



CRANFIELD UNIVERSITY

Ruifan Liu

**Decentralized Mission Planning for
Multiple Unmanned Aerial Vehicles**

School of Aerospace, Transport and Manufacturing

PhD Thesis

Cranfield University

School of Aerospace, Transport and Manufacturing

PhD Thesis

Academic Year 2019-2022

Ruifan Liu

Decentralized Mission Planning for Multiple
Unmanned Aerial Vehicles

Supervisors: Prof. Hyo-Sang Shin
Prof. Antonios Tsourdos

February 2023

This thesis is submitted in partial fulfilment of the requirements for the degree of
Doctor of Philosophy.

©Cranfield University 2023. All rights reserved. No part of this publication may
be reproduced without the written permission of the copyright owner.

Abstract

The focus of this thesis is the mission planning challenge for multiple unmanned aerial vehicles (UAVs), with a particular emphasis on their stable operations in a stochastic and dynamic environment. Mission planning is a crucial module in automated multi-UAV systems, allowing for efficient resource allocation, conflict resolution, and reliable operation. However, the distributed nature of the system and physical and environmental constraints make it challenging to develop effective mission planning algorithms.

The thesis begins with a review of the taxonomy, frameworks, and techniques in multiple-UAV mission planning. Following this, it identifies four critical research challenges, encompassing scalability, efficiency, adaptability and robustness, and energy management and renewable strategies. In response to these challenges, four objectives have been defined with the overall aim of developing a generic decentralized mission planning paradigm for multi-UAV systems. The thesis subsequently concentrates on accomplishing these objectives, with notable contributions in the development of *1)* a decentralized task coordination algorithm, *2)* an efficient route planner in consideration of recharging, and *3)* an energy-aware planning framework.

This research first proposes a decentralized auction-based coordination strategy for task-constrained multi-agent stochastic planning problems. Through casting the problem as task-constrained Markov decision processes (MDPs), the task dependency due to an exclusive constraint is despatched from Multi-agent Markov decision processes (M-MDPs) and then resolved by adopting an auction-based coordination method. For multi-agent stochastic planning problems, the suggested technique resolves the trade-off concern between computational tractability and solution quality. The proposed method ensures convergence, achieves at least 50% optimality under the assumption of a submodular reward function, and greatly reduces the computational complexity compared to multi-agent MDPs. Deep Auction is then proposed as an approximate modification of the suggested auction-based coordination method, where two neural network approximators are introduced to facilitate scaled-up implementations. By theoretical analysis, these two proposed algorithms achieve better robustness and feature less computing complexity compared to the state-of-the-art. Finally, a case study of drone delivery with time windows is implemented for validation. Simulation results demonstrate the theoretical benefits of the recommended methodologies.

Then, an efficient route planner for individual UAVs accounting for recharging services is proposed. Despite extensive research in decision-making algorithms, existing models have limitations in accurately representing real-world scenarios in terms of UAV's physical restrictions and stochastic operating environments. To address this, a drone delivery problem with recharging (DDP-R) is proposed. The problem is characterized by directional edges and stochastic edge costs affected by winds. To solve DDP-Rs, a novel edge-enhanced attention model (AM-E) is proposed and trained via the REINFORCE algorithm to map the optimal policy. AM-E consists of a series of edge-enhanced dot-product attention layers that capture the heterogeneous relationships between nodes in DDP-Rs by incorporating adjacent edge information. Simulation results show that the edge enhancement achieves better results with a simpler architecture and fewer trainable parameters, compared to other deep learning models. Extensive simulations demonstrate that the proposed DRL method outperforms state-of-the-art heuristics in solving the DDP-R problem, especially at large sizes, for both non-wind and windy scenarios.

Finally, we integrate the above route planning algorithm into an online energy inference framework, namely, the Energy-aware Planning Framework (EaPF), with the aim of optimizing solution quality in consideration of possible time-window violation and battery depletion. The framework comprises a statistical energy predictive model, a risk assessment module, and a route optimizer, with functions of modelling energy costs, estimating risks, and optimizing the risk-sensitive objective. Concretely, a Mixture Density Network (MDN) is established for predicting the distribution of future energy consumption taking account of wind conditions. The MDN is trained by historical data and continuously updated as new data is collected. Then, a risk-sensitive criterion is formed based on the MDN model of energy consumption to assess the risk of task lateness and battery depletion. To minimize the risk-sensitive objective, the EaPF incorporates the proposed AM-E planner using a Model-based Multi-Sampling (MBMS) route construction strategy, to further improve solution quality and planning robustness. In the context of drone deliveries, simulations validate the effectiveness of the MDN energy model and the EaPF. Results show that the integration of EaPF achieves an average cost reduction of 25%, which implies a lower energy cost, a higher task accomplishment rate, and a smaller battery depletion risk compared to the stand-alone DRL planner.

Acknowledgements

First of all, I would like to express my sincere gratitude to Prof. Antonios Tsourdos for his help in my life and academics. Without his support, I would never have had the opportunity to enjoy the invaluable time of my PhD study at Cranfield University. He has always been supportive of my PhD study and actively provided me with a nice working environment to pursue my PhD goals and great opportunities to broaden the scope of my research.

My appreciation also goes out to Prof. Hyo-Sang Shin, who has been giving me his insightful supervision, encouragement, and support on my work. His genuine dedication to academic research has greatly motivated me and will continue to do so throughout the remainder of my research career.

I would like to thank all Autonomous and Intelligent Systems Group members: Dr. Min-Guk Seo, Dr. Teng Li, Dr. Seokwon Lee, Saki Omi and so on. Their generous sharing of knowledge has expanded my vision and inspired my research. I would also like to thank my DARTeC lab mates and friends Huw Whitworth and Anirudh Warriar for giving my Ph.D. life more laughter.

Finally, my deepest appreciation goes to my family, for their eternal love and faith in me.

Contents

Contents	v
List of figures	ix
List of tables	xii
Abbreviations	xiv
Notations	xvii
1 Introduction	1
1.1 Background and Motivation	1
1.1.1 Multi-UAV Mission Planning Architecture and Workflow . . .	4
1.1.2 Existing Multi-UAV Mission Planning Workflows	7
1.1.3 Key Technologies in Multi-UAV Mission Planning	14
1.1.4 Challenges in Multi-UAV Mission Planning	20
1.2 Aim and Objectives	21
1.3 Contributions	22
1.4 Organization	25
1.5 Publications	25
2 An Auction-Based Task Coordination Strategy	28
2.1 Introduction	28
2.2 Related Works	30

2.2.1	Task-Constrained Stochastic Planning	30
2.2.2	Multi-Agent Coordination Strategy	31
2.3	Mathematical Formulation	32
2.3.1	Problem Definition	32
2.3.2	Problem Formulation	33
2.3.3	Preliminary of Submodularity	35
2.4	Auction-based Coordination Strategy	36
2.4.1	CBBA and its Robust Extension	36
2.4.2	The Proposed Coordination Strategy	37
2.4.3	Convergence Analysis with Submodular Rewards	39
2.4.4	Optimality Analysis	41
2.5	Approximation via Neural Networks	42
2.5.1	Deep Auction	42
2.5.2	Network Architecture	44
2.5.3	Training Algorithm	45
2.5.4	Convergence Guarantee	46
2.6	Comparison with Base Methods	47
2.7	Case Study on Drone Deliveries	49
2.7.1	Drone Delivery Problem with Time Windows	49
2.7.2	Simulations with Small-Scale Problems	49
2.7.3	Deep Auction Training Result	51
2.7.4	Comparative Study with Deep Auction	53
2.8	Conclusion	57
3	Edge-enhanced Attentions for Route Planning	59
3.1	Introduction	59
3.2	Related Works	62
3.2.1	Exact and Heuristic Methods	62

3.2.2	Learning-Based Approaches	63
3.3	Problem Statement	64
3.3.1	Drone Delivery Problem with Recharging	64
3.3.2	Drone's Energy Consumption Model with the Wind Effect	65
3.3.3	Markov Decision Process for DDP-Rs	69
3.4	Reinforcement Learning Model	71
3.4.1	Edge-Enhanced Attention Model	71
3.4.2	Masking Scheme	78
3.4.3	Training Method	78
3.5	Numerical Simulations	79
3.5.1	Experimental Setup	80
3.5.2	Ablation Study of Attention Model with Edges	81
3.5.3	Evaluation Against Other Models	81
3.5.4	Validation of Edge Feature Enhancement	81
3.5.5	DDP-Rs without Winds	83
3.5.6	DDP-Rs in Presence of Winds	85
3.5.7	Generalization with Dimensions	87
3.6	Conclusion	89
4	A Model-based Energy-aware Route Planning Framework	91
4.1	Introduction	91
4.2	Related Works	94
4.2.1	Drone Energy-aware Planning	94
4.2.2	Simulation-based Optimization for Routing Problem	95
4.3	Energy-constrained Drone Delivery	96
4.3.1	Route Optimization for Drone Delivery	96
4.3.2	Markov Modelling for Drone Delivery	98
4.4	Energy-Aware Planning Framework	99

