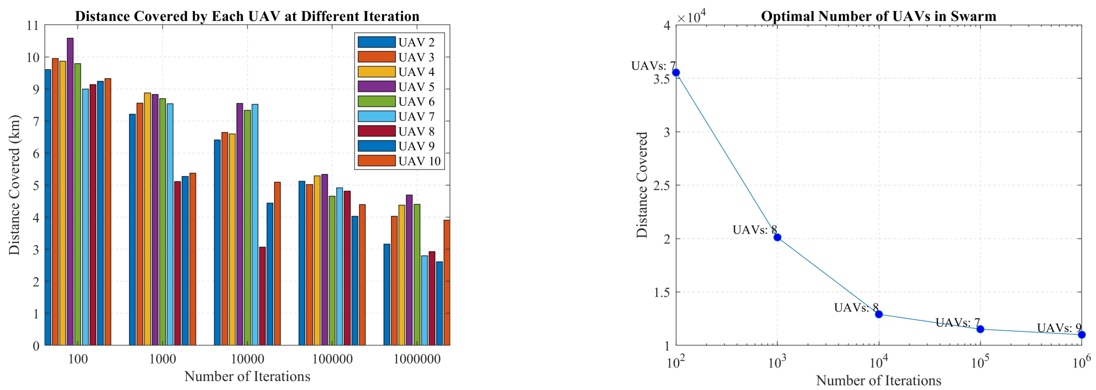


Fig. 3.13 The performance of the proposed GA-RF is evaluated in terms of completion time. It is compared with the following algorithms: MTSPGA, PSO, 2-OPT, SA, and AC. The evaluation is conducted for 100 hotspots (or towns) and various UAV swarm sizes: (a) 3 UAVs in the swarm, (b) 4 UAVs in the swarm, (c) 5 UAVs in the swarm, (d) 6 UAVs in the swarm, (e) 7 UAVs in the swarm, (f) 8 UAVs in the swarm, (g) 9 UAVs in the swarm, (h) 10 UAVs in the swarm



(a) (b)

Fig. 3.14 Selection of optimal number of UAVs in the Swarm.

Chapter 4

Active Inference-Driven World Modeling for Adaptive UAV Swarm Trajectory Design.

This chapter presents a novel Active Inference–based Trajectory Design Framework for UAV swarms. The proposed approach enables UAVs to autonomously perform mission distribution, route ordering, and motion planning through probabilistic reasoning and self-learning. In the offline phase, expert trajectories are generated using a GA–RF optimizer proposed in Chapter 3 and used to train a World Model that captures UAV swarm behavior across mission, route, and motion abstraction levels [18]. During online operation, each UAV infers optimal actions by continuously minimizing divergence between current beliefs and reference states encoded in the world model, allowing the swarm to adapt to new targets and environmental changes in real time. Furthermore, control strategies based on EKF and PF filters ensure smooth trajectories and effective collision avoidance, providing an interpretable, adaptable, and scalable framework for UAV swarms in complex and uncertain environments. The results demonstrate faster convergence, improved stability, and safer navigation compared to Q-Learning, establishing the proposed framework emerge as a scalable and knowledge-based solution for future intelligent UAV swarm networks. Moreover, the proposed model also performed effectively when tested on real-time simulated data, further strengthening its generalization ability and applicability to real-world scenarios.

4.1 Introduction

UAV swarms have recently gained significant attention due to their potential for distributed autonomy, scalability, and cooperative decision-making [59]. Their applications now extend beyond surveillance and mapping to dynamic multi-agent missions such as coordinated inspection, transportation, and communication support. The main challenge lies in enabling multiple UAVs to collaboratively plan trajectories that minimize energy, avoid collisions, and adapt to changing environments while maintaining global mission objectives [18].

Classical optimization frameworks offer deterministic solutions but depend on complete prior knowledge of system parameters, making them impractical in uncertain or dynamic conditions [113, 30]. Metaheuristic approaches, including Genetic Algorithms, provide flexibility but require extensive recomputation for new missions and lack online adaptability [198, 54, 124]. Similarly, data-driven models such as Deep Reinforcement Learning (DRL) and Multi-Agent Reinforcement Learning (MARL) [174, 111] achieve autonomy through experience but demand large training datasets and suffer from limited generalization to unseen scenarios.

Recent research trends are converging toward probabilistic generative modeling and cognitive inference frameworks that unify perception, reasoning, and action [152, 143]. Among these, Active Inference has emerged as a powerful Bayesian principle that links prediction, observation, and control through continuous belief updating. By minimizing the divergence between predicted and observed states, Active Inference provides a mathematically coherent mechanism for adaptive behavior under uncertainty.

Building on this paradigm, this work introduces an Active Inference–Based Framework for UAV Swarm Trajectory Design. The framework integrates probabilistic reasoning with symbolic hierarchical decision-making to enable UAVs to perform self-consistent, multi-level adaptation in real time. It combines model-based structure with learning-based flexibility, allowing each UAV to reason about mission allocation, route ordering, and motion generation within a unified cognitive process.

The main contributions of this chapter are as follows:

- A World Model that represents UAV swarm behavior across mission, route, and motion abstraction levels.
- A Probabilistic Decision-Making Mechanism based on Active Inference, enabling UAVs to minimize divergence between expected and observed symbolic states for adaptive mission planning.

- A Filter-Assisted Control Strategy that integrates filters (EKF and PF) estimation for smooth trajectory correction and robust collision avoidance.
- This research provides a unified probabilistic–symbolic framework for UAV swarm autonomy, offering interpretable, adaptive, and scalable trajectory design suitable for complex, uncertain environments.

The chapter is organized as follows: Section II introduces the system model and problem formulation. Section III details the proposed active inference based framework. Section IV presents the active inference decision making and online action. Section V discusses the results and provides an analysis. Lastly, Section VI includes the conclusion and outlines future work.

4.2 System Model and Problem Formulation

We consider a swarm of Q unmanned aerial vehicles (UAVs), $U = \{u_1, \ldots, u_Q\}$, deployed to cooperatively visit a set of N target locations $C = \{c_1, \ldots, c_N\}$ within a bounded area. Each UAV u_q starts and ends at a common depot $L_0 = [x_0, y_0, z_0]$ and is described by its position $\mathbf{x}_q(t) = [x_q, y_q, z_q]$, velocity $v_q(t)$, and heading change $\Delta\phi_q(t)$. UAVs maintain altitude limits and a minimum inter-UAV distance $d_{\min}$ to ensure safe operation.

The swarm must ensure that each target $c_i \in C$ is visited once by exactly one UAV while minimizing overall mission cost (distance, time, or energy). The decision process is organized hierarchically:

High-level (Mission Allocation): The set of target towns $\mathscr{C}$ is partitioned into Q disjoint subsets $\{\mathscr{C}_1, \mathscr{C}_2, \ldots, \mathscr{C}_Q\}$, where $\mathscr{C}_q$ is assigned to UAV u_q. The allocation aims to balance workload among UAVs and minimize total travel cost, subject to

$$\bigcup_{q=1}^{Q} \mathscr{C}_q = \mathscr{C}, \qquad \mathscr{C}_i \cap \mathscr{C}_j = \emptyset, \ \forall i \neq j. \tag{4.1}$$

Medium-level (Route Sequencing): For each UAV u_q, determine the visiting order of its assigned towns,

$$\pi_q = [c_{q,1}, c_{q,2}, \ldots, c_{q,|\mathscr{C}_q|}], \tag{4.2}$$

which minimizes the local travel distance or time. This level is analogous to solving a traveling salesman problem (TSP) for each UAV.

Low-level (Trajectory Generation): Given the visiting order π_q, UAV u_q generates a dynamically feasible and collision-free trajectory $\mathbf{x}_q(t)$ connecting consecutive towns, while respecting dynamic and safety constraints: Let $d_{i,j}$ denote the Euclidean distance between two towns c_i and c_j, forming the distance matrix $\mathbb{D} = [d_{i,j}]_{N\times N}$. Each UAV u_q follows a closed tour that starts and ends at the depot, visiting all towns in $\mathscr{C}_q$ exactly once. The optimization problem is formulated as a multi-traveling salesman problem (MTSP) with the following decision variables:

$$X_{i,j}^q = \begin{cases} 1, & \text{if UAV } u_q \text{ travels directly from city } i \text{ to } j, \\ 0, & \text{otherwise.} \end{cases} \tag{4.3}$$

The objective is to minimize the total travel cost of all UAVs:

$$\begin{aligned}
\min_{X_{i,j}^q} \quad & \sum_{q=1}^{Q}\sum_{i=1}^{N}\sum_{j=1}^{N} d_{i,j}X_{i,j}^q, && && \text{(4.4a)} \\
\text{s.t.} \quad & \sum_{q=1}^{Q}\sum_{i=1}^{N} X_{i,j}^q = 1, && \forall j, && \text{(each city visited once)} \quad \text{(4.4b)} \\
& \sum_{i=1}^{N} X_{i,j}^q - \sum_{j=1}^{N} X_{j,i}^q = 0, && \forall q, \forall i, && \text{(flow conservation)} \quad \text{(4.4c)} \\
& \sum_{i\in S}\sum_{j\in S} X_{i,j}^q \le |S| - 1, && \forall S \subset \mathscr{C}, && \text{(no subtours)} \quad \text{(4.4d)}
\end{aligned}$$

The global problem therefore determines, simultaneously, (i) the optimal allocation of towns among UAVs, (ii) the optimal visiting order for each UAV, and (iii) the feasible trajectories connecting the towns. Because of its combinatorial nature and high dimensionality, this problem is computationally challenging to solve exactly for large N and Q. To approximate the global optimum, a *genetic optimization algorithm* is employed. Each individual (chromosome) in the population encodes a potential solution consisting of city allocation, route sequencing, and trajectory parameters. The fitness of each individual is evaluated using the global cost function J in (4.4a), with penalty terms included for constraint violations such as collision risk or unvisited targets. Through iterative evolution using selection, crossover, and mutation operations, the algorithm converges toward near-optimal coordinated plans for the UAV swarm.

4.3 Proposed Active Inference-Based Framework

The proposed approach transforms the deterministic MTSP formulation into a self-learning, probabilistic decision system governed by the principle of Active Inference. It consists of two tightly coupled phases: (i) an offline learning phase, where a hierarchical world model is learned from expert demonstrations generated by the GA–RF optimizer, and (ii) an online active inference phase, where UAVs perform adaptive trajectory planning and belief updating in real time. Proposed framework is depicted in Fig. 4.1. The framework features include:

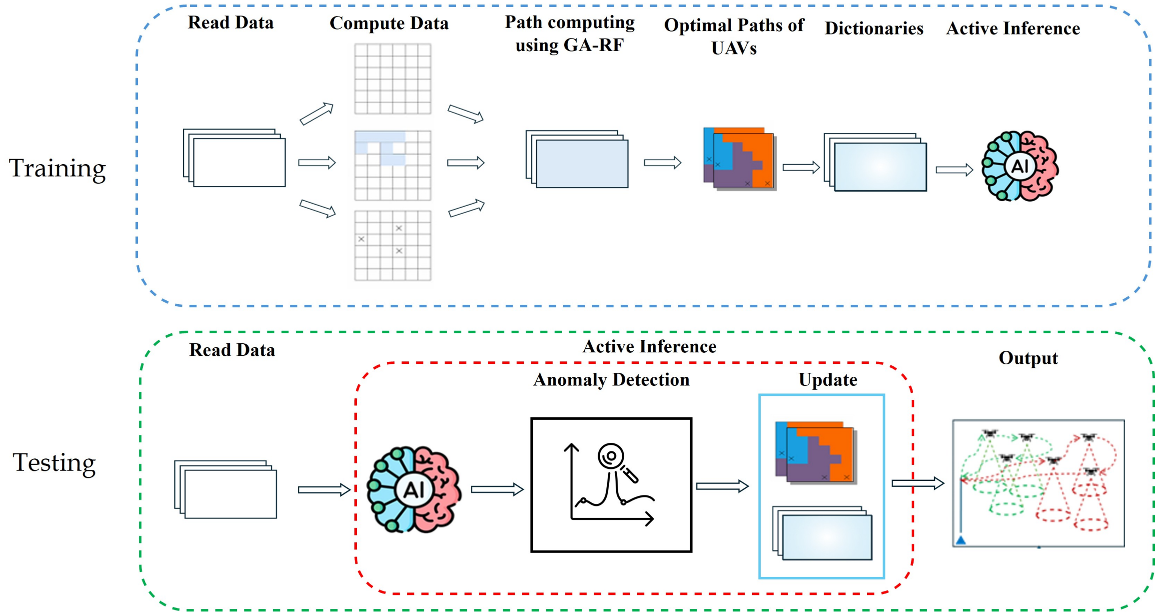


Fig. 4.1 Workflow of proposed Active inference-driven world modeling for adaptive UAV swarm trajectory design

- **Training process:** The model learns an internal representation of the world from its experiences, where generative models and dictionaries are built.
- **Data reading and computation:** The system reads environmental information such as city locations and UAV characteristics to create generative priors, which form the basis for learning.
- **Path computation (GA–RF):** The GA–RF optimizer generates different path possibilities, which makes it possible to learn the distribution and transfer of missions between UAVs.
- **Best path search:** For each UAV, the GA–RF algorithm finds the paths that are most efficient in terms of time, distance, and energy.

- **Dictionary construction:** All paths and decisions are stored in symbolic dictionaries that act as internal world models in active inference.
- **Integration with Active Inference:** When the dictionaries are complete, the system becomes a generative agent that makes decisions based on the principle of least surprise.
- **Online phase (Testing):** The system makes decisions based on real-time data, updating its model according to new observations.
- **Real-time Data Reading:** New data coming from UAVs are incorporated into the observation model to represent real situations.
- **Active Inference and Anomaly Detection:** When the difference between the prediction and the actual observation increases, the system updates its beliefs to reduce surprises.
- **Beliefs and Dictionary Update:** In the new situation, the system performs online learning by updating the dictionary and transition matrix under the Bayesian principle.
- **Output Generation:** Based on the updated beliefs, new paths and movements are predicted to effectively reach the target.
- As a result, the UAV swarm becomes an active cognitive system capable of self-learning, decision-making, and adjusting to changing conditions.

4.3.1 Expert Demonstrations via GA–RF

The GA–RF jointly optimizes mission allocation, route ordering, and trajectory feasibility. Each chromosome encodes a multi-UAV solution, while repulsion forces impose penalties whenever the inter-UAV distance approaches $d_{\min}$. For each mission instance D_m, the optimizer yields an expert demonstration

$$\tau_m = \{D_m, \mathscr{A}_m, \mathscr{P}_m, \mathscr{M}_m\}, \tag{4.5}$$

where D_m defines the cities and UAVs, $\mathscr{A}_m$ the mission allocation (division of cities), $\mathscr{P}_m$ the visiting order of each UAV, and $\mathscr{M}_m$ the corresponding motion trajectories. These demonstrations form a structured dataset describing how global missions are decomposed into ordered routes and feasible motions. They serve as the empirical foundation for constructing the generative world model.