4.4.1	MBMS Route Construction Strategy	100
4.5	Energy Modelling and Risk-Aware Criterion	102
4.5.1	Energy Prediction Model via MDN	102
4.5.2	Risk-Aware Criterion	104
4.6	DRL Route Optimizer	106
4.6.1	Policy Network and Masking Scheme	106
4.6.2	Training of the Planning Network	106
4.7	Validation of the MDN Energy Model	107
4.8	Case Study on Drone Deliveries	109
4.8.1	Experimental Setup	110
4.8.2	Simulation Results	113
4.9	Conclusion	117
5	Overall Discussion	118
5.1	Simulation Environment Development	118
5.2	Accomplishments Toward Objectives	119
5.3	Benefits to the Real World	121
5.4	Gaps from the Real-World Implementation	121
5.5	New Insights from the Research	123
6	Conclusion and Future work	125
6.1	Conclusion	125
6.2	Future Work	127
	References	128

List of Figures

1.1	NASA's vision for civil applications of drones [1].	2
1.2	Representative challenges for mission planning specific to multi-UAV systems. Coloured bricks indicate main concerns of the thesis.	3
1.3	Three-tier architecture of mission planning for multi-UAV systems	5
1.4	Workflow for the multi-robot system proposed in [2]	6
1.5	Three different drone delivery configurations: (top left) multi-drone delivery framework using transit networks [3]; (bottom left) multi-trip vehicle routing problem [4]; (right) multi-drone delivery framework with recharging services [5].	9
1.6	Workflow of multi-UAV systems in search and rescue.	10
1.7	The workflow of active multi-UAV multi-target surveillance.	12
1.8	Information-driven loop of multi-UAV cooperative monitoring.	13
1.9	Blocked diagram highlighting the relations of proposed algorithms.	23
2.1	The training phase of the Deep Auction method	43
2.2	The implementation phase of the Deep Auction method	43
2.3	The neural network architecture. The critic network and actor network share the same architecture except for the output layer for the corresponding output format.	44
2.4	Running time for all three methods as the customer number changes.	53
2.5	Learning curves of training actor and critic networks: (left) cumulative reward on training and test datasets per epoch; (right-top) critic loss on training and test datasets per epoch; (right-bottom) policy loss on training and test datasets per epoch.	54

2.6	Running time for Deep Auction and CBBA as the customer number changes.	56
3.1	Drone delivery with recharging	65
3.2	The flight profile for the VTOL drone delivery, consisting of four phases: takeoff, level flight, hovering and landing.	66
3.3	Energy consumptions for a 10km flight with airspeed and package weight varying. In the first figure, three cases are considered with package weight $m_{load} = 0, 2kg, 5kg$ respectively. In the second figure, a contour map of energy consumption is depicted with regard to airspeed and package weight. In simulations, no wind is considered, altitude for level flight $a = 100m$, hovering time $t_{hover} = 5s$, and vertical speed is assumed as a constant $v_v = 3\frac{m}{s}$	67
3.4	Statistical distribution of energy consumption for a 10km flight with the wind. In Monte Carlo simulations, the shape parameter of the wind Weibull distribution is set as $k = 2$	68
3.5	An example of the adjacent matrix in DDP-Rs. According to delivery logic and payload constraint, the connectivity matrix is defined as shown in the figure.	72
3.6	Two edge-enhanced GAT architectures. In the first architecture, both of node features and edge features are evolved for each attention layer, while the second one only updates the node features.	73
3.7	(left) Scaled dot-product attention. (right) Edge-enhanced scaled dot-product attention. In edge-enhanced scaled dot-product attention, edge features are projected and then merged into the original <i>value</i> vector and the <i>compatibility</i> vector, i.e., the scaled dot-product of <i>query</i> and <i>key</i> vectors.	74
3.8	AM-E policy network.	76
3.9	Learning curves for solving the DDP-R with 10 delivery requests and 3 recharge stations.	82
3.10	Expected costs and actual costs with different margins of safety. . . .	85
3.11	Generalization performance of AM-E.	88
4.1	Overview of the Energy-aware Planning Framework. It integrates an MDN energy cost model, a DRL route optimizer, and a risk assessment block. The DRL optimizer is adapted with an MBMS route construction strategy with the help of the energy model and the risk assessment.	100

4.2	Three steps for the Model-based Multi-Sampling Route Construction Strategy.	101
4.3	The mixture density network for predicting the distribution of the drone's energy cost, given the input including the wind distributions and the payload weight.	103
4.4	Energy consumption dataset generation.	108
4.5	Energy prediction via MDN for four typical cases. Results obtained by MDN and Monte Carlo simulations are plotted in the form of probability densities functions and cumulative densities respectively. Four cases are simulated in the presence of winds from different directions: $0deg$ (tailwind), $90deg$, $180deg$ (headwind), $270deg$. The wind velocity v_w and direction θ_w follow: $v_w \sim \mathcal{N}(5m/s, 1m/s)$, $\theta_w \sim \mathcal{N}(\bar{\theta}_w, 30deg)$	110
4.6	Simulation results in terms of overall cost, energy cost, task fail rate, and energy depletion rate, for solving drone delivery problems with different battery safety margins, and standard deviations of wind speed and wind direction. Results are compared among energy-aware Or-tools, stand-alone DRL planner, DRL planner (sample128), and DRL planner (EaPF).	115

List of Tables

1.1	Summary of mission planning workflows in different UAV applications and their key research challenges	7
2.1	Complexity comparison	48
2.2	Generic parameter values or distributions used to generate instances of drone delivery. N_{small}, M_{small} are sets of customer counts and drone counts for small-scale simulations. N_{large}, M_{large} are sets of customer counts and drone counts for large-scale simulations. $\mathcal{U}(a, b)$ denotes a discrete uniform distribution on integer interval $[a, b]$	50
2.3	Parameter distributions related to time windows. $\mathcal{U}(a, b)$ denotes a discrete uniform distribution on integer interval $[a, b]$; $\mathcal{U}\{x_1, \dots, x_n\}$ denotes a discrete uniform distribution on finite set $\{x_1, \dots, x_n\}$; and $\mathcal{B}(p)$ is a Bernoulli distribution of parameter p	51
2.4	The distribution of drone speed	51
2.5	Total task scores: Expected vs. Actual	52
2.6	Running time of three methods for solving 1000 validation instances of each problem size	52
2.7	Comparison of cumulative rewards for solving deterministic drone delivery problems with time windows of different dimensions. The value is averaged over 1000 randomly generated instances.	55
2.8	Comparison of the total coordinating time required by the proposed method and the original CBBA for solving 1000 validation instances of drone delivery problems of different dimensions.	55
2.9	Comparison of cumulative rewards for solving stochastic drone delivery problems with time windows of different dimensions. The stochastic feature of deliveries is emulated by sampling the drone speed from a normal distribution defined by different variances. Larger variances indicate higher randomness. The value is averaged over 1000 randomly generated instances.	56

3.1	Results for solving TSPs using various models. Results are cited from each reference. Note that for the Graph embedding model, results are for 15-20, 40-50, and 50-100 node graphs.	81
3.2	Total trainable parameters of models and running time for one epoch*.	82
3.3	Comparison among AM, AM-E and heterogeneous AM*.	83
3.4	Costs and computing time for solving deterministic DDP-Rs	84
3.5	Costs and computing time for solving stochastic DDP-Rs	86
3.6	Costs using AM-E models trained with different dimensions	88
4.1	Drone specifications	107
4.3	Running time for the MDN energy predict model and Monte Carlo simulations	108
4.2	Key parameters for MDN model and dataset. $\mathcal{U}(a, b)$ denotes a uniform distribution on interval $[a, b]$	109
4.4	Parameter settings for drone delivery instances. $\mathcal{U}(a, b)$ denotes a uniform distribution on interval $[a, b]$; $\mathcal{U}\{x_1, \dots, x_n\}$ denotes a discrete uniform distribution on finite set $\{x_1, \dots, x_n\}$	111
4.5	Parameter settings for wind modelling.	112
4.6	Route planning results for drone deliveries in the presence of stochastic winds	116

Abbreviations

AM	Attention Model
AM-E	Attention Model with Edges
A2C	Advantage Actor and Critic
CBBA	Consensus-Based Bundle Algorithm
CPP	Coverage Path Planning
DDP-R	Drone Delivery Problem with Recharging
Dec-MDP	Decentralized Markov Decision Process
DES	Discrete Event Simulation
DL	Deep Learning
DMG	Diminishing Marginal Gain
DRL	Deep Reinforcement Learning
EaPF	Energy-aware Planning Framework
EGAT	Edge-featured Graph Attention Network
EVRP	Electric Vehicle Routing Problem
GA-Dec-MDP	Group-Aggregation Decentralized Markov Decision Process
GAT	Graph Attention Network
GCN	Graph Convolutional Network
GM	Gaussian Mixture
GP	Gaussian Process
GVRP	Green Vehicle Routing Problem
ITS	Intelligent Transport System
JMTPD	Joint Multi-process Probability Distribution
MAS	Multi-Agent System

MBMS	Model-Based Multi-Sampling
MCKP	Multi-Choice Knapsack Problem
MDP	Markov Decision Process
MDN	Mixture Density Network
M-MDP	Multi-agent Markov Decision Process
MRTA	Multi-Robot Task Allocation
MTVRP	Multi-Trip Vehicle Routing Problem
PDP	Pickup and Delivery Problem
POMDP	Partially Observable Markov Decision Process
POSG	Partially Observable Stochastic Game
RL	Reinforcement Learning
SAR	Search and Rescue
SG	Stochastic Game
SI-Dec-MDP	Sparse-Interaction Decentralized Markov Decision Process
SoC	State of Charge
TCN	Temporal Convolutional Network
TI-Dec-MDP	Transition-Independent Markov Decision Process
TSP	Travel Salesman Problem
UAV	Unmanned Aerial Vehicle
VRP	Vehicle Routing Problem

Notations

$\mathcal{A}g$	the set of all agents
$\mathcal{T}$	the set of all tasks
$\mathcal{Q}$	the set of customer requests
$\mathcal{Q}_p$	the set of pickup requests
$\mathcal{Q}_q$	the set of delivery requests
$\mathcal{G}$	the set of renewable facilities
$\mathcal{D}$	the set of depots
$\mathbf{M}_i$	the MDP model of agent i
τ_i	the decision epoch of agent i
$\mathcal{S}_i$	state set of agent i
$\mathcal{A}_i$	action set of agent i
$p_i(\cdot)$	probability density function
$r_i(\cdot)$	reward function
x_{ij}	a binary variable indicating whether task j is assigned to agent i
$\mathcal{N}_{\mathcal{T}}$	a collection of disjoint task sets
π_i	the policy for agent i
J_i	the overall score of agent i
$V(\cdot)$	value function
$E[\cdot]$	expectation operator
A	advantage function
$\mathcal{L}$	loss function
c_{ij}	the marginal score of task j for agent i
$\mathbf{p}_i$	the task sequence of agent i

$bl(\cdot)$	baseline function
θ, ω	parameters of the policy network and the critic network
$R(\cdot)$	accumulative reward
e	energy cost
P_{aero}	total aerodynamic power required for drone flight
$P_{parasite}$	parasite power for drone flight
$P_{induced}$	induced power for drone flight
$P_{profile}$	profile power for drone flight
$P_{gravity}$	power required against gravity
P_{hotel}	hotel power (i.e., the power required to supply internal electronics)
η	power efficiency
v_w	wind speed
v_g	ground speed
m_{pack}	package weight
χ	course angle
ψ	heading angle
ψ_w	wind course angle
$\mathcal{N}(\mu, \sigma^2)$	normal distribution with mean value μ and variance σ^2
μ_{v_w}	mean value of wind speed
$\sigma_{v_w}^2$	variance of wind speed
μ_{θ_w}	mean value of wind course angle
$\sigma_{\theta_w}^2$	variance of wind course angle
$P(\cdot \theta)$	the probability defined by uncertainty parameter θ
$D_{KL}(p \parallel q)$	Kullback-Leibler divergence between distribution p and distribution q
$\cup$	union (set operation)
$\cap$	intersection (set operation)
$\setminus$	subtraction (set operation)
$ \cdot $	cardinality operation (set operation)
$\oplus$	append the right term to the end of the left term (list operation)
$\Rightarrow$	the left term derives the right term (logical operation)

Chapter 1

Introduction

1.1 Background and Motivation

UAVs, also known as drones, have received momentous attention in a great diversity of disciplines due to their mobility, flexibility, and rapid deployment. Its applications are expanding across a variety of civil scenarios, including routine patrol [6], traffic surveillance [7, 8], search and rescue [9, 10, 11], incidence response [12], infrastructure inspection [13, 14], precision agriculture [15, 16], forest fire monitoring [17, 18] and logistics [19, 20, 21, 22]. Some of these are illustrated in the vision from NASA, shown in Figure 1.1. To date, UAVs have emerged as a highly autonomous platform to collect information, and a competitive alternative to existing transport solutions, with advantages including effortless deployment, agility, cost efficiency, scalability, and effective evaluation of real-world situations at any time and anywhere.

The expanding use of UAVs has motivated the development of multi-UAV systems [23, 24], as the multi-UAV systems could obtain more advantages compared with single-agent systems, e.g., increased flexibility, enhanced resilience and reliability [25], simultaneous broad area coverage [26] and parallel task execution. In multi-UAV systems, UAVs are designed with various configurations, sizes, ranges, and weights, carrying a wide range of payloads including communication gadgets, navigation equipment, sensors, and cameras [27]. Those vehicles coordinate to overcome individual limitations and accomplish complicated tasks.

Despite the benefits of its distributed nature, the multi-UAV system, as an instantiation of the MAS, faces significant challenges in terms of system operation. Firstly, resources (e.g. UAV itself or sensors onboard) are distributed throughout the net-



Figure 1.1: NASA's vision for civil applications of drones [1].

work, which complicates the system operation, making the proper allocation and scheduling of these resources key to maximize the advantages of multi-UAV systems. Second, common situation awareness across the system is unlikely to happen, as the information obtained from individual sensors is distributed, and not able to be shared instantaneously and synchronized across the network. Inconsistent understandings of the environment may cause conflicted behaviours among agents degrading the utilization of resources or even failing on tasks [28]. Therefore, coordinated mission planning is regarded as a vital tool for the successful functioning of multi-UAV systems. Mission planning of multi-UAV systems is constituted by a series of decision-making problems including mission refinement, resource allocation, routing, scheduling, and planning. It performs either in a centralized or decentralized manner. Centralized methods [29, 10] achieve the consensus of information and actions by establishing a central decision entity, which requires highly reliable and efficient communication links and comes under the risk of node vulnerability. Multi-agent mission planning is thus preferably performed in a decentralized manner [30, 31], where decisions are made locally and are resilient against dynamic and asynchronous communication networks. However, this implementation further complicates the problem by requiring the resolution of conflicts between the overall goal and the interests of individuals. The situation becomes even more challenging when limited knowledge about the environment and team members exists.

Distinct from the general MAS mission planning, operating on the UAV platform may place additional considerations in terms of vehicle physical restrictions, dynamic network topology, and stochastic operating environment, as shown in Figure 1.2.

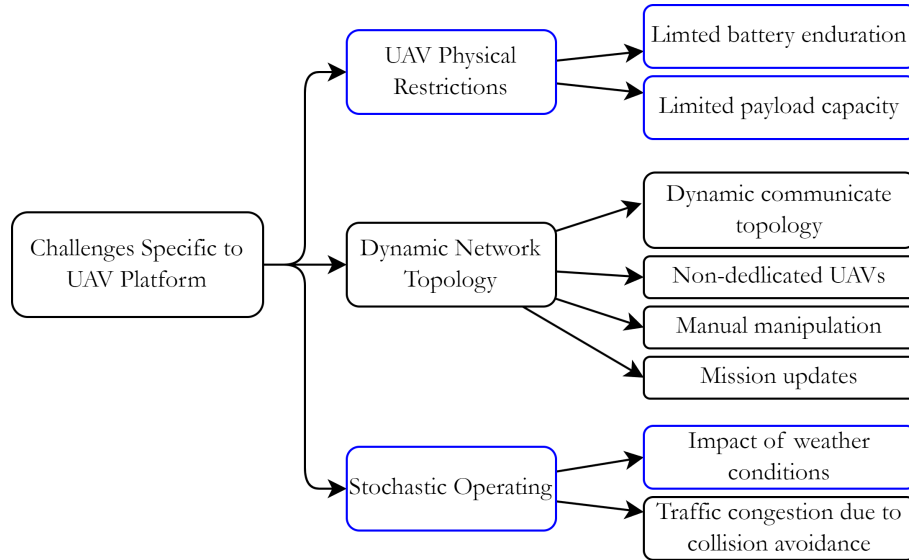


Figure 1.2: Representative challenges for mission planning specific to multi-UAV systems. Coloured bricks indicate main concerns of the thesis.