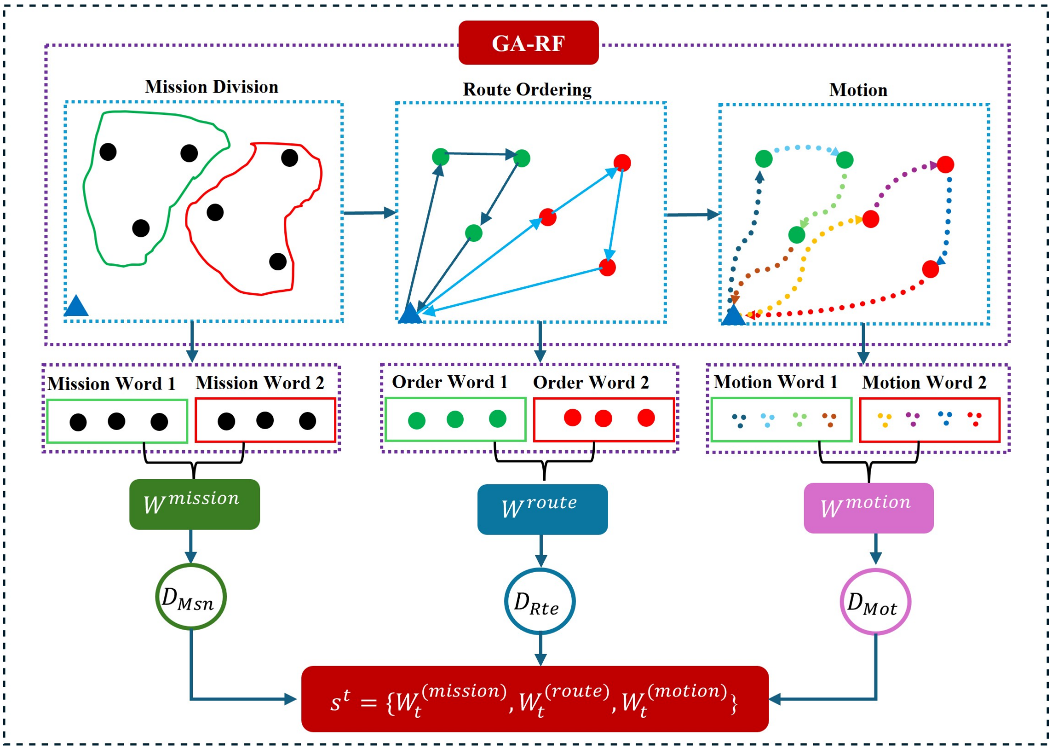


Fig. 4.2 Hierarchical world model construction from GA–RF expert demonstrations, encoding mission division, route ordering, and motion behavior across abstraction levels.

4.4 Active Inference—Decision Making and Online Action

4.4.1 Hierarchical Symbolic World Model

Each expert demonstration τ_m is transformed into a *hierarchical symbolic representation*, as illustrated in Fig. 4.2, which captures UAV swarm behavior across multiple levels of abstraction. This representation organizes knowledge into three interconnected symbolic dictionaries—*Mission*, *Route*, and *Motion*—forming the **World Model**.

Mission Dictionary (high level). Let $C = \{c_1, \ldots, c_N\}$ be the set of target locations (atomic symbols or *letters*). The GA–RF divides C into Q disjoint subsets $\mathscr{C}_1, \ldots, \mathscr{C}_Q$, where each subset $\mathscr{C}_q$ represents the cities assigned to UAV q. The subset is encoded as a *Mission Word* $\mathscr{W}_q^{(\text{mission})} = \{c \in \mathscr{C}_q\}$, and the set of all Mission Words within a mission forms a *Mission Phrase*. The collection of all phrases obtained from expert data constitutes the *Mission Dictionary* $\mathscr{D}_{\text{Msn}}$, modeled probabilistically as $p(\mathscr{W}^{(\text{mission})}|\mathscr{D})$.

Route Dictionary (mid level). For each Mission Word, the expert provides an ordered sequence of visits $\pi_q = [c_{q,1}, \ldots, c_{q,|\mathscr{C}_q|}]$ defining the UAV's optimal traversal order. This sequence is encoded as a *Route Word* $\mathscr{W}_q^{(\text{route})}$, and the collection of such words across missions defines the *Route Dictionary* $\mathscr{D}_{\text{Rte}}$. This level captures the probabilistic mapping $p(\mathscr{W}^{(\text{route})}|\mathscr{W}^{(\text{mission})})$, describing how high-level task divisions are realized as ordered routes.

Motion Dictionary (low level). At the motion level, each consecutive pair $(c_{q,i} \to c_{q,i+1})$ generates a trajectory segment $\gamma_{q,i}(t)$, which is transformed into a feature vector $\phi(\gamma)$ describing kinematic properties such as velocity, curvature, and heading rate. Clustering these features yields a finite alphabet of *Motion Letters*, and the concatenation of letters along a route forms a *Motion Word* that characterizes the UAV's local dynamic behavior.

Two principal motion categories are identified:

– **Attractive Motion Letters:** goal-directed segments dominated by the attractive potential

$$U_{\text{att}}(\mathbf{x};\mathbf{p}) = \tfrac{1}{2}k_{\text{att}}\|\mathbf{x}-\mathbf{p}\|^2, \tag{4.6}$$

which drives the UAV toward its target position $\mathbf{p}$.

– **Repulsive Motion Letters:** avoidance segments dominated by the repulsive potential

$$U_{\text{rep}}(\mathbf{x}) = \sum_{o\in\mathscr{O}} \frac{1}{2}k_{\text{rep}}^{(o)}\left(\max\left(0, \frac{1}{d_o(\mathbf{x})} - \frac{1}{d_0}\right)\right)^2 + \sum_{\substack{r=1\\ r\neq q}}^{Q} \frac{1}{2}k_{\text{rep}}^{(\text{uav})}\left(\max\left(0, \frac{1}{d_r(\mathbf{x})} - \frac{1}{d_0}\right)\right)^2, \tag{4.7}$$

where $\mathscr{O}$ denotes the set of obstacles, $d_o(\mathbf{x})$ and $d_r(\mathbf{x})$ are the distances from the UAV to obstacle o and to another UAV r, respectively, and d_0 is the safety distance beyond which the repulsive influence vanishes. The coefficients $k_{\text{rep}}^{(o)}$ and $k_{\text{rep}}^{(\text{uav})}$ are the corresponding repulsion gains.

The UAV motion evolves according to the gradient flow of the total potential field:

$$\dot{\mathbf{x}} = -K\nabla\big(U_{\text{att}}(\mathbf{x}) + U_{\text{rep}}(\mathbf{x})\big). \tag{4.8}$$

A trajectory segment is labeled as *attractive* or *repulsive* depending on the ratio of potential energies,

$$\rho = \frac{\int U_{\text{rep}}\,dt}{\int (U_{\text{att}} + U_{\text{rep}})\,dt}. \tag{4.9}$$

All Motion Words extracted from the demonstrations constitute the *Motion Dictionary* $\mathscr{D}_{\text{Mot}}$, whose statistical dependence on the Route Dictionary is modeled as $p(\mathscr{W}^{(\text{motion})}|\mathscr{W}^{(\text{route})})$.

Hierarchical probabilistic coupling. The three dictionaries are linked through learned transition operators $T_{\text{Msn}\to\text{Rte}}$ and $T_{\text{Rte}\to\text{Mot}}$. The overall hierarchical generative model factorizes as

$$p(\mathscr{W}^{(\text{motion})},\mathscr{W}^{(\text{route})},\mathscr{W}^{(\text{mission})}|\mathscr{D}) = p(\mathscr{W}^{(\text{mission})}|\mathscr{D})\,p(\mathscr{W}^{(\text{route})}|\mathscr{W}^{(\text{mission})})\,p(\mathscr{W}^{(\text{motion})}|\mathscr{W}^{(\text{route})}), \quad (4.10)$$

which serves as the generative prior for Active Inference during online decision-making. Equation (4.10) encodes how global mission assignments evolve into ordered routes and, finally, into dynamically feasible motion behaviors.

4.4.2 Online Decision-Making via Active Inference

During online operation, the UAV swarm continuously interprets sensory observations $o_t = \{C_t, \mathbf{X}_t\}$, where C_t denotes the set of cities (letters) and $\mathbf{X}_t = \{\mathbf{x}_c\}_{c\in C_t}$ their coordinates. The swarm relies on the World Model to infer the most plausible symbolic configuration $s_t = \{\mathscr{W}_t^{(\text{mission})},\mathscr{W}_t^{(\text{route})},\mathscr{W}_t^{(\text{motion})}\}$ and to decide subsequent actions that preserve coherence with prior knowledge while adapting to new environmental realizations. Decision-making is performed hierarchically across three action levels: mission division, route ordering, and motion generation.

The World Model provides probabilistic reference distributions learned from expert demonstrations: $p_{\text{ref}}^{(\text{Msn})} = p(\mathscr{W}^{(\text{mission})}|\mathscr{D})$, $p_{\text{ref}}^{(\text{Rte})} = p(\mathscr{W}^{(\text{route})}|\mathscr{W}^{(\text{mission})})$, $p_{\text{ref}}^{(\text{Mot})} = p(\mathscr{W}^{(\text{motion})}|\mathscr{W}^{(\text{route})})$.

At each level $\ell \in \{\text{Msn}, \text{Rte}, \text{Mot}\}$, the UAV evaluates candidate actions $a^{(\ell)}$ and selects the one minimizing the divergence between the predicted posterior and the corresponding reference distribution:

$$\mathscr{A}_\ell(a^{(\ell)}) = D_{\text{KL}}\Big(q_t(\mathscr{W}^{(\ell)}|a^{(\ell)}) \,\big\|\, p_{\text{ref}}^{(\ell)}\Big). \quad (4.11)$$

Fig. 4.3 illustrates the hierarchical dynamic bayesian network followed by the UAV swarm.

This hierarchical abnormality minimization ensures that decisions remain consistent with the statistical structure encoded in the world model. Fig. 4.4 illustrates the hierarchical Bayesian decision-making cycle followed by the UAV swarm.

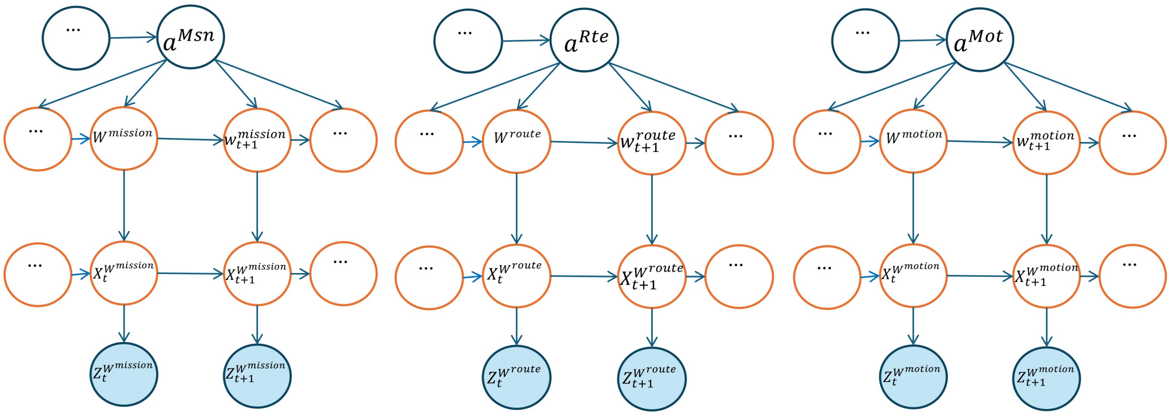


Fig. 4.3 Illustration of the candidate actions $a^{(\ell)}$ at each level.

(1) Division-level decision (Mission Words). At the highest level, the prior $p_{\text{ref}}^{(\text{Msn})}$, derived from the transition matrix $T_{D\to\text{Msn}}$, provides *reference Mission Words* representing typical partitions of cities among UAVs. A division action $a^{(\text{Msn})}$ specifies a partition $\mathscr{W}^{(\text{mission})} = \{\mathscr{C}_1,\ldots,\mathscr{C}_Q\}$, and the selected action is

$$a^{(\text{Msn})*} = \arg\min_{a^{(\text{Msn})}} D_{\text{KL}}\Big(q_t(\mathscr{W}^{(\text{mission})}|a^{(\text{Msn})}) \parallel p_{\text{ref}}^{(\text{Msn})}\Big). \tag{4.12}$$

If a new city $c^\star$ appears, the swarm determines which UAV should serve it by minimizing its divergence from the reference Mission Words:

$$q^\star = \arg\min_{q\in\{1,\ldots,Q\}} D_{\text{KL}}\Big(q_t(\mathscr{W}^{(\text{mission})} \cup \{c^\star\} \in \mathscr{C}_q) \parallel p_{\text{ref}}^{(\text{Msn})}\Big). \tag{4.13}$$

Equation (4.13) ensures that new cities are assigned to the subset producing the smallest deviation from the reference Mission Words.

(2) Ordering-level decision (Route Words). Given the selected division, the prior $p_{\text{ref}}^{(\text{Rte})}$— from $T_{\text{Msn}\to\text{Rte}}$– provides *reference Route Words* describing the expected visiting order of each UAV. For UAV q with subset $\mathscr{C}_q$, a candidate ordering action $a^{(\text{Rte})}$ defines a permutation $\mathscr{W}_q^{(\text{route})} = [c_{q,1},\ldots,c_{q,|\mathscr{C}_q|}]$, and the chosen order minimizes

$$a^{(\text{Rte})*} = \arg\min_{a^{(\text{Rte})}} D_{\text{KL}}\Big(q_t(\mathscr{W}^{(\text{route})}|a^{(\text{Rte})}) \parallel p_{\text{ref}}^{(\text{Rte})}\Big). \tag{4.14}$$

When a new city $c^\star$ is introduced within a route, it is first assigned to its subset by (4.13), and its insertion position $j^\star$ is then determined by scanning all possible positions to find

$$j^\star = \arg\min_j D_{\mathrm{KL}}\Big(q_t(\mathscr{W}^{(\mathrm{route})} \cup \{c^\star\} \text{ at } j) \parallel p_{\mathrm{ref}}^{(\mathrm{Rte})}\Big). \tag{4.15}$$

Equations (4.13) and (4.15) separately govern subset assignment and position selection, maintaining consistency with the reference Route Words.

(3) Motion-level decision (Motion Words). At the lowest level, the prior $p_{\mathrm{ref}}^{(\mathrm{Mot})}$—derived from $T_{\mathrm{Rte}\to\mathrm{Mot}}$—provides *reference Motion Words* composed of symbolic motion letters that describe short-term behaviors such as attractive or repulsive flight segments. Each motion action $a^{(\mathrm{Mot})}$ corresponds to a candidate motion word selected from the Motion Dictionary $\mathscr{D}_{\mathrm{Mot}}$. The selected motion policy minimizes

$$a^{(\mathrm{Mot})*} = \arg\min_{a^{(\mathrm{Mot})}} D_{\mathrm{KL}}\Big(q_t(\mathscr{W}^{(\mathrm{motion})}|a^{(\mathrm{Mot})}) \parallel p_{\mathrm{ref}}^{(\mathrm{Mot})}\Big), \tag{4.16}$$

ensuring that instantaneous dynamics remain coherent with the reference Motion Words. When the state estimator (e.g., EKF) predicts possible collisions or obstacles, abnormality in (4.16) increases, prompting a policy switch toward motion words with higher likelihood of repulsive motion letters.

Hierarchical integration. The three action levels are coupled through the World Model factorization:

$$p(\mathscr{W}^{(\mathrm{motion})}, \mathscr{W}^{(\mathrm{route})}, \mathscr{W}^{(\mathrm{mission})}|\mathscr{D}) = p_{\mathrm{ref}}^{(\mathrm{Msn})}\, p_{\mathrm{ref}}^{(\mathrm{Rte})}\, p_{\mathrm{ref}}^{(\mathrm{Mot})}. \tag{4.17}$$

The global objective integrates the level-wise divergence measures:

$$\mathscr{A}_{\mathrm{total}} = \mathscr{A}_{\mathrm{Msn}} + \mathscr{A}_{\mathrm{Rte}} + \mathscr{A}_{\mathrm{Mot}}, \quad \mathscr{A}_\ell = D_{\mathrm{KL}}\big(q_t(\mathscr{W}^{(\ell)}|a^{(\ell)}) \big\| p_{\mathrm{ref}}^{(\ell)}\big). \tag{4.18}$$

Minimizing (4.18) yields a globally consistent inference process across all abstraction levels, enabling the swarm to reorganize mission assignments, update route orders, and refine motion behaviors in response to novel environmental realizations while maintaining coherence with the reference Mission, Route, and Motion Words encoded in the world model.

4.5 Results and Analysis

The evaluation of the proposed active inference-based framework consists of two parts:

The first part is based on the evaluation of the proposed method with purely simulated data, while the second part presents an analysis of the model performance by incorporating data collected from real drone flights into the simulation.

The performance of the proposed Active Inference–based framework is evaluated in a fully online UAV swarm trajectory design setting. Multiple UAVs operated in a 1000×1000 m region at a constant altitude of 200 m. During training, 50 towns were randomly placed within the area, and $M = 5000$ mission instances were generated using the GA–RF optimizer. Each instance provided optimal velocity and visiting-order sequences, which were encoded into symbolic words to construct the world model. The testing phase used new random subsets of towns, relying solely on the learned model for decision-making. A modified Q-learning (QL) method trained on the same data served as a baseline. So that a comparative analysis of the results can be made in terms of performance, time, and collision avoidance.

To evaluate the effectiveness of the framework in real-world scenarios, real-time data were collected from flight experiments conducted at Universidad Carlos III de Madrid (UC3M). These experiments were carried out solely for data generation purposes, so that the collected data could later be incorporated into simulations to test the proposed model. During the experiments, two DJI Air 2S drones were used, each controlled by human operators (pilots) via Wi-Fi. The experiments took place in UC3M's internal laboratory, where the drones flew between specific target points, providing sensor data related to position, velocity, and angular motion (yaw, pitch, and roll).

Drone Platform and Sensors:

- Front-facing RGB camera (3840×2160 pixels, 30 Hz frame rate).
- Two IMU units (accelerometer, gyroscope, magnetometer).
- Automatic position and orientation estimation via visual inertial odometry (VIO).

Indoor Scenery and Flight Configuration: 10 fixed points were used as targets, and a central "depot" location was the start and end point for all flights. Each drone was operated by a separate pilot, and the pilots were instructed to fly over all targets at least once. The flight altitude was set at 10 to 15 feet, and only one drone was allowed to fly over each target to avoid path interference.

The model's performance is tested against real observations by inputting sensor data from these flights into a simulation environment. This analysis is conducted to determine whether the model maintained the same stability, decision-making, and consistency in uncertain real data that it demonstrated in the simulated scenarios.

(1) Performance evaluation on simulated data:

Fig. 4.5 and 4.6 shows the multi-level behavior of the proposed framework. At the high and mid levels (Fig. 4.5), Active Inference determines mission division and route ordering consistent with the reference Mission and Route Words from the world model. At the low level (Fig. 4.6), UAVs execute local motions between towns using symbolic motion words representing attractive and repulsive flight behaviors. This illustrates coherent hierarchical planning where global objectives are translated into dynamically feasible local trajectories.

When a new target unexpectedly appears, Active Inference detects the resulting prediction–action mismatch and triggers belief revision. Fig. 4.7 depicts the "surprise" region, and Fig. 4.8 shows the updated trajectory after belief correction. The UAV autonomously selects the new path that minimizes abnormality (divergence) from the reference world model, achieving adaptive trajectory re-optimization in real time.

To enhance motion consistency and safety, an EKF module was integrated for local state estimation. Fig. 4.9a shows EKF-based path prediction, enabling smooth, goal-directed motion correction. As demonstrated in Fig. 4.9b, the EKF further supports static and dynamic obstacle avoidance by fusing sensor feedback with predicted motion states, ensuring collision-free navigation through adaptive repulsion control.

Fig. 4.10a shows that the PF not only performs belief updating within the Active Inference framework, but also enables reactive motion adaptation. In this way, the UAV adapts its behavior in real time according to its model predictions, which significantly improves both the safety and adaptability of the system.

Furthermore, Fig. 4.10b shows that the UAV maintains a safe trajectory even in the presence of static obstacles, using its predictive motion model and sensor data.

Fig. 4.11 compares trajectories produced by the proposed method and modified QL. Active Inference achieves closer alignment with expert demonstrations, maintaining belief–action consistency across mission, route, and motion levels, whereas QL exhibits deviations at the motion level. Fig. 4.12 compares mission completion time and total travel distance. The proposed framework consistently outperforms both GA–RF and QL, producing shorter, smoother, and more energy-efficient paths. This confirms that learning from expert demonstrations through a probabilistic world model enables the UAV to generalize underlying optimization principles rather than merely replicate solutions.

Fig. 4.13 illustrates the similarity ratios between the trajectories generated by the AI and those produced by the GA-RF Optimizer, as well as between the Modified-QL method and the GA-RF Optimizer. Two types of trajectories are analyzed: (1) task division and

(2) visiting order. The results indicate that the proposed method produces trajectories more closely aligned with those of the GA-RF Optimizer, suggesting that it has partially learned the optimizer's strategy and can indirectly infer the underlying objective function.

Overall, the results demonstrate that the proposed method yields faster convergence, higher stability, and more adaptive performance than QL, validating its capability for autonomous, probabilistic reasoning and real-time re planning in complex environments.

(2) Performance Evaluation Using Real-Time Experimental Data:

Fig. 4.14-a shows the trajectories of two UAVs obtained from real flights controlled by pilots. These experimental flights were conducted to provide real-world data for the model so that this data could later be used for testing and validation of the model.

Fig. 4.14-b presents the combined transition matrix generated based on the cluster labels obtained from different velocities. In this process, unsupervised clustering was first performed using the Growing Neural Gas (GNG) algorithm, which divided the velocities into different groups. Later, a transition matrix was created based on the cluster labels from the same GNG, which shows the transition probabilities of the UAVs between different dynamic states. This transition matrix is later used in the prediction phase to estimate the possible future behavior and direction of movement of the UAVs.

Fig. 4.15 shows the trajectory prediction results and the associated errors from the model. In this experiment, we tested a new trajectory using the trained model.

The first part of the Fig. 4.15 shows the actual cluster labels at each time instant, which represent the observed data. The middle part shows the predicted labels from the model, which indicate how accurately the model identified different clusters or movements during the prediction. The bottom part of the Fig. 4.15 shows the prediction errors at the cluster level, which represent the difference between the actual and predicted labels at each time instant.

These results confirm that before conversion, the model is not able to make correct predictions and the prediction error is very high.

Fig. 4.16 presents the results obtained after full convergence of the proposed Bayesian inference model.

Fig. 4.16-(left) shows the actual 3D trajectory, with clusters represented by different colors. This trajectory is based on actual observations and was used as a reference for the model.

Fig. 4.16-(right) shows the predicted trajectory by the model. The red circles highlight the locations where the model made corrections or inference adjustments during its estimation process. These results demonstrate that the proposed model not only learns effectively from the observation data but also has the ability to correct errors during prediction compared to

the actual trajectory. This feature highlights the self-corrective nature of the model and its ability to support online learning.

Fig. 4.17 shows a comparison between actual clusters and predicted clusters at each time instance. Furthermore, the figure also includes prior cluster knowledge to illustrate how the model used the available prior information in the cluster prediction process.

The figure highlights that when the model corrected the predictions using Bayesian inference, the error is significantly reduced.

This observation shows that the model not only predicts the clusters initially but also has the ability to improve its predictions over time, producing results that are closer to the true labels.

4.6 Summary

This chapter presented an Active Inference–based framework for self-learning UAV swarm trajectory design. The proposed method learns a probabilistic world model from expert demonstrations and employs it online for hierarchical decision-making across mission, route, and motion levels. By continuously updating beliefs and minimizing divergence from expected outcomes, UAVs achieve adaptive, energy-efficient, and goal-directed coordination. An EKF–assisted module enhances local state estimation and ensures smooth, collision-free navigation in dynamic environments. The results of this chapter consist of two main parts.The first part is based on the performance analysis of the proposed method with purely simulated data. The results show that the proposed framework demonstrated faster convergence, improved stability, and enhanced generalization compared to Modified Q-Learning. These results confirm that the proposed method provides a scalable, secure, and cognitively robust control framework for UAV swarms, which provides a solid foundation for future AI-driven aerial networks. The second part consists of experiments in which data collected from real drone flights were included in the simulation to test the model's performance under real observational conditions. These results demonstrate that the proposed model not only learns effectively from observational data but also has the ability to correct errors during prediction. This feature highlights the self-corrective nature of the model and its ability to learn online, which promotes the potential for more autonomous and intelligent decision-making for UAV swarms.

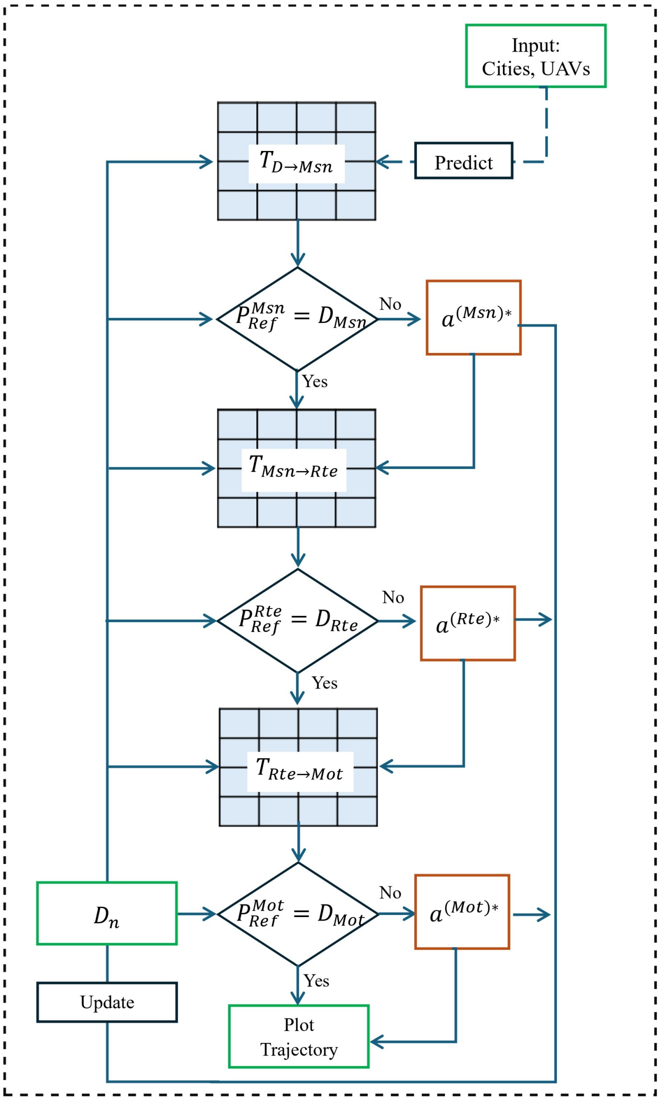


Fig. 4.4 Hierarchical decision-making across three levels: mission division, route order, and motion level decision.

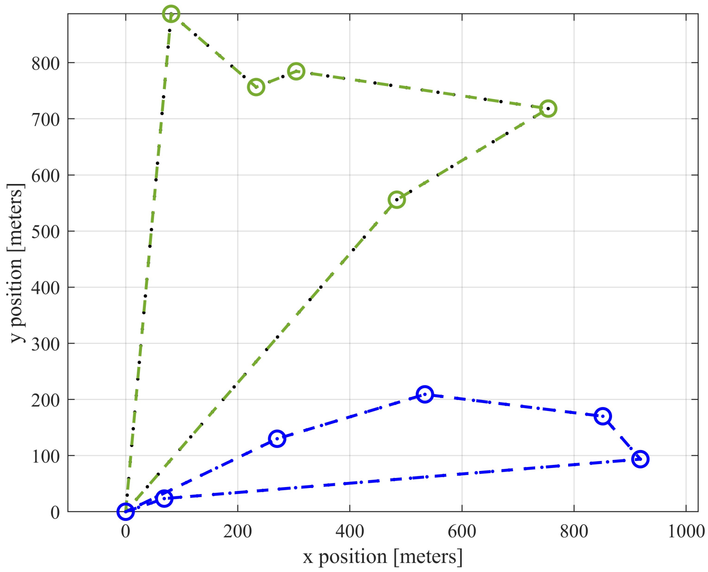


Fig. 4.5 Active Inference–based UAV swarm trajectories. (High- and mid-level mission division and route ordering.)

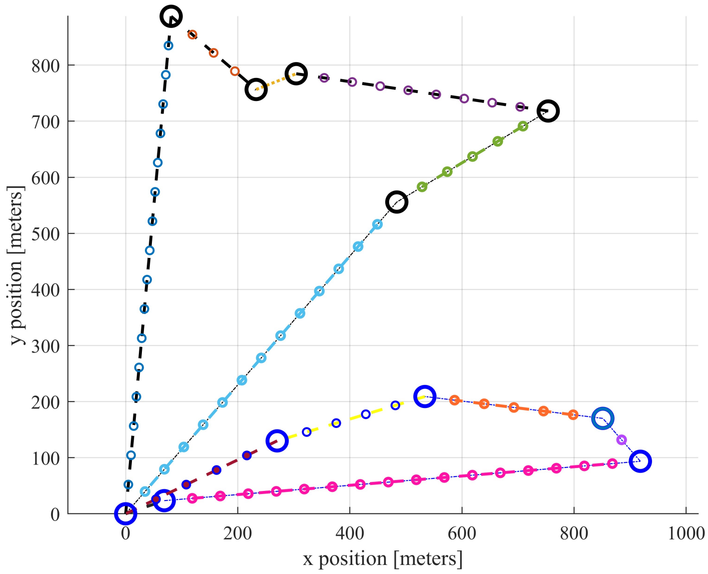


Fig. 4.6 Active inference–based UAV swarm trajectories. (Low-level motion execution using learned motion words.)