Vehicle Physical Restrictions: While drones have great characteristics like easy deployment and cost efficiency, their light-weight design also denotes a very limited battery endurance and load capacity [27, 32], which restrict the flight range of UAVs and pose further challenges to the mission planning. First, effective energy management is necessary for battery-powered UAVs. Reliable monitoring and management of the battery use can help UAVs to better achieve their mission and prevent potential loss and damage [33, 34]. Second, load capacity constraints limit computational, communication and sensory resources in each UAV platform. A computing-efficient mission planning algorithm should be developed to reuse UAVs by making multiple trips and performing functional coordination of heterogeneous UAVs, which makes its mission planning more complex.

Dynamic Network Topology The high manoeuvrability of aerial vehicles forms a quickly changing communication topology of the system [35]. With the fluid networking topology, the instant synchronization of situational awareness and actions is unlikely to take place, which drives the development of asynchronous mission coordination strategies for multi-UAV systems. Moreover, mission configuration updates could also cause a change in the network topology, as a consequence of agent failures, manipulation from operators, or mission updates. Therefore, the mission planning algorithm must have the adaptability to dynamic network topologies.

Stochastic Operating Environment Unlike ground-based vehicles, the flight of drones is significantly affected by weather conditions [36, 37], especially the wind direction and speed as they directly impact the kinetics of the drone. As weather is by its nature uncertain and unpredictable, planning for UAVs shall accommodate its stochastic state transitions in terms of uncertain time cost and energy consumption. Other stochastic factors might also impact UAV's operation including the traffic congestion due to collision avoidance [38], and noisy and deviated information generated from sensors.

To summarize, mission planning facilitates the effective execution of multi-UAV systems and there is a fact that the current mission planning tools have limited capabilities when transferring to multi-UAV platforms. Thus, this thesis is motivated to investigate the decentralized mission planning techniques for multi-UAV systems, while considering the stochastic system transitions, UAV's physical restrictions, and distinctive mission requirements (Figure 1.2). The investigation is mostly conducted and validated in the context of drone delivery applications throughout the thesis. The developed techniques are transferable to other multi-UAV civil applications and even other multiple-robot systems.

The remainder of this chapter first gives a review of the taxonomy, current frameworks, and technologies of multi-UAV mission planning systems, and then identifies four key research challenges.

1.1.1 Multi-UAV Mission Planning Architecture and Workflow

Multi-UAV planning, as an instantiation of multi-agent planning, involves designing plans for a group of agents (specifically, UAVs in this context) that generates an ordered action sequence that, when successfully executed, achieves goals for either individual agents or the entire system. Typically, multi-agent planning systems encounter a broad range of situations across these three dimensions [39]:

- **From independent to strongly related:** Dependence always indicates the cases where no resources are shared among agents, and no dependencies exist during the task execution. Adversely, strongly related multi-agent systems will perform actions in a joint manner and resources are shared within the team.
- **From cooperative to self-interested agents:** In cooperative multi-agent

systems, agents share the same global goal, to which the team member is motivated to sacrifice their own interest when there is a conflict, e.g., robots in the RoboCup. On the other hand, in certain situations, agents involved may prioritize optimizing their individual benefits exclusively. For instance, companies in a supply chain may solely focus on maximizing their own profits.

- **From no communication to reliable communication:** When operating in unfriendly environments, agents are unable to communicate during the execution phase, necessitating all coordination to occur beforehand. Conversely, in other scenarios, agents are connected through dependable and effective communication channels.

In this thesis, we focus on proposing an efficient decentralized multi-UAV planning scheme that can deal with the planning problem for a group of agents where there is a weak dependency on each other's actions and goals, cooperate operation, and access to efficient and reliable communication. To make it more specific, the entire research falls under the purview of studying decentralized decision-making architecture, assuming ideal inter-UAV communication without any packet loss or delay. Nevertheless, the proposed algorithms demonstrated resilience against asynchronized communication network topologies.

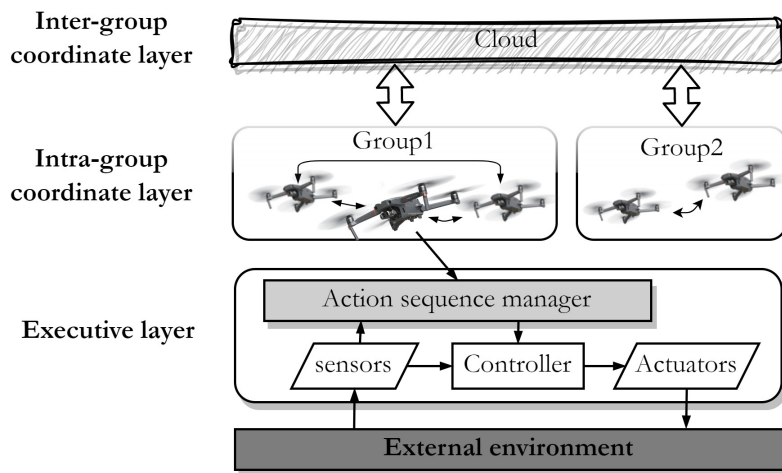


Figure 1.3: Three-tier architecture of mission planning for multi-UAV systems

As a comprehensive topic that integrates multiple disciplines, mission planning for multi-agent systems mostly presents in the form of hierarchical architectures and entails a succession of sub-procedures [40, 41, 42]. For example, the distributed decisional node of UAV proposed in [41] encompasses an Executive layer which acts as an executive using UAVs' own sensory input and actuators, and a Deliberative

Layer providing with higher levels of decisional capabilities dealing mission refinement and coordination activities. One more layer is presented in a heterogeneous MRS architecture [43] to further enable the connection among groups of robots. This highest layer is built on the cloud, which stores the global data of all robots and tasks. Agents communicate via the cloud and utilize any computation or information available in the cloud. This three-tier architecture is depicted in Figure 1.3.

Following the hierarchical structure from top to bottom, a workflow (shown in Figure 1.4) is suggested in Reference [43] and followed by most multi-agent planning systems. Four key procedures are identified in the workflow: (1) task decomposition, which involves breaking down complex tasks into a series of executable sub-tasks based on their attributes. (2) formation coalition, referring to the creation of agent alliances; (3) task allocation/re-allocation, which is the process of assigning sub-tasks to coalitions of agents for execution; (4) task planning and control, where the agent performs a series of actions on the environment to complete the assigned task.

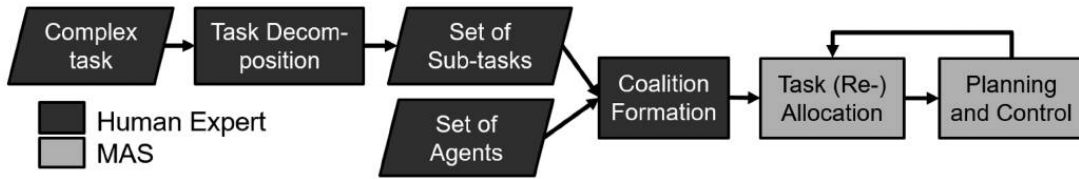


Figure 1.4: Workflow for the multi-robot system proposed in [2]

Notably, not all stages of the multi-agent planning procedure must be included in every situation. For example, if the task has no dependencies and can be completed entirely by one agent, task decomposition is unnecessary. Some methodologies might also integrate different stages. Agents can coordinate on their task lists while constructing their individual plans, i.e., conducting a combination of task allocation and planning [44]. Some systems simultaneously perform coalition formation and task allocation, e.g., IQ-ASyMTRWE proposed in [45] integrates both problems to address a wider range of complex tasks.

In general, any selection and combination of the four phases can be viable, depending on the problem, suggesting a diverse range of problem categories. The following subsection explores its various implementations to handle issues in real-world UAV civil applications.

1.1.2 Existing Multi-UAV Mission Planning Workflows

With the rising interest in using multiple UAVs for civil applications, various workflows of mission planning to coordinate the flight of UAVs have been suggested. Table 1.1 summarizes existing literature in terms of multi-UAV mission planning workflows in four typical applications, including drone delivery, search and rescue, multi-target surveillance, and large-field monitoring. It is followed by a more detailed review, which is constituted by a discussion of their research concerns, system workflows, and key challenges associated with the discussed applications.

Table 1.1: Summary of mission planning workflows in different UAV applications and their key research challenges

UAV application	Reference	Sub-procedure				Key research challenges
		Task decomposition	Coalition formation	Task allocation	Planning and control	
Drone delivery	[46, 47]				✓	Battery management; Efficient routing optimization
	[48, 49, 50, 51, 3]			✓	✓	
Search and Rescue	[10]				✓	Explore in a communication-limited environment; Coupling between the task assignment and UAV operating mode switch
	[52]			✓		
	[53]	✓		✓		
	[54, 55]			✓	✓	
Multi-target surveillance	[56, 57]			✓		Information fusion; Active tracking with uncertainties.
	[58, 59, 60, 61]			✓	✓	
Large-field Monitoring	[62, 63, 64, 23]				✓	Energy efficiency; Cooperative coverage path planning

Drone Delivery

Leading online retailers such as Walmart, DHL, Google, and Amazon have shown significant interest in utilizing UAVs [65, 66, 19]. This interest is attributed to the increasing popularity of online shopping, which has resulted in a surge in customer purchases and a growing demand for last-mile delivery services. In response, retailers are exploring options to enhance delivery services. UAVs have emerged as a promising solution in this context, with their speedy and innovative designs en-

abling efficient and eco-friendly last-mile delivery [67]. In addition to delivering commercial products, drone delivery has also attracted great attention from health-care for its quicker delivery in urban areas and the access to hard-to-reach places [68, 69]. Medical services now heavily rely on ground-based transportation with long and unpredictable Turn Around Time [70]. It may cause compromises for time-critical situations and becomes exacerbated in pandemic situations such as the recent COVID-19, during which the strain on medical services and response becomes evident and difficult to manage. Therefore, medical organizations are increasingly resorting to logistics based on unmanned aviation, which provides speedy, efficient, and reliable transports in a flexible and repeatable way [71].

Although drone delivery holds great potential, unique challenges, such as the limited energy budget, unpredicted payload weight, and weather-sensitive property, hinder its way to real-world implementations. To that end, researchers have built innovative planning frameworks specifically for drone delivery scenarios. A flying sidekick travelling salesman problem is proposed to allow delivery trucks to deploy drones, enabling combined solutions for last-mile deliveries [46]. Combined delivery is also studied in [47], where a drone coordinates with a truck to efficiently deliver packages to a set of dispersed customers while subjecting to precedence constraints. This kind of hybrid implementation relies on a fleet of trucks and is still restricted to the ground traffic flows. Multi-trip VRP considers the limited carrying capacity of drones and compensates for it by reusing drones for multiple trips [48]. Subjecting to the energy budget, some planning algorithms for drone delivery include an energy consumption model, which is established as a function of payload and mostly assumed to be linear for simplicity [48, 72]. However, dynamic weather conditions (like winds) will cause an unexpected energy-consuming rate that may lead to a delay in delivery services or even a loss of drones. Francesco et al [49], thereby define a particular variant of the VRP, namely the drone Mission-Feasibility Problem, where the changes of energy consumption due to winds are reflected by dynamic edge costs of a routing graph. To address the Mission-Feasibility Problem, three planning algorithms are suggested. In [50], a wind-resistant mission planning strategy is presented to find a collision-free plan of delivery mission maximizing order fulfilment. The delivery mission is composed of a sequence of UAV multi-trip sub-missions, each of which is planned within a certain weather forecast window and has to terminate before the end of the time horizon.

Existing research on mission planning for drone logistics is mostly built on the generic VRP frameworks and innovatively develops its variants for particular con-

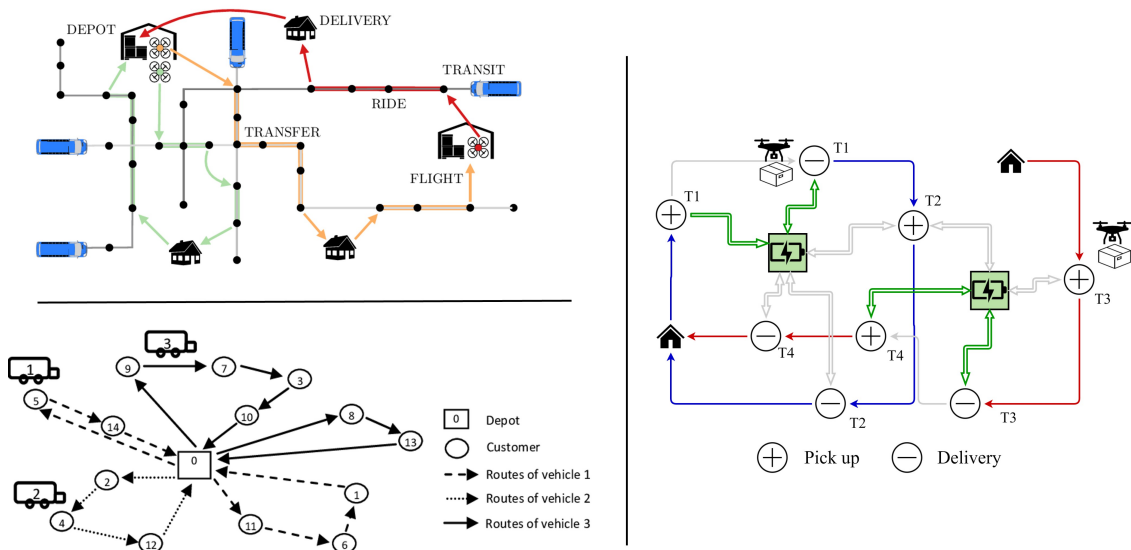


Figure 1.5: Three different drone delivery configurations: (top left) multi-drone delivery framework using transit networks [3]; (bottom left) multi-trip vehicle routing problem [4]; (right) multi-drone delivery framework with recharging services [5].

siderations. Among these studies, one certain subject is pursued as the energy-constrained delivery planning [73], which arises from the UAV's nature of limited battery capacity. Estimating energy requirements for a delivery plays an essential role in energy-constrained delivery planning. It calls for high efficiency as a large number of iterations between the route planner and the drone energy prediction model are involved in energy-aware planning, which can cause a huge increase in computation amounts and a dramatic decrease in planning efficiency.

Search and Rescue

Numerous projects have explored the potential of UAVs in SAR operations, UAVs are added to the toolkit of disaster responders for locating and rescuing missing individuals in complex environments such as fire, wilderness, or flood situations. For example, COMETS leverages a diverse set of UAVs with varying types and capabilities to aid first responders in locating and monitoring wildfires [74]. CLOSE-SEARCH, on the other hand, focuses on SAR operations in unfamiliar terrain [75], while RESCUECELL seeks to address logistics and transportation considerations for UAV systems in disaster-stricken areas [76]. The use of drones significantly increases the success of the mission, as they can rapidly cover a large area and provide situational awareness in a quick way [77]. It is desirable to develop cooperative SAR planning for multiple UAVs, such that UAVs can coordinate to explore kilometre-

scale areas to discover casualties as quickly as possible.

Existing multi-UAV SAR planning mostly follows the workflow in Figure 1.6 [78], with key components including: 1) Searching area decomposition, which is the division of the mission area into small fractions. 2) Task (re-)assignment, that allocates or re-allocates subareas and founded victims among searcher and rescuer aircraft. 3) Path planning, the generation of pathes including the way-points for searching. 4) UAV operating mode shift, where the UAVs switch between different operating modes when a rescue is activated or finished. In addition to the workflow, some search and execution applications also consider the coalition formation, particularly for the search and prosecute operation [79], as the detection and attack on a target require various resources which is carried by different vehicles due to the limitation of carrying capacities.