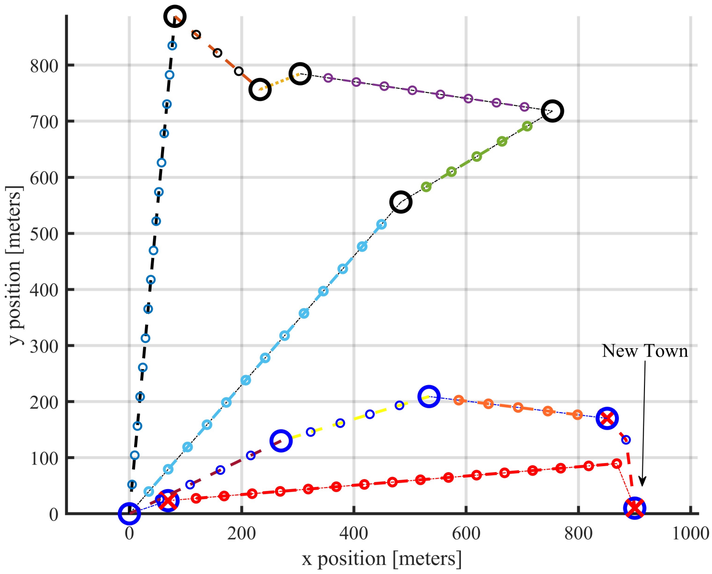


Fig. 4.7 Adaptive re-planning under environmental changes (a new target introduces surprise;)

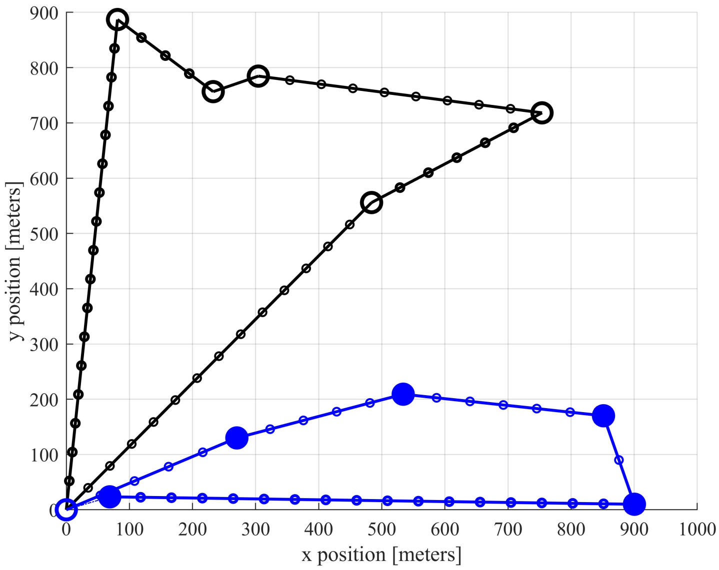
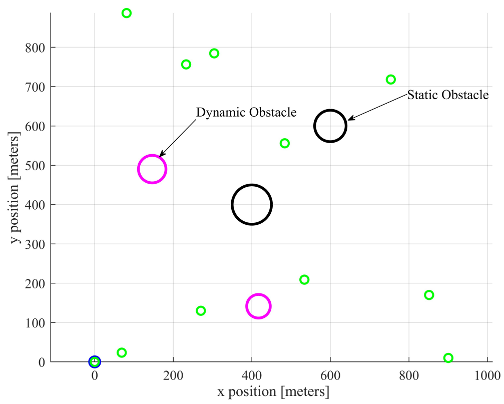
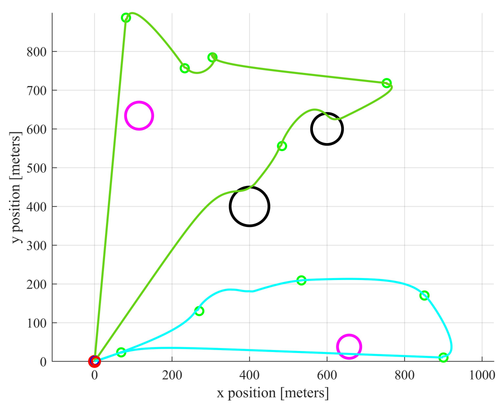


Fig. 4.8 Adaptive re-planning under environmental changes (belief updating and trajectory correction restore mission consistency.)

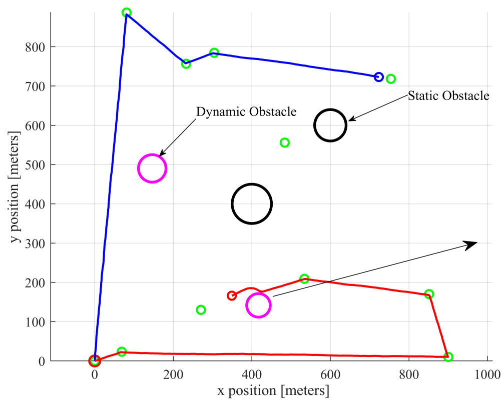


(a) Real-time trajectory prediction and correction.

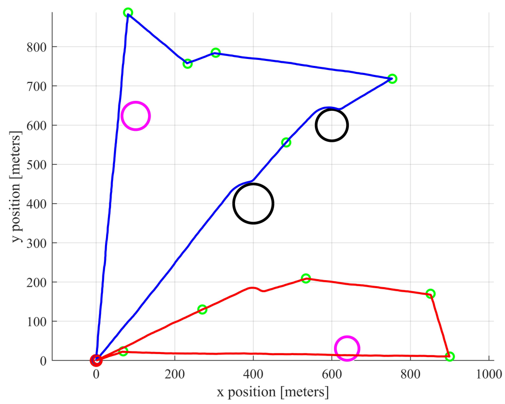


(b) Avoidance of static and dynamic obstacles.

Fig. 4.9 EKF-assisted state estimation and collision avoidance: (a) real-time trajectory prediction and correction; (b) avoidance of static and dynamic obstacles.

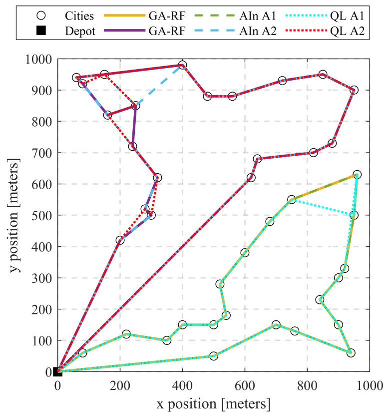


(a) Real-time trajectory prediction and correction.

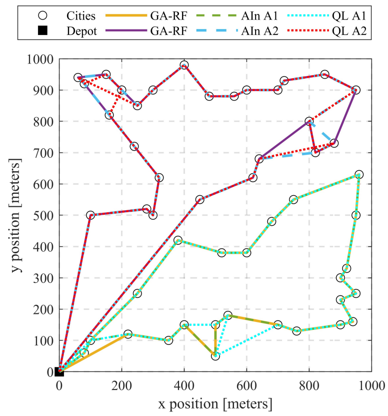


(b) Avoidance of static and dynamic obstacles.

Fig. 4.10 PF-assisted state estimation and collision avoidance: (a) real-time trajectory prediction and correction; (b) avoidance of static and dynamic obstacles.

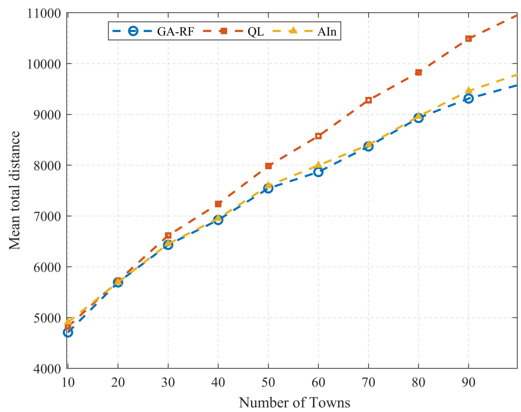


(a) 40 towns.

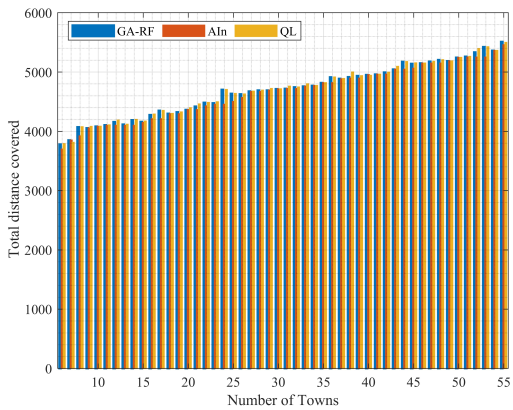


(b) 50 towns.

Fig. 4.11 Qualitative comparison of Active Inference and Modified Q-Learning: Active Inference maintains belief–action consistency and smoother paths, while Q-Learning exhibits deviations from model-based trajectories.

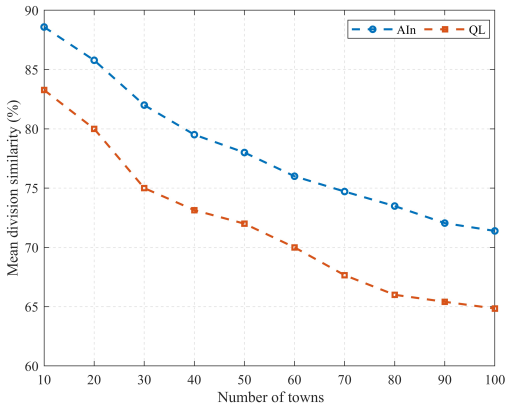


(a) Mission completion time.

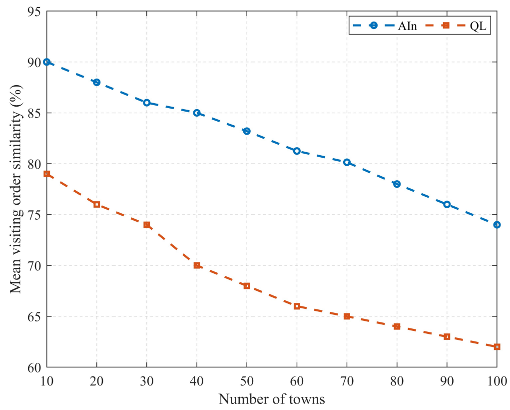


(b) Total covered distance.

Fig. 4.12 Quantitative performance comparison: (a) mission completion time and (b) total distance, showing improved efficiency over GA–RF and Modified Q-Learning.

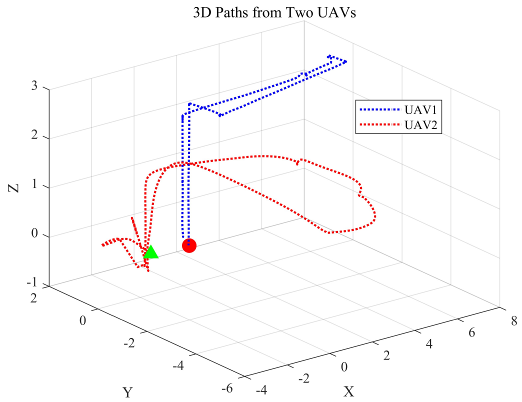


(a) Division similarity

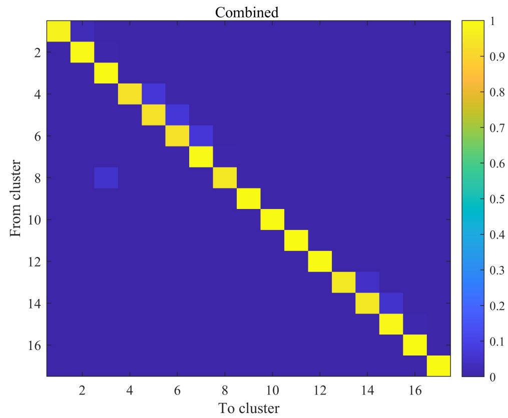


(b) Visiting order similarity.

Fig. 4.13 Comparison of completion time and total distance covered by the GA-RF Optimizer, AI, and Modified-QL for different numbers of towns.



(a) Example trajectory of 2 UAVs



(b) Combine transition matrix of multiple trajectories.

Fig. 4.14 (a).Example of trajectory generated by 2 UAVs (b).Combining transition matrices of cluster labels from multiple trajectories

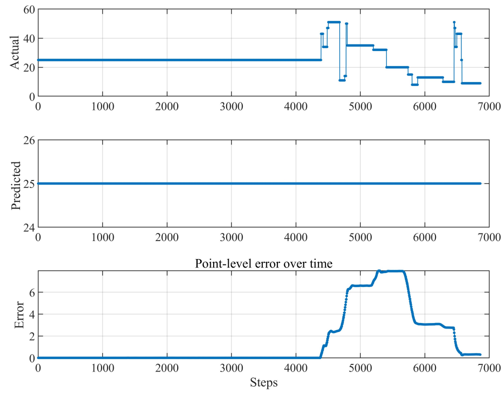


Fig. 4.15 Trajectory prediction and prediction errors before convergence. The (First) original cluster labels, the (Middle) model predicted labels, and the (Bottom) prediction errors at each cluster level.)

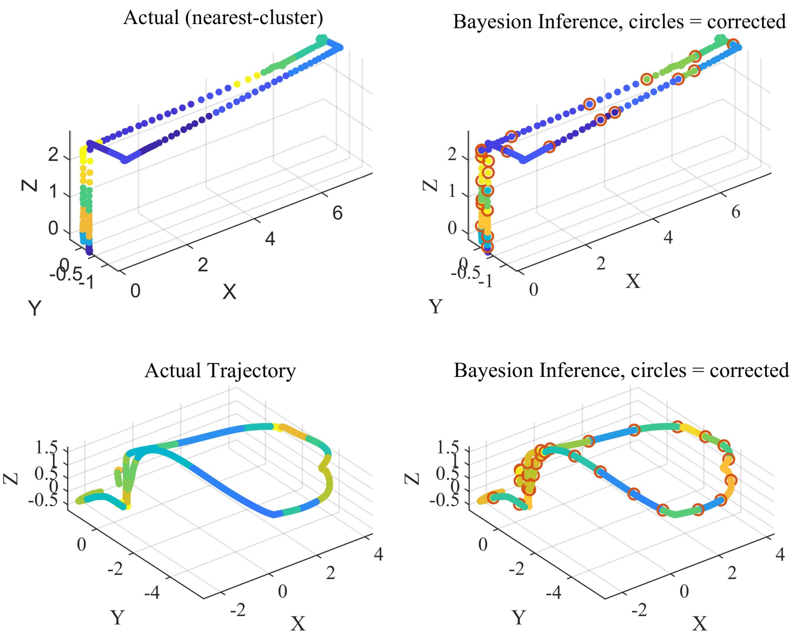


Fig. 4.16 Results of the proposed model based on bayesian inference, (left) the original 3D trajectory with different cluster colors, and (right) the predicted trajectory with red circles indicating the locations where the model corrected the prediction.

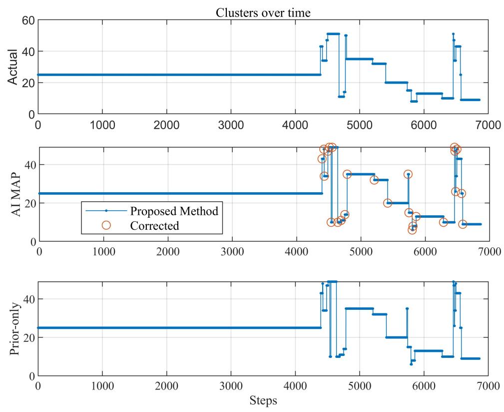


Fig. 4.17 Comparison of the original, predicted, and prior clusters at each time point. The prediction errors are minimized after bayesian correction.

Chapter 5

Conclusion and Future Directions

5.1 Conclusion

This thesis presents a comprehensive, self-learning, and knowledge-driven approach for UAV swarm trajectory design, which begins with data generation through biologically inspired (GA-RF) method and gradually extends to the construction of a causal probabilistic world model, followed by online decision-making based on active inference. This approach not only enables UAVs to self-learn and continuously update their beliefs based on observations from the environment but also allows them to make effective decisions in real time according to changing conditions. The research process consists of the following sections:

In Chapter 2, the study highlights the differences between traditional, biologically inspired, metaheuristic, and modern AI-based strategies and their impact on UAV swarm missions.

- The results show that traditional algorithms provide a strong foundation in structured and static environments, but their performance is limited in dynamic and uncertain situations.

- Biologically inspired methods have achieved significant success in global optimization and solution diversity, but challenges such as convergence speed and computational complexity remain.

- On the other hand, modern strategies based on artificial intelligence, especially DRL, MARL, and Active Inference, have revealed new possibilities for autonomous decision-making, decentralized control, and real-time path adaptation within UAV swarms, but these approaches face practical problems such as data availability and computational burden.

- Consequently, this chapter emphasizes that a hybrid framework combining the reliability of TAs, the global search capability of BIAs, and the real-time adaptability of AI techniques is an effective future direction for UAV swarm navigation. Moreover, transparent decision-making theories such as Explainable AI and Interpretable RL are essential for building trust and stability in future intelligent UAV systems.

Chapter 3 introduces a new approach MTSP–GA–RF for UAV swarm trajectory design, which is based on genetic algorithms and repulsion forces.

- This method optimizes the routes of UAVs in the MTSP domain to reduce collisions, overlaps, and interference, ensuring that each UAV serves its assigned areas effectively and in a balanced manner.
- Furthermore, a size-optimization mechanism was developed to determine the ideal size for the UAV swarm that minimizes both time and distance.
- Compared with algorithms such as MTSP–GA, PSO, SA, and ACO in different scenarios, this method was found to be more efficient in terms of distance, time, and energy consumption. These results show that the proposed GA–RF method not only outperforms classical metaheuristic methods but is also more applicable to real operational environments.
- The optimized paths obtained from this framework were subsequently used as expert demonstrations to train a high-level decision-making world model based on Active Inference, paving the way for goal-directed and autonomous trajectory planning for UAV swarms.

In Chapter 4, a self-learning Active Inference–based framework was presented that enables real-time decision-making and online adaptation for UAV swarms.

- The proposed model created a causal probabilistic world model based on expert knowledge obtained from GA–RF, reflecting learned behaviors at the mission division, ordering, and movement levels.
- The model enabled UAVs to adopt energy-efficient, adaptive, and goal-oriented behaviors by continuously updating their beliefs with expected outcomes.
- To improve real-time accuracy and collision avoidance, an EKF- and PF-based auxiliary module was integrated, ensuring smooth and safe navigation in dynamic environments.

- The experiments consisted of two stages: the first stage analyzed the convergence, stability, and generalization of the framework using simulated data, while the second stage evaluated the model's performance by incorporating real-time data from drone flights into the simulation.
- The results show that the proposed framework achieved faster convergence, better stability, and significant improvements in generalization compared to Modified Q-Learning. Furthermore, the experiments based on real data highlighted the self-corrective feature of the model, where prediction errors were reduced through Bayesian correction, and the model produced results closer to the actual trajectories.
- This framework provides a scalable, secure, and cognitively robust solution for UAV swarms that can serve as a reliable foundation for future AI-based aerial networks.

Overall, this research introduces a clear evolutionary direction in UAV swarm trajectory design by integrating artificial intelligence and functional reasoning, moving beyond traditional and biological models. The overall research framework provides UAVs with the ability to self-learn, make explainable decisions, and adapt in real time, which not only improves system performance and stability but also ensures energy conservation and collision avoidance. Furthermore, the use of data obtained from real-world experiments brings this research closer to practical application, representing a strong step toward the deployment of UAV networks.

5.2 Future Directions

The results of this research and the proposed framework provide a solid foundation for self-learning, active reasoning, and knowledge-based decision-making for UAV swarms. However, the success of this work also opens the door to several new research directions and practical improvements that can be expanded upon in the future.

- First, making the proposed model more explainable and interpretable is an important next step. Since the internal dynamics of active reasoning systems are often complex and opaque, future work needs to develop interpretable architectures and post-hoc explanation tools that help researchers and practitioners better understand the model's decisions and predictions. For this purpose, visualization techniques such as dynamic mapping of neural state space, visual representations of Bayesian belief progressions, or interactive graphs of policy-decision paths can be useful, so that the decision-making process of each UAV is more transparent and can be integrated with human knowledge and experience.

- Second, to improve the generalizability of the model, it is necessary to test it in more diverse and realistic scenarios. This includes testing and applying the model on diverse datasets obtained from different geographical regions, weather conditions, and operational scenarios. In this context, the inclusion of multi-sensor data in future experiments will be crucial, particularly the use of LiDAR, video cameras, and infrared (IR) sensors to obtain more in-depth and detailed observations for UAVs. By incorporating data from these sensors into simulations, it will be possible to test the accuracy, robustness, and sensor-fusion capability of the model, which will further enhance the generalizability and reliability of the model in diverse real-world situations.

- Third, it is important to expand the experiments to larger fleets to observe the scalability, robustness, and emergent swarm behaviors of the system. In this regard, experiments involving hundred or more UAVs can help discover new principles of automatic coordination, collective decision-making, and energy distribution. These experiments will show how effectively the system copes with real-world problems such as latency, signal collisions, and communication pressure in large networks.

- Furthermore, making the architecture fully decentralized is a key future direction to improve system performance. To this end, it will be necessary to incorporate consensus algorithms, distributed belief propagation, and fault-tolerant communication mechanisms into the active reasoning framework so that each UAV can make its own decisions based on local observations, but still be consistent with the global goal. Such a fully decentralized system will not only increase autonomy but also reduce the risk of central control deviation or failure.

- In addition, it is also important to make inter-UAV communication more robust and secure. Future work could develop low-latency, reliable, and secure communication protocols that can also operate in electronic interference or contested environments. Shared bandwidth sharing, code division schemes, and frequency-hopping techniques can ensure timely transmission of information within the swarm in such situations, while Bayesian prediction modules will help predict communication failures and identify alternative routes.

- Furthermore, implementing and testing the proposed framework on real UAV platforms is also a key next step. Real-time experiments will provide an opportunity to test the model's practical performance, latency effects, system stability, and safe navigation requirements. This process will enable a better understanding of the simulation-to-reality gap, which will allow the model to be optimized for practical operations.

- The integration of jammer and blockage prediction modules to protect against radio frequency interference (RF interference) and physical obstructions is also crucial in future work. These modules will predict potential interference or blockage based on observational data, signal strength, and historical patterns, allowing the system to adjust its communication and control strategy in advance. In this way, the UAV swarm will not only be more autonomous in changing conditions but also gain a better ability to protect against dynamic threats.

Ultimately, expanding research in all these directions will not only scientifically provide new insights into the functional reasoning models of UAV swarms, but also practically lay the foundation for building safe, autonomous, and intelligent aerial networks. Such networks could not only revolutionize logistics, industrial, and defense operations in the future, but also help implement advanced concepts of real-time collective learning, self-awareness and decision-making.

References

[1] Abualigah, L., Yousri, D., Abd Elaziz, M., Ewees, A. A., Al-Qaness, M. A., and Gandomi, A. H. (2021). Aquila optimizer: a novel meta-heuristic optimization algorithm. *Computers & Industrial Engineering*, 157:107250.

[2] Abujabal, N., Fareh, R., Sinan, S., Baziyad, M., and Bettayeb, M. (2023). A comprehensive review of the latest path planning developments for multi-robot formation systems. *Robotica*, 41(7):2079–2104.

[3] Agrawal, S., Patle, B. K., and Sanap, S. (2024). A systematic review on metaheuristic approaches for autonomous path planning of unmanned aerial vehicles. *Drone Systems and Applications*, 12:1–28.

[4] Ahmad, F., Mirza, M. Y., Hussain, I., and Arshid, K. (2025). A Multi-Ray Channel Modelling Approach to Enhance UAV Communications in Networked Airspace. *Inventions*, 10(4):51.

[5] Ajith, V. and Jolly, K. (2023). Hybrid optimization based multi-objective path planning framework for unmanned aerial vehicles. *Cybernetics and Systems*, 54(8):1397–1423.

[6] Akay, B. and Karaboga, D. (2009). A modified artificial bee colony algorithm for real-parameter optimization. *Information Sciences*, 192:120–142.

[7] Alabbadi, A. J. and Sababha, B. H. (2025). On the optimization of uav swarm aco-based path planning. *Jordanian Journal of Computers and Information Technology (JJCIT)*, 11(03).

[8] Alam, M. M., Trina, S. A., Hossain, T., Mahmood, S., Ahmed, M. S., and Arafat, M. Y. (2025). Variations in multi-agent actor–critic frameworks for joint optimizations in uav swarm networks: Recent evolution, challenges, and directions. *Drones*, 9(2):153.

[9] Aljalaud, F., Kurdi, H., and Youcef-Toumi, K. (2023). Bio-inspired multi-uav path planning heuristics: A review. *Mathematics*, 11(10):2356.

[10] Alqefari, S. and Menai, M. E. B. (2025). A hybrid method to solve the multi-uav dynamic task assignment problem. *Sensors*, 25(8):2502.

[11] Alqudsi, Y. and Makaraci, M. (2025). Uav swarms: Research, challenges, and future directions. *Journal of Engineering and Applied Science*, 72(1):12.

[12] AlShabi, M., Ballous, K. A., Nassif, A. B., Bettayeb, M., Obaideen, K., and Gadsden, S. A. (2024). Path planning for a ugv using salp swarm algorithm. In *Autonomous Systems: Sensors, Processing, and Security for Ground, Air, Sea, and Space Vehicles and Infrastructure 2024*, volume 13052, pages 151–159. SPIE.

[13] An, Y., Liu, A., Liu, H., and Geng, L. (2024). Multidimensional trajectory prediction of uav swarms based on dynamic graph neural network. *IEEE access*, 12:57033–57042.

[14] Arnold, R., Mezzacappa, E., Jablonski, M., Jablonski, J., and Abruzzo, B. (2021). Performance comparison of decentralized undirected swarms versus centralized directed swarms at different levels of quality of knowledge. In *2021 IEEE International Symposium on Technologies for Homeland Security (HST)*, pages 1–9. IEEE.

[15] Arranz, R., Carramiñana, D., de Miguel, G., Besada, J. A., and Bernardos, A. M. (2025). Application of deep reinforcement learning to uav swarming for ground surveillance. *arXiv preprint arXiv:2501.08655*.

[16] Arshad, M. A., Ahmed, J., and Bang, H. (2023). Quadrotor path planning and polynomial trajectory generation using quadratic programming for indoor environments. *Drones*, 7(2):122.

[17] Arshad, U. and Halim, Z. (2025). Secure and optimized drone swarm operations with decentralized adaptive differential evolution. *Computers and Electrical Engineering*, 126:110487.

[18] Arshid, K., Krayani, A., Marcenaro, L., Gomez, D. M., and Regazzoni, C. (2025a). Toward autonomous uav swarm navigation: A review of trajectory design paradigms. *Sensors*, 25(18):5877.

[19] Arshid, K., Krayani, A., Marcenaro, L., Gomez, D. M., and Regazzoni, C. (2025b). Uav swarm trajectory design for wireless networks using genetic algorithm-driven repulsion forces. *IEEE Access*.

[20] Arulkumaran, K., Deisenroth, M. P., Brundage, M., and Bharath, A. A. (2017). Deep reinforcement learning: A brief survey. *IEEE Signal Processing Magazine*, 34(6):26–38.

[21] Awadallah, M. A., Makhadmeh, S. N., Al-Betar, M. A., Dalbah, L. M., Al-Redhaei, A., Kouka, S., and Enshassi, O. S. (2025). Multi-objective ant colony optimization. *Archives of Computational Methods in Engineering*, 32(2):995–1037.

[22] Badar, T., Särkkä, S., Zhao, Z., and Visala, A. (2024). Rao–blackwellized particle filter using noise adaptive kalman filter for fully mixing state-space models. *IEEE Transactions on Aerospace and Electronic Systems*, 60(5):6972–6982.

[23] Balanji, H. M. and Yanmaz, E. (2024). Priority-based dynamic multi-uav positioning for multi-target search and connectivity. In *2024 IEEE Wireless Communications and Networking Conference (WCNC)*, pages 1–6. IEEE.

[24] Bektas, T. (2006). The multiple traveling salesman problem: an overview of formulations and solution procedures. *Omega*, 34(3):209–219.

[25] Bello-Orgaz, G., Ramirez-Atencia, C., Fradera-Gil, J., and Camacho, D. (2015). Gampp: Genetic algorithm for uav mission planning problems. In *Intelligent Distributed Computing IX: Proceedings of the 9th International Symposium on Intelligent Distributed Computing–IDC'2015, Guimarães, Portugal, October 2015*, pages 167–176. Springer.

[26] Bolognini, M., Fagiano, L., and Limongelli, M. P. (2023). A fault-tolerant automatic mission planner for a fleet of aerial vehicles. *Control Engineering Practice*, 135:105501.

[27] Bu, Y., Yan, Y., and Yang, Y. (2024). Advancement challenges in uav swarm formation control: A comprehensive review. *Drones*, 8(7):320.

[28] Busoniu, L., Babuska, R., De Schutter, B., and Ernst, D. (2008). A comprehensive survey of multiagent reinforcement learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 38(2):156–172.

[29] Cao, Y. and Nor, N. M. (2024). An improved dynamic window approach algorithm for dynamic obstacle avoidance in mobile robot formation. *Decision Analytics Journal*, 11:100471.

[30] Chai, Y., Zhang, Z., Yu, H., Han, J., Fang, Y., and Liang, X. (2025). A trajectory planning scheme for collaborative aerial transportation systems by graph-based searching and cable tension optimization. *IEEE/ASME Transactions on Mechatronics*.

[31] Chandan, R. R., Soni, S., Raj, A., Veeraiah, V., Dhabliya, D., Pramanik, S., and Gupta, A. (2023). Genetic algorithm and machine learning. In *Advanced Bioinspiration Methods for Healthcare Standards, Policies, and Reform*, pages 167–182. IGI Global.