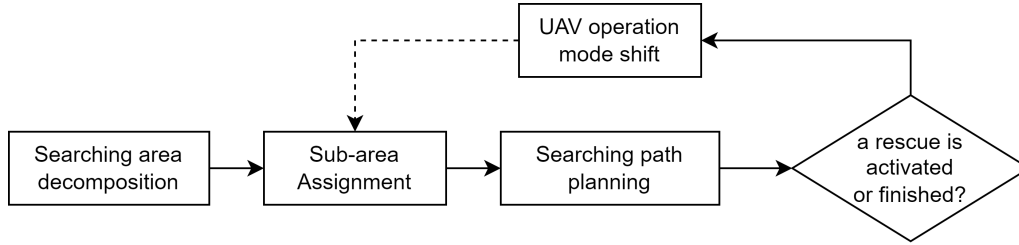


Figure 1.6: Workflow of multi-UAV systems in search and rescue.

Following the SAR planning workflow, the searching area is firstly partitioned according to the mission geography features or the clustering of victims. The layered search and rescue algorithm [53] decomposes the mission area into a set of incremental regions, the centre of which is specified by its coordinates and the priority is degrading from the central layer towards the outer layers. After area decomposition, sub-areas are allocated to UAVs, where the vehicles start to generate their paths for searching the allocated area. Path planning in SAR has been investigated for various objectives, especially in consideration of communications. Collaborative localization and mapping in a communication-limited environment is exploited in [10] to enable multiple UAVs operating for SAR under the forest canopy. To enable information dissemination throughout the UAV network, a centralized path planning strategy is proposed in [54], with a mission goal to simultaneously inform and connect. Once a victim is discovered during a search, the UAV will shift to another mode fitting into the current system requirement. For example, in [55], UAVs act as relay nodes by switching to propagating mode to form a multi-hop streaming link with the ground control station once a victim is found. Switching strategies are mostly defined prior to the mission, e.g., the searcher will be shifted from the adjacent layer with lower

priority and become a rescuer once a survivor is located [53]. A more advanced auction-based algorithm is used in [52] to optimize the role change process. To be specific, once a missing person is found, the searching aircraft will announce the task for bids, and allocate it to the UAV with the highest bid.

To summarize, mission planning for multiple UAVs operating in SAR normally follows four key procedures: area composition, task assignment, path planning, and mode shifting. Most existing SAR systems employed a central mission controller linking with the UAVs through direct communication or a Cloud server [55]. During the planning, two particular topics have attracted the attention of researchers. First, UAVs collaboratively explore the searching area in communicate-limited environments, such as the multi-UAV system for exploration in an unknown forest or above a devastated area. Second, the assignment of SAR sub-tasks is coupled with the process of switching between different operating modes of the UAV. Existing resolutions mostly rely on human intervention, so that there is still room for autonomy and performance enhancement.

Multi-Target Surveillance

Surveillance usually indicates the monitoring and/or tracking of a set of objects, with typical applications spreading across traffic monitoring, traffic control, border patrol, and general surveillance of the population [80]. In recent years, UAVs have been increasingly employed in surveillance to reduce labour costs by completing tasks efficiently and safely [81, 82, 83]. Networked UAVs, equipped with different types of sensors, collaboratively collect a wide range of data and might actively track moving targets with their high manoeuvrability. To gather the most information, cooperative mission planning for multi-UAV multi-target surveillance is a necessity that involves the decision-making for target selection and the decision of the best action to track them.

To address the cooperative multi-target surveillance problem, key technologies have been developed including surveillance target assignment and multi-target tracking [80, 59]. Surveillance task assignment is a combinatorial optimization problem, the aim of which is to pair the surveillance task with UAVs, to maximize the system-wide performance metrics under a given set of constraints. An efficient decentralized task allocation algorithm is proposed in [56] based on the lazy sample greedy to deal with task assignment for large-scale surveillance missions. Considering the dynamic surveillance scenario, a holistic reputation model is used to decide the pairing be-

tween the UAVs and the targets, enabling the dynamic multi-target tracking and data collection [57]. A conflict exists between maximizing the surveillance performance and minimizing the energy usage while planning for multi-UAV multi-target surveillance. A bi-objective genetic algorithm is thus employed in [84] to analyse the trade-offs between these two metrics.

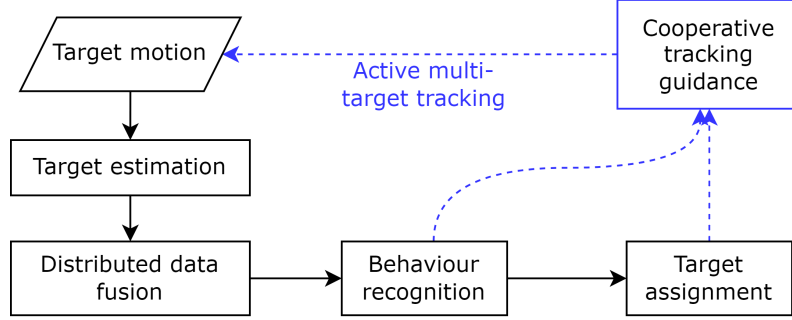


Figure 1.7: The workflow of active multi-UAV multi-target surveillance.

Active target tracking (as shown in Figure 1.7) estimates the motion of the target from sensory data, and actively controls the agent to gather more data. As operating on a manoeuvring platform, more challenges are brought to deal with the joint problem of multi-target information fusion and cooperative tracking decision-making. Uncertainties also fill in the surveillance mission due to factors such as mobile targets, imperfect sensors and dynamic models of UAVs. Therefore, POMDPs are established in [58] to model the multi-robot surveillance with uncertainties. Behaviours defined by the POMDPs are then assigned dynamically to the robots during the execution of the mission [58]. Further, the factorization of the POMDP model is exploited in [59] to apply the sampling approach to multi-target surveillance without increasing the problem complexity. Similarly, a systemic design for multi-target tracking is presented in [60], integrating a distributed information fusion strategy and a time-efficient auction algorithm. The state of targets is represented by the JMTPD and estimated by the distributed information fusion strategy. Based on the consistent JMTPD, the tracking actions are then decided by auctioning the bid upon the reward value of POMDPs. When the UAV swarm consists of homogeneous drones, a decentralized DRL algorithm is proposed in [61] for the cooperative tracking guidance, which learns a cooperative policy by using a sharing reciprocal reward that enables collaboration among a large-scale group of homogeneous UAVs.

To sum up, planning for static multi-target surveillance has mainly focused on solving the task allocation problem, while mobile surveillance presents a new challenge involving both information fusion and tracking guidance. Despite extensive study of task allocation, there are still some gaps in surveillance task planning, e.g., defining

a reward function for UAV-task pairs in the presence of stochastic factors, and efficient task re-allocation in dynamic surveillance scenarios. Active target tracking is commonly modelled as POMDPs to account for uncertainties, e.g., mobile targets, imprecise estimations, and manoeuvring UAVs. Existing solutions have sought to reduce complexity by factoring POMDPs into local decision-making schemes, but this approach relies heavily on reliable network communication to achieve consensus among local decisions.

Large-Field Monitoring

Achieving complete coverage of a specific area is the primary objective of monitoring. With the advantages of rapid deployment and aerial perspective, UAVs' use in monitoring has become increasingly prevalent, particularly in large-field monitoring applications such as precision agriculture, fire monitoring, archaeology, and infrastructure inspection. Monitoring applications have overlapping purposes with the initial stages of search and rescue and multi-target surveillance, which aim to gather as much information as possible over an unknown region. The main distinction is that after the UAV obtains valuable information, it continuously monitors the rest area.

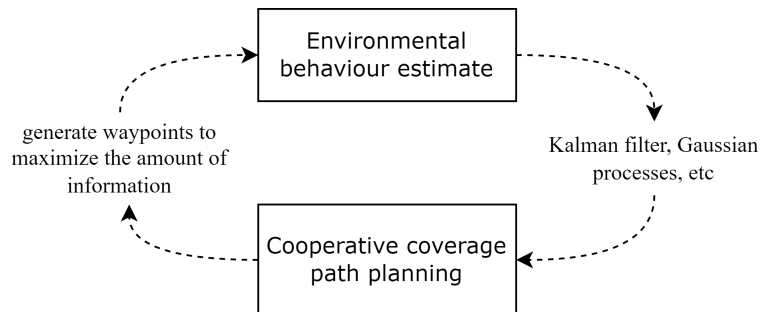


Figure 1.8: Information-driven loop of multi-UAV cooperative monitoring.