[32] Chang, X., Chen, X., Liu, Z., Chen, Z., Wang, Q., and Liu, X. (2025). Research on multi-uav autonomous obstacle avoidance algorithm integrating improved dynamic window approach and orca. *Scientific Reports*, 15(1):14646.

[33] Chen, L. (n.d.). *UAV Path Planning and Obstacle Avoidance Based on Fuzzy Logic and Kinodynamic RRT Methods*. Doctoral dissertation, Concordia University. Available from Concordia University Library.

[34] Chen, Y., Chen, R., Huang, Y., Xiong, Z., and Li, J. (2024). Drl-based improved uav swarm control for simultaneous coverage and tracking with prior experience utilization. *Drones*, 8(12):784.

[35] Cheng, Z., Zhang, H., and Guo, L. (2023). Multi-uav cooperative task planning based on an improved adaptive simulated annealing and genetic algorithm. In *Third International Conference on Advanced Algorithms and Neural Networks (AANN 2023)*, volume 12791, pages 144–153. SPIE.

[36] Cheng, Z., Zhao, L., and Shi, Z. (2022). Decentralized multi-uav path planning based on two-layer coordinative framework for formation rendezvous. *IEEE Access*, 10:45695–45708.

[37] Dang, A. D., La, H. M., Nguyen, T., and Horn, J. (2019). Formation control for autonomous robots with collision and obstacle avoidance using a rotational and repulsive force–based approach. *International Journal of Advanced Robotic Systems*, 16(3):1729881419847897.

[38] Dasgupta, A., Zope, V., and Ismail, A. (2025). Implementation of the bees algorithm for uav mission plan. *Engineering Headway*, 13:11–18.

[39] De Sá, D. F. S. and Neto, J. V. D. F. (2023). Multi-agent collision avoidance system based on centralization and decentralization control for uav applications. *IEEE Access*, 11:7031–7042.

[40] Debnath, D., Vanegas, F., Sandino, J., Hawary, A. F., and Gonzalez, F. (2024). A review of uav path-planning algorithms and obstacle avoidance methods for remote sensing applications. *Remote Sensing*, 16(21):4019.

[41] Deng, L., Chen, H., Zhang, X., and Liu, H. (2023). Three-dimensional path planning of uav based on improved particle swarm optimization. *Mathematics*, 11(9):1987.

[42] Dharmaraj, R., Kumar, P., and Iqbal, M. (2019). Collision-free path planning for uavs using improved genetic algorithms. *IEEE Access*, 7:110123–110135.

[43] Dhulkefl, E., Durdu, A., and Terzioğlu, H. (2020). Dijkstra algorithm using uav path planning. *Konya Journal of Engineering Sciences*, 8:92–105.

[44] Dorling, K., Heinrichs, J., Messier, G., and Magierowski, S. (2017). Vehicle routing problems for drone delivery. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 47(1):70–85.

[45] Du, Y. (2023). Multi-uav search and rescue with enhanced a* algorithm path planning in 3d environment. *International Journal of Aerospace Engineering*, 2023(1):8614117.

[46] Dubins, L. E. (1957). On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents. *American Journal of Mathematics*, 79(3):497–516.

[47] Ekechi, C. C., Elfouly, T., Alouani, A., and Khattab, T. (2025). A survey on uav control with multi-agent reinforcement learning. *Drones*, 9(7):484.

[48] Elfring, J., Torta, E., and Van De Molengraft, R. (2021). Particle filters: A hands-on tutorial. *Sensors*, 21(2):438.

[49] Fei, C., Lu, Z., and Jiang, W. (2024). Heuristic optimization-based trajectory planning for uav swarms in urban target strike operations. *Drones*, 8(12):777.

[50] Fox, D., Burgard, W., and Thrun, S. (2002). The dynamic window approach to collision avoidance. *IEEE Robotics & Automation Magazine*, 4(1):23–33.

[51] Friston, K., FitzGerald, T., Rigoli, F., Schwartenbeck, P., and Pezzulo, G. (2017). Active inference: a process theory. *Neural Computation*, 29(1):1–49.

[52] Friston, K., Parr, T., and Pezzulo, G. (2021). The free energy principle: a unified brain theory? *Nature Reviews Neuroscience*, 22(2):125–138.

[53] Fu, H., Li, Z., Zhang, W., Feng, Y., Zhu, L., Fang, X., and Li, J. (2024). Research on path planning of agricultural uav based on improved deep reinforcement learning. *Agronomy*, 14(11):2669.

[54] Gad, A. G. (2022). Particle swarm optimization algorithm and its applications: a systematic review. *Archives of computational methods in engineering*, 29(5):2531–2561.

[55] Gao, W. and Li, Y. (2021). Search-based algorithms for uav path planning: A comprehensive review. *Applied Sciences*, 11(17):8234.

[56] Gao, X., Zhang, Y., Wang, B., Leng, Z., and Hou, Z. (2024). The optimal strategies of maneuver decision in air combat of ucav based on the improved td3 algorithm. *Drones*, 8(9):501.

[57] Gasparetto, A., Boscariol, P., Lanzutti, A., and Vidoni, R. (Dec. 2015). Path planning and trajectory planning algorithms: A general overview. In *Motion and operation planning of robotic systems: Background and practical approaches*, pages 3–27. Springer, Cham.

[58] Ge, F., Wei, Y., Yu, W., and Li, J. (2020). Path planning of uav for oilfield inspections in a three-dimensional dynamic environment with moving obstacles based on an improved pigeon-inspired optimization algorithm. *Applied Intelligence*, 50(9):2800–2817.

[59] Ghazali, M. H. M., Teoh, K., and Rahiman, W. (2021). A systematic review of real-time deployments of uav-based lora communication network. *IEEE Access*, 9:124817–124830.

[60] Ghdiri, O., Jaafar, W., Alfattani, S., Abderrazak, J. B., and Yanikomeroglu, H. (2021). Offline and online uav-enabled data collection in time-constrained iot networks. *IEEE Transactions on Green Communications and Networking*, 5(4):1918–1933.

[61] Gopalakrishnan, S. K., Al-Rubaye, S., and Inalhan, G. (2021). Adaptive uav swarm mission planning by temporal difference learning. In *2021 IEEE/AIAA 40th Digital Avionics Systems Conference (DASC)*, pages 1–10. IEEE.

[62] Goricanec, J., Milas, A., Markovic, L., and Bogdan, S. (2023). Collision-free trajectory following with augmented artificial potential field using uavs. *IEEE access*.

[63] Grewal, M. S. and Andrews, A. P. (2010). Applications of kalman filtering in aerospace 1960 to the present [historical perspectives]. *IEEE Control Systems Magazine*, 30(3):69–78.

[64] Grondman, I., Busoniu, L., Lopes, G. A., and Babuska, R. (2012). A survey of actor-critic reinforcement learning: Standard and natural policy gradients. *IEEE Transactions on Systems, Man, and Cybernetics*, 42(6):1291–1307.

[65] Gu, Y., Cheng, Y., Chen, C. P., and Wang, X. (2021). Proximal policy optimization with policy feedback. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 52(7):4600–4610.

[66] Guan, S., Zhu, Z., and Wang, G. (2022). A review on uav-based remote sensing technologies for construction and civil applications. *Drones*, 6(5):117.

[67] Guruji, A. K., Agarwal, H., and Parsediya, D. K. (2016). Time-efficient a* algorithm for robot path planning. *Procedia Technology*, 23:144–149.

[68] Gyenes, Z., Bölöni, L., and Szádeczky-Kardoss, E. G. (2023). Can genetic algorithms be used for real-time obstacle avoidance for lidar-equipped mobile robots? *Sensors*, 23(6):3039.

[69] Hai, X., Qiu, H., Wen, C., and Feng, Q. (2023). A novel distributed situation awareness consensus approach for uav swarm systems. *IEEE Transactions on Intelligent Transportation Systems*, 24(12):14706–14717.

[70] Han, L., Zhang, H., and An, N. (2025). A continuous space path planning method for unmanned aerial vehicle based on particle swarm optimization-enhanced deep q-network. *Drones*, 9(2):122.

[71] Haoran, Z., Hang, F., Fan, Y., Che, Q., and Yaoming, Z. (2024). Data-driven offline reinforcement learning approach for quadrotor's motion and path planning. *Chinese Journal of Aeronautics*, 37(11):386–397.

[72] He, K., Sun, L., Hong, H., Wang, N., Xiang, X., Lu, Z., and Cai, L. (2025). Target allocation for multiple uavs via swarm intelligence simulation and ddpg reinforcement learning. *International Journal of Modeling, Simulation & Scientific Computing*, 16(4).

[73] He, W., Qi, X., and Liu, L. (2021). A novel hybrid particle swarm optimization for multi-uav cooperate path planning. *Applied Intelligence*, 51(10):7350–7364.

[74] Hematulin, W., Kamsing, P., Torteeka, P., Somjit, T., Phisannupawong, T., and Jarawan, T. (2023). Trajectory planning for multiple uavs and hierarchical collision avoidance based on nonlinear kalman filters. *Drones*, 7(2):142.

[75] Hooshyar, M. and Huang, Y.-M. (2023). Meta-heuristic algorithms in uav path planning optimization: A systematic review (2018–2022). *Drones*, 7(12):687.

[76] Hou, Y., Zhao, J., Zhang, R., Cheng, X., and Yang, L. (2023). Uav swarm cooperative target search: A multi-agent reinforcement learning approach. *IEEE Transactions on Intelligent Vehicles*, 9(1):568–578.

[77] Houtekamer, P. L. and Mitchell, H. L. (2005). Ensemble kalman filtering. *Quarterly Journal of the Royal Meteorological Society: A journal of the atmospheric sciences, applied meteorology and physical oceanography*, 131(613):3269–3289.

[78] Hu, J., Bruno, A., Ritchken, B., Jackson, B., Espinosa, M., Delimitrou, C., et al. (2018). To centralize or not to centralize: A tale of swarm coordination. *arXiv preprint arXiv:1805.01786*.

[79] Huang, S., Zhang, H., and Huang, Z. (2022). Multi-uav collision avoidance using multi-agent reinforcement learning with counterfactual credit assignment. *arXiv preprint arXiv:2204.08594*.

[80] Hussein, A., Gaber, M. M., Elyan, E., and Jayne, C. (2017). Imitation learning: A survey of learning methods. *ACM Computing Surveys (CSUR)*, 50(2):1–35.

[81] Iqbal, H., Campo, D., Marcenaro, L., Gomez, D. M., and Regazzoni, C. (2021). Data-driven transition matrix estimation in probabilistic learning models for autonomous driving. *Signal Processing*, 188:108170.

[82] Iqbal, H., Sadia, H., Al-Kaff, A., and Garcia, F. (2025). Novelty detection in autonomous driving: A generative multi-modal sensor fusion approach. *IEEE Open Journal of Intelligent Transportation Systems*.

[83] Iqbal, M. M., Ali, Z. A., Khan, R., and Shafiq, M. (2022). Motion planning of uav swarm: Recent challenges and approaches. *Aeronautics-New Advances*. Published August 6.

[84] Javed, S., Hassan, A., Ahmad, R., Ahmed, W., Ahmed, R., Saadat, A., and Guizani, M. (2024). State-of-the-art and future research challenges in uav swarms. *IEEE Internet of Things Journal*, 11(11):19023–19045.

[85] Kaliappan, V., Nguyen, T., Jeon, S., Lee, J., and Min, D. (2021). Deep multi agent reinforcement learning based decentralized swarm uav control framework for persistent surveillance. In *Asia-Pacific International Symposium on Aerospace Technology*, pages 951–962, Singapore. Springer Nature Singapore.

[86] Karaboga, D. (2005). An idea based on honey bee swarm for numerical optimization. *Technical Report-TR06, Erciyes University*.

[87] Kennedy, J. and Eberhart, R. (1995). Particle swarm optimization. In *Proceedings of IEEE International Conference on Neural Networks*, pages 1942–1948. IEEE.

[88] Kerr, L. (2021). matlab-tsp-ga: Matlab functions to solve tsp / mtsp and other variations using a custom genetic algorithm (ga). https://github.com/LenKerr/matlab-tsp-ga. [Online].

[89] Khan, K. (2025). *PhD Thesis*. Doctoral dissertation, University of Genova, Department of DITEN, Genova, Italy.

[90] Khatib, O. (1986). Real-time obstacle avoidance for manipulators and mobile robots. *The international journal of robotics research*, 5(1):90–98.

[91] Khodagholi, A. (2022). Mopso-matlab: Multi-objective particle swarm optimization (pso) with matlab language programming. https://github.com/ali-khodagholi/MOPSO-MATLAB. [Online].

[92] Killian, L. and Backhaus, J. (2021). Utilizing the rrt*-algorithm for collision avoidance in uav photogrammetry missions. *arXiv preprint arXiv:2108.03863*.

[93] Kim, J., Park, M., and Lee, H. (2020). Imitation learning for uav swarm formation and coordination. *Robotics and Autonomous Systems*, 131:103568.

[94] Kladis, G. P., Doitsidis, L., and Tsourveloudis, N. C. (2024). Energy-efficient path-planning for uav swarm based missions: A genetic algorithm approach. In *2024 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 458–463. IEEE.

[95] Konda, V. R. and Tsitsiklis, J. N. (2000). Actor-critic algorithms. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 1008–1014.

[96] Krayani, A., Alam, A. S., Marcenaro, L., Nallanathan, A., and Regazzoni, C. (2022). A novel resource allocation for anti-jamming in cognitive-uavs: An active inference approach. *IEEE Communications Letters*, 26(10):2272–2276.

[97] Krayani, A., Khan, K., Marcenaro, L., Marchese, M., and Regazzoni, C. (2023). A goal-directed trajectory planning using active inference in uav-assisted wireless networks. *Sensors*, 23(15):6873.

[98] Krayani, A., Khan, K., Marcenaro, L., Marchese, M., and Regazzoni, C. (2024). Self-supervised path planning in uav-aided wireless networks based on active inference. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 13181–13185. IEEE.

[99] Kumar, K., Ghosh, D., Upadhayay, A., Yao, J. C., and Zhao, X. (2023). Quasi-newton methods for multiobjective optimization problems: A systematic review. *Applied Set-Valued Analysis & Optimization*, 5(2):291–321.

[100] Kumar, P., Pal, K., and Govil, M. (2025). Comprehensive review of path planning techniques for unmanned aerial vehicles (uavs). *ACM Computing Surveys*. To appear.

[101] Landmann, J. M., Künsch, H. R., Huss, M., Ogier, C., Kalisch, M., and Farinotti, D. (2021). Assimilating near-real-time mass balance stake readings into a model ensemble using a particle filter. *The Cryosphere*, 15(11):5017–5040.

[102] Latombe, J.-C. (1991). *Robot Motion Planning*. Springer. Fundamental reference for classical path planning methods like Dijkstra and A*.

[103] LaValle, S. (1998a). Rapidly-exploring random trees: A new tool for path planning. Department of Computer Science, Iowa State University.

[104] LaValle, S. M. (1998b). Rapidly-exploring random trees: A new tool for path planning. In *Technical Report TR 98-11*. Computer Science Department, Iowa State University.

[105] LaValle, S. M. (2006). *Planning Algorithms*. Cambridge University Press. Covers path planning vs. motion/trajectory planning concepts in detail.

[106] Lawler, E. L., Lenstra, J. K., Rinnooy Kan, A. H., and Shmoys, D. B. (1985). *The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization*. Wiley. Classic foundational book on TSP algorithms.

[107] Lee, J., Cho, J.-R., and Yoon, H. (2025). Feasibility analysis of applying particle filtering for data-based updating of existing seismic fragility. *Journal of the Computational Structural Engineering Institute of Korea*, 38(1):43–55.

[108] Lee, J. and Han, S. (2022). 3d a* path planning for uavs using octree-based space partitioning. *Robotics and Autonomous Systems*, 152:104045.

[109] Li, B. and Chen, B. (2021). An adaptive rapidly-exploring random tree. *IEEE/CAA Journal of Automatica Sinica*, 9(2):283–294.

[110] Li, L., Zhang, F., Yu, J., Zhu, Q., Lu, H., and Liu, S. (2023a). Exact and heuristic multi-robot dubins coverage path planning for known environments. *Sensors*, 23(5):2560.

[111] Li, X., Qin, Y., Huo, J., and Huangfu, W. (2023b). Computation offloading and trajectory planning of multi-uav-enabled mec: A knowledge-assisted multiagent reinforcement learning approach. *IEEE Transactions on Vehicular Technology*, 73(5):7077–7088.

[112] Lin, S., Li, F., Li, X., Jia, K., and Zhang, X. (2022). Improved artificial bee colony algorithm based on multi-strategy synthesis for uav path planning. *IEEE Access*, 10:119269–119282.

[113] Liu, J., Wang, Y., Huang, P.-Q., and Jiang, S. (2021a). Car: A cutting and repulsion-based evolutionary framework for mixed-integer programming problems. *IEEE Transactions on Cybernetics*, 52(12):13129–13141.

[114] Liu, L. S., Lin, J. F., Yao, J. X., He, D. W., Zheng, J. S., Huang, J., and Shi, P. (2021b). Path planning for smart car based on dijkstra algorithm and dynamic window approach. *Wireless Communications and Mobile Computing*, 2021(1):8881684.

[115] Liu, W., Zhang, B., Liu, P., Pan, J., and Chen, S. (2024). Velocity obstacle guided motion planning method in dynamic environments. *Journal of King Saud University-Computer and Information Sciences*, 36(1):101889.

[116] Liu, W.-h., Zheng, X., and Deng, Z.-h. (2021c). Dynamic collision avoidance for co-operative fixed-wing uav swarm based on normalized artificial potential field optimization. *Journal of Central South University*, 28(10):3159–3172.

[117] Liu, Y. and Jebelli, H. (2024). Intention-aware robot motion planning for safe worker–robot collaboration. *Computer-Aided Civil and Infrastructure Engineering*, 39(15):2242–2269.

[118] López, B., Muñoz, J., Quevedo, F., Monje, C. A., Garrido, S., and Moreno, L. E. (2021). Path planning and collision risk management strategy for multi-uav systems in 3d environments. *Sensors*, 21(13):4414.

[119] Lou, T.-s., Chen, N., Jiao, Y., Zhao, H., and Zhao, L. (2023). A consider unscented particle filter with genetic algorithm for uav multi-source integrated navigation. *Measurement Science and Technology*, 34(9):095105.

[120] Lu, L., Dai, J., and Ying, J. (2022). Distributed multi-uav cooperation for path planning by an ntvpso-ade algorithm. In *2022 41st Chinese Control Conference (CCC)*, pages 5973–5978. IEEE.

[121] Luo, D., Li, S., Shao, J., Xu, Y., and Liu, Y. (2022). Pigeon-inspired optimisation-based cooperative target searching for multi-uav in uncertain environment. *International Journal of Bio-Inspired Computation*, 19(3):158–168.

[122] Ma, B., Ji, Y., and Fang, L. (2025a). A multi-uav formation obstacle avoidance method combined with improved simulated annealing and an adaptive artificial potential field. *Drones*, 9(6):390.

[123] Ma, X., Yu, M., Luo, Z., and Gao, M. (2025b). Game-theoretic optimization of uav swarm and relay communication for interference resilience. *IEEE Transactions on Vehicular Technology*.

[124] Manullang, M. J. C., Priandana, K., and Hardhienata, M. K. D. (2023). Optimum trajectory of multi-uav for fertilization of paddy fields using ant colony optimization (aco) and 2-opt algorithms. In *AIP conference proceedings*, volume 2482. AIP Publishing.

[125] Marinakis, Y. (2024). Heuristic and metaheuristic algorithms for the traveling salesman problem. In *Encyclopedia of Optimization*, pages 1–12. Springer International Publishing, Cham.

[126] Masoud, A. A. (2012). A harmonic potential approach for simultaneous planning and control of a generic uav platform. *Journal of Intelligent & Robotic Systems*, 65(1):153–173.

[127] Memon, S., Krayani, A., Zontone, P., Marcenaro, L., Gomez, D. M., and Regazzoni, C. (2025). Leveraging dynamic interaction models for anomalous behavior detection in 3d environments. In *2025 33rd European Signal Processing Conference (EUSIPCO)*, pages 1677–1681. IEEE.

[128] Memon, S., Krayani, A., Zontone, P., Marcenaro, L., Gómez, D. M., and Regazzoni, C. (2026). Integrated sensing and communication for blockage detection in v2x networks. *IEEE Open Journal of the Communications Society*, 7:559–582.

[129] Meyer, F. and Glock, K. (2021). Trajectory-based traveling salesman problem for multirotor uavs. In *2021 17th International Conference on Distributed Computing in Sensor Systems (DCOSS)*, pages 335–342. IEEE.

[130] Millidge, B., Tschantz, A., and Buckley, C. L. (2021). Deep active inference: Scaling active inference using deep learning. *Frontiers in Computational Neuroscience*, 15:658112.

[131] Minu, M., Rani, P., Sonthi, V. K., Shankar, G., Mohan, E., and Rajesh, A. (2024). An innovative privacy preservation and security framework with fog nodes in enabled vanet system using hybrid encryption techniques. *Peer-to-Peer Networking and Applications*, pages 1–25.

[132] Mirjalili, S., Gandomi, A. H., Mirjalili, S. M., Saremi, S., Faris, H., and Mirjalili, S. (2017). Salp swarm algorithm: A bio-inspired optimizer for engineering design problems. *Advances in Engineering Software*, 114:163–191.

[133] Mnih, V., Kavukcuoglu, K., Silver, D., et al. (2015). Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533.

[134] Mohamed, A. and Alsharif, K. (2023). Hierarchical a* for large-scale uav mission planning. *Applied Soft Computing*, 134:109896.

[135] Mok, J., Lee, Y., Ko, S., Choi, I., and Choi, H. S. (2017). Gaussian-mixture based potential field approach for uav collision avoidance. In *2017 56th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE)*, pages 1316–1319. IEEE.

[136] Momoh, J. A. and Zhu, J. (1999). Improved interior point method for opf problems. *IEEE Transactions on Power Systems*, 14(3):1114–1120.

[137] Moon, B., Hong, J. H., Mettler, E., Rathinam, S., and Tsiotras, P. (2023). Time-optimal path planning in a constant wind for uncrewed aerial vehicles using dubins set classification. *IEEE Robotics and Automation Letters*, 9(3):2176–2183.

[138] Mozga, N., Gutans, J., Kubulins, R., and Chatys, R. (2024). Calculation and design of the main equipment for mobile space simulation system. *Transactions on Aerospace Research*, 274(1):71–89.

[139] Muhsen, D. K., Raheem, F. A., and Sadiq, A. T. (2024). Improved rapidly exploring random tree using salp swarm algorithm. *Journal of Intelligent Systems*, 33(1):20230219.

[140] Muntasha, G., Karna, N., and Shin, S. (2021). Performance analysis on artificial bee colony algorithm for path planning and collision avoidance in swarm unmanned aerial vehicle. In *2021 International Conference on Artificial Intelligence and Mechatronics Systems (AIMS)*, pages 1–6. IEEE.

[141] Nekovář, F., Faigl, J., and Saska, M. (2021). Multi-tour set traveling salesman problem in planning power transmission line inspection. *IEEE Robotics and Automation Letters*, 6(4):6196–6203.

[142] Nguyen, V. D., Yang, Z., Buckley, C. L., and Ororbia, A. (2024). R-aif: Solving sparse-reward robotic tasks from pixels with active inference and world models.

[143] Nozari, S., Krayani, A., Marcenaro, L., Martin, D., and Regazzoni, C. (2022). Incremental learning through probabilistic behavior prediction. In *2022 30th European Signal Processing Conference (EUSIPCO)*, pages 1502–1506. IEEE.

[144] Olofsson, M., Andersson, G., and Soder, L. (1995). Linear programming based optimal power flow using second-order sensitivities. *IEEE Transactions on Power Systems*, 10(3):1691–1697.

[145] Osa, T., Pajarinen, J., Neumann, G., Bagnell, J. A., Abbeel, P., Peters, J., et al. (2018). An algorithmic perspective on imitation learning. *Foundations and Trends® in Robotics*, 7(1-2):1–179.

[146] Page, M., McKenzie, J., Bossuyt, P., Boutron, I., Hoffmann, T., Mulrow, C., Shamseer, L., Tetzlaff, J., Akl, E., Brennan, S., et al. (2021). The prisma 2020 statement: An updated guideline for reporting systematic reviews. *BMJ*, 372:n71.

[147] Pan, L., Zhou, H., and Wang, Q. (2022). Imitation learning for multi-uav cooperative mission planning. *IEEE Access*, 10:45221–45233.

[148] Pan, Y., Yang, Y., and Li, W. (2021). A deep learning trained by genetic algorithm to improve the efficiency of path planning for data collection with multi-uav. *IEEE Access*, 9:7994–8005.

[149] Papageorgiou, D. (2019). Uav-tsp-simulation: Traveling salesman problem, uav simulation using 2-opt heuristic algorithm. https://github.com/dimitrisppt/UAV-TSP-Simulation. [Online].

[150] Parkinson, J. and Patel, N. (2022). Jump point search enhanced a* for uav real-time re-planning. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 4421–4428.

[151] Perrusquía, A. and Guo, W. (2024). Uncovering reward goals in distributed drone swarms using physics-informed multiagent inverse reinforcement learning. *IEEE transactions on cybernetics*.

[152] Pezzulo, G., Parr, T., and Friston, K. (2024). Active inference as a theory of sentient behavior. *Biological Psychology*, 186:108741.

[153] Pezzulo, G., Rigoli, F., and Friston, K. (2022). Bayesian active inference models for adaptive uav decision-making. *Cognitive Systems Research*, 72:1–15.

[154] Pham, Q.-C., Dequidt, J., and Pham, M.-T. (Dec. 2017). Admissible velocity propagation: Beyond quasi-static path planning for high-dimensional robots. *The International Journal of Robotics Research*, 36(1):44–67.

[155] Phung, M. D. and Ha, Q. P. (2021). Safety-enhanced uav path planning with spherical vector-based particle swarm optimization. *Applied Soft Computing*, 107:107376.

[156] Priya, P. and Kamlu, S. S. (2023). Unmanned aerial system trajectory tracking based on diversified grey wolf optimization algorithm. *IEEE Access*, 11:145975–145988.

[157] Puente-Castro, A., Rivero, D., Pazos, A., and Fernandez-Blanco, E. (2022). A review of artificial intelligence applied to path planning in uav swarms. *Neural Computing and Applications*, 34(1):153–170.

[158] Qi, L., Zhang, T., Chen, G., and Wang, W. (2025). Robust unmanned aerial vehicles tracking amid electronic interference utilizing auxiliary particle filtering. *PLoS One*, 20(9):e0333009.

[159] Qian, F., Su, K., Liang, X., and Zhang, K. (2023). Task assignment for uav swarm saturation attack: A deep reinforcement learning approach. *Electronics*, 12(6):1292.

[160] Qiu, H. and Duan, H. (2020). A multi-objective pigeon-inspired optimization approach to uav distributed flocking among obstacles. *Information Sciences*, 509:515–529.

[161] Reda, M., Onsy, A., Haikal, A. Y., and Ghanbari, A. (2024). Path planning algorithms in the autonomous driving system: A comprehensive review. *Robotics and Autonomous Systems*, 174:104630.

[162] Rezaee, M. R., Hamid, N. A. W. A., Hussin, M., and Zukarnain, Z. A. (2024). Comprehensive review of drones collision avoidance schemes: Challenges and open issues. *IEEE Transactions on Intelligent Transportation Systems*, 25(7):6397–6426.

[163] Ribeiro, M. I. (2004). Kalman and extended kalman filters: Concept, derivation and properties. *Institute for Systems and Robotics*, 43(46):3736–3741.

[164] Richter, C., Bry, A., and Roy, N. (2016). Polynomial trajectory planning for aggressive quadrotor flight in dense indoor environments. In *Robotics Research*, pages 649–666. Springer.

[165] Rizk, Y., Awad, M., and Tunstel, E. W. (2018). Decision making in multiagent systems: A survey. *IEEE Transactions on Cognitive and Developmental Systems*, 10(3):514–529.

[166] Sabetghadam, B., Cunha, R., and Pascoal, A. (2022). A distributed algorithm for real-time multi-drone collision-free trajectory replanning. *Sensors*, 22(5):1855.

[167] Saeed, R. A., Omri, M., Abdel-Khalek, S., Ali, E. S., and Alotaibi, M. F. (2022). Optimal path planning for drones based on swarm intelligence algorithm. *Neural Computing and Applications*, 34(12):10133–10155.

[168] Salgado, R., Brameller, A., and Aitchison, P. (1990). Optimal power flow solutions using the gradient projection method. part 1: Theoretical basis. In *IEE Proceedings C (Generation, Transmission and Distribution)*, volume 137, pages 424–428. IET.

[169] Salve, S. S., Chaudhari, S. Y., Dandekar, A. R., and Gaikwad, P. (2025). Anti collision drone traffic control system using swarm technology.

[170] Schöllig, A., Mueller, M., and D'Andrea, R. (2012). Trajectory generation for quadrotor swarms. *IEEE Transactions on Robotics*, 28(5):1186–1199.

[171] Schumann, J., Engstroem, J., Johnson, L., O'Kelly, M., Messias, J., Kober, J., and Zgonnikov, A. (2025). Active inference as a unified model of collision avoidance behavior in human drivers. *arXiv preprint arXiv:2506.02215*.

[172] Shao, S., Li, H., Zhao, Y., and Wu, X. (2023). A new method for multi-uav cooperative mission planning under fault. *IEEE Access*, 11:52653–52667.

[173] Sharma, A., Shoval, S., Sharma, A., and Pandey, J. K. (2022). Path planning for multiple targets interception by the swarm of uavs based on swarm intelligence algorithms: A review. *IETE Technical Review*, 39(3):675–697.