In multi-UAV monitoring, mission planning is primarily driven by information, creating a loop (as shown in Figure 1.8), where the UAVs gather data on environmental behaviours, which in turn affect decisions on the subsequent flight path. First, a model of the environment is set up prior to capturing the target variable of interest. Occupancy grids are the mostly-used representation for spatial sensing with uncorrelated measurements [85]. The occupancy mapping model is easy to employ and offers high computational efficiency. However, many natural characteristics exhibit complex inter-dependencies. To handle such a relationship, GP are applied to collect data accounting for map structures and uncertainties [86, 87, 62]. The

area to be monitored is discretized in [63] for simplicity and the unmeasured state is estimated by a Kalman Filter subject to the spatiotemporal dependencies of the observed phenomenon. For more complex scenarios, such as wildfire monitoring, an empirical wildfire model is used to describe the propagation behaviour of wildfire [64]. Based on the model, a Kalman filter-based data assimilation method is designed for behaviour estimation.

Based on the estimate of environmental behaviours, cooperative CPP problems are optimized to gather the maximum amount of information. For example, driven by information, the flight path of UAVs in [63] is calculated by casting the path planning problem into an orienteering problem and maximizing the Fisher information matrix of a Kalman filter to avoid redundant data. The cooperative coverage problem for wildfire monitoring is modelled as a distributed optimization problem and then solved by adapting the game theory [64]. For a comprehensive review of the cooperative CPP algorithm, the reader could refer to [88], as well as [23] which places additional emphasis on the energy efficiency in CPP problems.

In summary, cooperative mission planning of multi-UAV monitoring is mainly driven by information and is generally composed of information gathering, environment mapping, environmental behaviour estimation, and cooperative coverage path planning. Limited flight endurance is the major challenge in deploying UAVs for extensive coverage missions. To address this concern, various strategies have been proposed, e.g., minimizing path length, mission time, and turning manoeuvres to conserve energy. Additionally, researchers have validated the significance of considering wind direction and intensity in UAV path planning, with the optimal path being aligned vertically with the wind direction and turning manoeuvres executed against the wind direction [89]. This approach can be combined with the previous method to achieve even greater energy savings.

1.1.3 Key Technologies in Multi-UAV Mission Planning

Key techniques in multi-UAV mission planning cover a wide range of solutions to different phases of the problem (Figure 1.4). However, few research has considered the complete autonomy of four stages in the multi-agent mission planning or even reached the third level of automation [43], where coalition formation, task allocation, and task execution were performed by an MAS without any human interference. For this reason, this section will only focus on reviewing the latest technology developments regarding the last two processes of solving a multi-UAV planning problem,

i.e., task allocation and decentralized planning.

Task Allocation

Task allocation assigns tasks to a team of agents while optimizing some overall performance metrics. The characteristics and complexity present in an MRTA problem are determined by the requirements of the specific domain of concern [90]. A framework proposed in [91] is now considered a standard for describing task allocation problems. It proposes three axes to characterize an instance of task allocation: 1) single-task robots versus multi-task robots; 2) single-robot tasks versus multi-robot tasks; 3) instantaneous assignment versus time-extended assignment. Based on this taxonomy, Korsah et al. [90] add an extra dimension of task allocation to cover situations with task dependencies such as related utilities and task-coupling constraints. The task allocation studied in this thesis falls into the category of multi-task single-robot time-extended assignments with task dependencies.

Many approaches have been developed to solve the task allocation problem in planning. A comprehensive discussion of centralized approaches to the problem can be found in the book by Burkard, Dell’Amico, and Martello [92]. There are, however, many other methods to assign tasks in a more distributed way, giving the system better scalability and entitling the agents to a higher degree of autonomy and privacy.

Auctioning algorithms are one of the popular algorithms employed in decentralized task allocation problems capable of handling task priorities and heterogeneous tasks or robots [93]. In an auction, each agent makes bids to a task and the task will be assigned by an auctioneer to the agent that attaches the highest value or lowest cost to it. Choi et al [94] implemented the sequential greedy algorithm in a decentralized manner via auctioning, and proposed the CBBA, which is an efficient decentralized task allocation algorithm that guarantees at least 50% optimality. As variants of CBBA, a sampling pre-process and a lazy strategy are integrated into the algorithm for better scalability [56, 95]. These methods show great performance when tackling large-scale deterministic task assignment problems, yet they are not guaranteed to perform well in a stochastic environment. In an effort to retain the outcome with transition uncertainties, Ponda et al. [96] present a robust extension to CBBA using the integral of expected margin utility instead. Later, the same group [97] lifts the requirement of diminishing marginal gain condition by adopting a wrapping function, expanding the algorithm with more general applications. However, it is

at the cost of surrendering all theoretical performance guarantees. Nevertheless, computing these robust scores takes a huge cost of time, making the allocation process cumbersome. [98, 99].

Metaheuristic-based approaches, often nature-inspired, have been widely used in MRTA problems [93]. Metaheuristic-based algorithms usually randomly generate one or multiple task allocation solutions and optimize the solutions throughout iterations. They are accordingly classified as single-solution algorithms and population-based algorithms[100]. Single-solution algorithms aim to modify and improve a single-candidate solution. Typical algorithms include iterated local search [101], guided local search [102], variable neighbourhood search [103], and simulated annealing [104]. Population-based algorithms focus on maintaining and improving multiple candidate solutions. Typical algorithms include genetic algorithms [105, 106], ant colony optimization [107, 108], particle swarm optimization [109, 110], and artificial bee colony algorithm [111]. As a synergistic combination of the auction algorithm and the meta-heuristic approach, a series of stochastic clustering auction algorithms have been proposed in [112, 113], in which the designers are allowed to choose the rate, to adjust the trade-off between the convergence speed and the solution quality. The metaheuristic approach is not problem-specific as it requires relatively few assumptions [100]. By iterating through a large number of feasible solutions, metaheuristic algorithms can often achieve reasonably good solutions with a certain convergence speed [100]. According to [114], the metaheuristic approach outperforms the market-based approach in terms of scalability. However, metaheuristic algorithms cannot provide any approximation guarantee for the solution quality. Sometimes, metaheuristic algorithms may allocate unsuitable tasks to agents and respond slowly to environment changes [115].

The multi-robot task allocation problem is NP-hard, which means it is unlikely to be solved in polynomial time. Consequently, researchers tried multiple approaches to devise practical task allocation frameworks. The first category is built on the basic framework of the sequential greedy algorithm, leveraging the specific assumptions of the problem to enforce tractability. The second series of approaches adopt the common meta-heuristic techniques, which are problem-independent but face a larger computational burden without problem-specific information. In addition, most of the existing task allocation algorithms are developed under the assumption that agents are aware of all the tasks and the exact utility attached to them. However, in practice, new tasks are dynamically introduced and the utility value for the certain task-agent pair is difficult to precisely estimate in some cases especially when the

operating environment is in the presence of stochastic factors [44].

Multi-Agent Planning and Control

Multi-agent planning and control, also known as multi-agent decision-making, is a fundamental component of MAS. It determines a sequence of actions or policies, that should be performed by agents to complete the certain mission goal [43]. Notably, MAS decision-making might either take place after tasks are decomposed into simpler tasks and allocated to individuals, or it interleaves with the phases of coalition formation or the phase of task allocation, during which agents coordinate while constructing their own plans. For example, a simultaneous task allocation and planning method is proposed in [116], which solves the complete problem in a single process via building a joint model of MDP and linear temporal logic.

Various frameworks have been proposed to model and solve multi-agent decision problems, among which typical propositions include MDPs, game theory, and swarm intelligence. While referring to [117] for a more detailed review of multi-agent decision-making models, we will briefly discuss some of the representative techniques, and evaluate them in terms of both policy optimality and the time and space complexity.