[174] Shi, B., Chen, Z., and Xu, Z. (2024). A deep reinforcement learning based approach for optimizing trajectory and frequency in energy constrained multi-uav assisted mec system. *IEEE Transactions on Network and Service Management*.

[175] Shin, J.-J. and Bang, H. (2020). Uav path planning under dynamic threats using an improved pso algorithm. *International Journal of Aerospace Engineering*, 2020(1):8820284.

[176] Shukla, P., Shukla, S., and Singh, A. K. (2024). Trajectory-prediction techniques for unmanned aerial vehicles (uavs): A comprehensive survey. *IEEE Communications Surveys & Tutorials*. Early Access.

[177] Sindiramutty, S. R. (2025). Swarm intelligence and multi-drone coordination with edge ai. In *Computer Vision and Edge Computing Technologies for the Drone Industry*, pages 271–304. IGI Global Scientific Publishing.

[178] Singh, A. and Payal, A. (2021a). Development of a low-cost collision avoidance system based on coulomb's inverse-square law for multi-rotor drones (uavs). In *2021 International Conference on Computational Performance Evaluation (ComPE)*, pages 306–316. IEEE.

[179] Singh, A. and Payal, A. (Dec. 2021b). Development of a low-cost collision avoidance system based on coulomb's inverse-square law for multi-rotor drones (uavs). pages 306–316.

[180] Singh, N., Singh, S., and Houssein, E. H. (2022). Hybridizing salp swarm algorithm with particle swarm optimization algorithm for recent optimization functions. *Evolutionary Intelligence*, 15(1):23–56.

[181] Smith, T., Clark, A., and Rao, V. (2022). Active inference for autonomous uav navigation in uncertain environments. *Neural Networks*, 152:135–148.

[182] Song, C., Zhang, X., She, Y., Li, B., and Zhang, Q. (2025). Trajectory planning for uav swarm tracking moving target based on an improved model predictive control fusion algorithm. *IEEE Internet of Things Journal*.

[183] Song, X., Liu, X., and Lu, J. (2020). Dynamic local laplacian potential field for uav navigation in unknown environments. *IEEE Transactions on Control Systems Technology*. Early Access.

[184] Sönmez, S., Rutherford, M. J., and Valavanis, K. P. (2024). A survey of offline-and online-learning-based algorithms for multirotor uavs. *Drones*, 8(4):116.

[185] Sonny, A., Yeduri, S. R., and Cenkeramaddi, L. R. (2023). Autonomous uav path planning using modified pso for uav-assisted wireless networks. *IEEE Access*.

[186] Tang, H., Dou, H., Gao, Q., Mao, Z., Ji, Y., and Liu, J. (2025a). An improved gaussian sampling-based bidirectional rrt algorithm in 3d path planning for low-altitude urban environments. In *2025 37th Chinese Control and Decision Conference (CCDC)*, pages 2494–2499. IEEE.

[187] Tang, J., Duan, H., and Lao, S. (2023). Swarm intelligence algorithms for multiple unmanned aerial vehicles collaboration: A comprehensive review. *Artificial Intelligence Review*, 56(5):4295–4327.

[188] Tang, R., Tang, J., Talip, M. S., Aridas, N. K., and Xu, X. (2025b). Enhanced multi agent coordination algorithm for drone swarm patrolling in durian orchards. *Scientific Reports*, 15(1):9139.

[189] Tinney, W. F. and Hart, C. E. (1967). Power flow solution by newton's method. *IEEE Transactions on Power Apparatus and systems*, (11):1449–1460.

[190] Todrov, E. V. (1998). *Studies of goal directed movements*. PhD thesis, Massachusetts Institute of Technology.

[191] Tong, G., Jiang, N., Biyue, L., Xi, Z., Ya, W., and Wenbo, D. (2021a). Uav navigation in high dynamic environments: A deep reinforcement learning approach. *Chinese Journal of Aeronautics*, 34(2):479–489.

[192] Tong, G., Jiang, N., Biyue, L., Xi, Z., Ya, W., and Wenbo, D. (2021b). Uav navigation in high dynamic environments: A deep reinforcement learning approach. *Chinese Journal of Aeronautics*, 34(2):479–489.

[193] Tripathy, S. P., Biswas, A., and Pal, T. (2021). A multi-objective covering salesman problem with 2-coverage. *Applied Soft Computing*, 113:108024.

[194] ul Husnain, A., Mokhtar, N., Mohamed Shah, N., Dahari, M., and Iwahashi, M. (2023). A systematic literature review (slr) on autonomous path planning of unmanned aerial vehicles. *Drones*, 7(2):118.

[195] Venturini, F., Mason, F., Pase, F., Chiariotti, F., Testolin, A., Zanella, A., and Zorzi, M. (2021). Distributed reinforcement learning for flexible and efficient uav swarm control. *IEEE Transactions on Cognitive Communications and Networking*, 7(3):955–969.

[196] Wan, E. A. and Van Der Merwe, R. (2000). The unscented kalman filter for nonlinear estimation. In *Proceedings of the IEEE 2000 adaptive systems for signal processing, communications, and control symposium (Cat. No. 00EX373)*, pages 153–158. Ieee.

[197] Wan, Y., Tang, J., and Zhao, Z. (2023). Imitation learning of complex behaviors for multiple drones with limited vision. *Drones*, 7(12):704.

[198] Wang, F., Xu, G., and Wang, M. (2023a). An improved genetic algorithm for constrained optimization problems. *IEEE Access*, 11:10032–10044.

[199] Wang, H. and Zhao, J. (2023). A novel high-level target navigation pigeon-inspired optimization for global optimization problems. *Applied Intelligence*, 53(12):14918–14960.

[200] Wang, J., Li, Y., Li, R., Chen, H., and Chu, K. (2022). Trajectory planning for uav navigation in dynamic environments with matrix alignment dijkstra. *Soft Computing*, 26(22):12599–12610.

[201] Wang, J., Sun, Y., Wang, B., and Ushio, T. (2023b). Mission-aware uav deployment for post-disaster scenarios: A worst-case sac-based approach. *IEEE Transactions on Vehicular Technology*, 73(2):2712–2727.

[202] Wang, L., Huang, W., Li, H., Li, W., Chen, J., and Wu, W. (2024a). A review of collaborative trajectory planning for multiple unmanned aerial vehicles. *Processes*, 12(6):1272.

[203] Wang, R., Shan, Y., Sun, L., and Sun, H. (2025). Multi-uav cooperative task allocation based on multi-strategy clustering ant colony optimization algorithm. *ICCK Transactions on Intelligent Systematics*, 2(3):149–159.

[204] Wang, S., Qi, N., Jiang, H., Xiao, M., Liu, H., Jia, L., and Zhao, D. (2024b). Trajectory planning for uav-assisted data collection in iot network: A double deep q network approach. *Electronics*, 13(8):1592.

[205] Watkins, C. J. and Dayan, P. (1992). Q-learning. *Machine Learning*, 8(3-4):279–292.

[206] Whiteley, N. and Johansen, A. M. (2010). Recent developments in auxiliary particle filtering. *Inference and learning in dynamic models*, 38:39–47.

[207] William, N. J., Krayani, A., Marcenaro, L., and Regazzoni, C. (2024). Interactive bayesian generative models for abnormality detection in vehicular networks. In *2024 IEEE Wireless Communications and Networking Conference (WCNC)*, pages 1–7. IEEE.

[208] Wolek, A., Seidel, J., Kaminer, I., Dobrokhodov, V., Cobb, R., and Innes, J. (2025). Maximum kinetic energy paths for a decaying-speed dubins vehicle. In *AIAA SCITECH 2025 Forum*. American Institute of Aeronautics and Astronautics.

[209] Wu, X., Yin, Y., Xu, L., Wu, X., Meng, F., and Zhen, R. (2021). Multi-uav task allocation based on improved genetic algorithm. *IEEE Access*, 9:100369–100379.

[210] Xia, W., Li, G., and Zeng, Y. (2022). Research on multi-traveling salesman problem based on simulated annealing algorithm. In *2022 International Conference on Computers, Information Processing and Advanced Education (CIPAE)*, pages 390–393. IEEE.

[211] Xia, Z., Du, J., Wang, J., Jiang, C., Ren, Y., Li, G., and Han, Z. (2021a). Multi-agent reinforcement learning aided intelligent uav swarm for target tracking. *IEEE Transactions on Vehicular Technology*, 71(1):931–945.

[212] Xia, Z., Du, J., Wang, J., Jiang, C., Ren, Y., Li, G., and Han, Z. (2021b). Multi-agent reinforcement learning aided intelligent uav swarm for target tracking. *IEEE Transactions on Vehicular Technology*, 71(1):931–945.

[213] Xiang, J., Li, Q., Dong, X., and Ren, Z. (2019). Continuous control with deep reinforcement learning for mobile robot navigation. In *2019 Chinese Automation Congress (CAC)*, pages 1501–1506. IEEE.

[214] Xu, P., Liu, J., Sun, X., Chen, H., and Chen, Y. (2024). Distributed consensus control research of unmanned aerial vehicle (uav) swarms based on lennard-jones potential. In *International Conference on Machine Learning, Cloud Computing and Intelligent Mining*, pages 154–165. Springer Nature Singapore.

[215] Xu, R. and Yao, S. (2022). Research on ugv path planning in tunnel based on the dijkstra*-pso* algorithm. In *2022 6th Asian Conference on Artificial Intelligence Technology (ACAIT)*, pages 1–9. IEEE.

[216] Xu, W., Zhang, Y., Yu, L., Zhang, T., and Cheng, Z. (2023). A local path planning algorithm based on improved dynamic window approach. *Journal of Intelligent & Fuzzy Systems*, 45(3):4917–4933.

[217] Xu, X., Xie, C., Ma, L., Yang, L., and Zhang, T. (2025). Multi-objective evolutionary algorithm with two balancing mechanisms for heterogeneous uav swarm path planning. *Applied Soft Computing*, 173:112927.

[218] Yafei, W. and Liang, Z. (2023). Improved multi-objective particle swarm optimization algorithm based on area division with application in multi-uav task assignment. *IEEE Access*, 11:123519–123530.

[219] Yahia, H. S. and Mohammed, A. S. (2023). Path planning optimization in unmanned aerial vehicles using meta-heuristic algorithms: A systematic review. *Environmental Monitoring and Assessment*, 195(1):30.

[220] Yan, P., Ma, L., Li, Y., Yu, J., and Chen, C. (2018). A fixed wing uav path planning algorithm based on genetic algorithm and dubins curve theory. In *MATEC Web of Conferences*, volume 179, page 02015. EDP Sciences.

[221] Yang, B., Shi, H., and Xia, X. (2022a). Federated imitation learning for uav swarm coordination in urban traffic monitoring. *IEEE Transactions on Industrial Informatics*, 19(4):6037–6046.

[222] Yang, L., Yu, J., Yang, S., Wang, B., Nelson, B. J., and Zhang, L. (2021). A survey on swarm microrobotics. *IEEE Transactions on Robotics*, 38(3):1531–1551.

[223] Yang, Y., He, Q., and Yang, L. (2022b). Uav trajectory planning based on an improved sparrow optimization algorithm with multi-strategy integration. *Frontiers in Environmental Science*, 10:1055807.

[224] Yang, Y., Xiong, X., and Yan, Y. (2023). Uav formation trajectory planning algorithms: A review. *Drones*, 7(1):62.

[225] Yao, J., Sha, Y., Chen, Y., Zhang, G., Hu, X., Bai, G., and Liu, J. (2022). Ihssao: An improved hybrid salp swarm algorithm and aquila optimizer for uav path planning in complex terrain. *Applied Sciences*, 12(11):5634.

[226] Yasin, J. N., Mohamed, S. A., Haghbayan, M. H., Heikkonen, J., Tenhunen, H., and Plosila, J. (2020). Unmanned aerial vehicles (uavs): Collision avoidance systems and approaches. *IEEE Access*, 8:105139–105155.

[227] Yin, H., Li, B., Zhu, H., and Shi, L. (2023). Kinodynamic rrt* based uav optimal state motion planning with collision risk awareness. *Information Technology and Control*, 52(3):665–679.

[228] Yu, Z., Si, Z., Li, X., Wang, D., and Song, H. (2022). A novel hybrid particle swarm optimization algorithm for path planning of uavs. *IEEE Internet of Things Journal*, 9(22):22547–22558.

[229] Zeng, F., Wang, C., and Ge, S. S. (2020). A survey on visual navigation for artificial agents with deep reinforcement learning. *IEEE Access*, 8:135426–135442.

[230] Zhang, H. and Xu, S. (2025). Path planning technology for unmanned aerial vehicle swarm based on improved jump point algorithm. *International Journal of Advanced Computer Science & Applications*, 16(4).

[231] Zhang, K., Yang, Z., and Basar, T. (2021a). Multi-agent reinforcement learning: A selective overview of theories and algorithms. *Handbook of Reinforcement Learning and Control*, pages 321–384.

[232] Zhang, L., Peng, J., Yi, W., Lin, H., Lei, L., and Song, X. (2023a). A state-decomposition ddpg algorithm for uav autonomous navigation in 3-d complex environments. *IEEE Internet of Things Journal*, 11(6):10778–10790.

[233] Zhang, L., Xu, R., and Han, Y. (2023b). Hybrid imitation–reinforcement learning for uav swarms in dynamic environments. *Applied Soft Computing*, 135:109932.

[234] Zhang, L. and Zhao, M. (2021). Grid-based a* algorithm for uav swarm scheduling in urban environments. *Journal of Intelligent & Robotic Systems*, 101(2):1–15.

[235] Zhang, S., Liu, S., Xu, W., and Wang, W. (2022a). A novel multi-objective optimization model for the vehicle routing problem with drone delivery and dynamic flight endurance. *Computers & Industrial Engineering*, 173:108679.

[236] Zhang, W., Zhao, L., and Liu, H. (2022b). Multi-uav path planning based on deep reinforcement learning for mtsp. *Aerospace Science and Technology*, 126:107670.

[237] Zhang, Y., Han, X., Dong, Y., Xie, J., Xie, G., and Xu, X. (2021b). A novel state transition simulated annealing algorithm for the multiple traveling salesmen problem. *The Journal of Supercomputing*, 77:11827–11852.

[238] Zhang, Y., Yi, P., and Hong, Y. (2024). Cooperative safe trajectory planning for quadrotor swarms. *Sensors*, 24(2):707.

[239] Zhao, L., Chen, B., and Hu, F. (2024). Research on cooperative obstacle avoidance decision making of unmanned aerial vehicle swarms in complex environments under end-edge-cloud collaboration model. *Drones*, 8(9):461.

[240] Zhen, Z., Chen, Y., Wen, L., and Han, B. (2020). An intelligent cooperative mission planning scheme of uav swarm in uncertain dynamic environment. *Aerospace Science and Technology*, 100:105826.

[241] Zheng, J., Ding, M., Sun, L., and Liu, H. (2023). Distributed stochastic algorithm based on enhanced genetic algorithm for path planning of multi-uav cooperative area search. *IEEE Transactions on Intelligent Transportation Systems*, 24(8):8290–8303.