MDP describes a discrete-time stochastic control process and provides a decision-making framework for problems where state transitions are random, and outcomes are influenced by decisions of the agent. Solving an MDP is to find a policy that an agent can follow to maximize its expected accumulated reward over time. Benefiting from the ability to handle uncertainties, MDPs have been broadly applied to deal with stochastic planning problems in the real world, and further extended to M-MDPs and Dec-MDPs for multiple agents. M-MDPs are a straightforward extension of MDPs, where a global learner has access to all individual states and learns a joint policy that should be performed by the team [118]. The complexity of the M-MDPs for the worst case is P-complete, which indicates that an efficient algorithm exists and the problem can be solved within a polynomial time [119]. However, M-MDPs are subject to a high communication demand and the scale of the problem as the dimensions of the joint state and action spaces would exponentially explode. Alternatively, Dec-MDP factors the action space for each agent with a local reward function and assumes a joint fully observable environment [118], which reduces the state and action dimensions that need to be handled. When there are 3 or more agents involved, the finite-horizon Dec-MDP problem is proven

to be NEXP-Complete, that implies any algorithm for solving this problem will most likely take doubly exponential time in the worst case [120]. To reduce the complexity of the problem, various assumptions have been made to the observability and communication capabilities of the agents in the problem. For instance, a specific class of decentralized MDPs is identified as TI-Dec-MDP if agent transitions are independent and the collaboration is only tied through a structured global reward function [121, 122]. Characterized by independent transitions and observations, local observability, and joint reward structure, the TI-Dec-MDP simplifies Dec-MDPs to NP-complete and is then optimally solved by the coverage set algorithm. Another simplification is the SI-Dec-MDP that explicitly distinguishes the specific parts of state space in which the agents in the team must coordinate with others [123]. For all agents outside interaction areas, the SI-Dec-MDP degrades to a set of individual MDPs with factorized transitions and reward functions, while in the states of each interaction area, agents are modelled as local M-MDP and communicate freely. Some other methods, such as group aggregation in GA-Dec-MDP [124], are also proposed to reduce the computational complexity.

Game-theoretic models are another framework used to analyze multi-agent decision-making scenarios where each player's choices and outcomes are represented as a game, and the players must make decisions based on their own goals and the expected behaviour of other players. The model takes into account the strategies of other players and the possible outcomes of the game, in order to find the optimal decision for each player. SG [125], as a typical game-theoretic model, is a sequential probabilistic game in which the outcomes of player actions are determined by chance or probabilistic events. SG can be viewed as a generalization of the above-mentioned MDPs. Similar to a policy in MDP, a strategy profile is expected to be obtained by solving SG, which indicates the rule followed by players to select a move. POSGs extend the SG to a partially observable environment, where the payoffs are unknown to the players. Finding a Nash equilibrium strategy for POSG, i.e., a player's best strategy given strategies of others, has been proven to be NP-hard [126], implying that it is unlikely to have an efficient algorithm for solving it. Both heuristic and approximate methods have been proposed for solving POSG. Guo et al. proposed an iterative approach in [126] to eliminate dominant strategies. Hansen and Zilberstein incorporated a generalized version of dynamic programming for POMDP with the elimination of dominated strategies to address POSG problems [127]. While game theoretic models are initially proposed in competitive scenarios, they have also been used in cooperative applications, taking advantage of the capability of modelling others' behaviours [128, 129]. This feature can be especially helpful in cases where

agents cannot communicate with each other. However, when the number of players grows, the computational costs of solving the game model become increasingly prohibitive.

Swarm intelligence, which is inspired by the behaviour of social animals, creates models of the decentralized, cooperative autonomous agents [130]. These models are characterized by the self-organization and distributed behaviour of agents that interact locally and are locally aware [131]. Particle swarm optimization, introduced by Kennedy and Eberhart in 1995 [132], is modelled after flocks of birds and schools of fish. Pigeon-inspired optimization algorithms utilize magnetic field, the sun, and landmarks for path planning [133], while bee colony optimization is inspired by the behaviours of bees, relying on direct communication between agents [134]. Although swarm-based algorithms are known for their robustness, flexibility, scalability, computational efficiency, and fault tolerance [135], their application in MAS is confined as they can only apply to homogeneous swarms or those can be divided into a few clusters of homogeneous robots. More detailed overviews of swarm intelligence studies can be found in recent surveys [136, 137].

Despite notable developments in algorithms such as the aforementioned MDPs, game theories and bio-inspired methods, the decision-making of MAS continues to be riddled with some issues. The first is scalability, especially in heterogeneous MAS, where decision-making algorithms must handle increasingly complex tasks. However, current models can only scale with homogeneous agent teams of limited interaction levels. For example, swarm intelligence algorithms are able to scale to large MAS while subjecting to its homogeneity and restricted interactions. MDPs and game models, on the other hand, can model distinct individuals but do not scale well as the algorithm complexity exponentially increases with the team size. Second, UAVs generally have limited onboard resources, posing a restriction on the capability of computation and data exchange. Planning algorithms should be computationally efficient while carefully considering the interaction cost among agents. Further, decision-making algorithms must possess the capability to adjust to the dynamic environment and deal with the various uncertainties that may arise. Distinct from MDPs that assume a fully observed state, POMDPs and POSGs, are better equipped to handle uncertainties as they account for partial observability and imperfect information. However, it is still essential to consider the generalization ability of these algorithms to situations that have not been encountered during their design and training.

1.1.4 Challenges in Multi-UAV Mission Planning

From the review on the existing mission planning techniques, the following challenges are identified in the development of multi-UAV mission planning systems.

1. *Scalability*

Mission planning algorithms should be scalable, especially for heterogeneous multi-UAV systems. As reviewed, most of the multi-agent planning algorithms that allow heterogeneity, e.g., MMDPs and SGs, are restricted to small-scale problems. The complexity of these algorithms significantly increases with the scale of the problem. Hierarchical approaches with dense local decision-making and sparse global interaction might form a solution for the scaled heterogeneous multi-agent system mission planning.

2. *Efficiency*

The efficiency of mission planning algorithms is considered in terms of both computation and communication. UAVs generally have limitations on computational resources due to size and weight constraints. This necessitates that the algorithm used for real-time decision-making must be computationally efficient. In addition, tightly coordinated tasks demand intensive data exchanges among networked agents, posing challenges to communication efficiency, especially for the large-scale network topology. Given the limited amount of computing and communicating resources, a trade-off between these two resources might be carefully crafted in the mission planning algorithm.

3. *Adaptability and Robustness*

UAVs operating in a dynamic and unpredictable environment pose challenges for their robust mission planning, which indicates the ability of systems to handle uncertainties. As a robust multi-UAV system, it should be capable of accomplishing its mission against uncertainties or any expected event that might occur. The decision-making algorithm is expected to adjust to dynamic and diverse environments they may encounter and exhibit proficiency in dealing with imprecise and incomplete information produced by sensors, as well as non-deterministic actions.

4. *UAV Energy Management*

Distinct from ground-based autonomous robots, energy consumption is an important factor to consider in UAV mission planning due to its limited battery capacity.

To enlarge the flight area and avoid unnecessary return journeys, the deployment of recharging stations is motivated and applied in real-world implementations. Including recharging services in UAV mission configurations forms a more complicated planning problem, where in-advance scheduling of station visits is expected to happen. With the aim of improving planning performance and controlling energy risks, mission planning for UAVs is expected to be accomplished in an energy-aware manner. It involves a large number of iterations between the planner and the UAV energy prediction model which is mostly established on the simulation of aerodynamics and power estimations on the premise of influential factors. Energy-aware planning indicates a huge increase in computation amounts and a dramatic decrease in planning efficiency, posing extra challenges for UAV mission planning.

1.2 Aim and Objectives

The thesis focuses on the development of a decentralized mission planning algorithm for multi-UAV systems, in consideration of the dynamic environments. This thesis aims to address the challenges identified in Section 1.1.4, establishing a generic decentralized mission planning paradigm for multi-UAV systems and validating it in the context of drone delivery scenarios. The intended system will demonstrate resilience against uncertainties, exhibit effectiveness in offering solutions for scaled-up situations, and be capable of planning with energy awareness while incorporating recharging services.

To achieve the aim of this thesis, the following objectives are defined:

- **Enhancing the robustness of mission planning solutions against uncertainties:** The first objective is set as enhancing the robustness of multi-UAV mission planning, aiming for stable performance in the presence of stochastic factors, e.g., unpredictable weather, unexpected UAV failures, and measurement errors generated by the sensor. The enhancement is involved in the phases of both task allocation and decentralized planning. Towards this target, approaches include uncertainty identification and analysis, system modelling, and robust algorithm development and validation.
- **Reducing the computational complexity of multi-agent planning algorithms:** Multi-agent decision-making problems are significantly more complex than single-agent problems as agents must consider the behaviours of

other agents in the system. The complexity will also increase with the dimensions of joint action and state spaces, which grow exponentially with the number of agents. Further, the multi-agent planning problem can become even more complex in consideration of incomplete or uncertain information. Therefore, decreasing the computational complexity of multi-agent planning algorithms by taking advantage of system properties is critical for real-world implementations.

- **Developing an efficient decision-making algorithm for UAV operating with recharging stations:** To enlarge the flight range, increase mission efficiency, and be eco-friendly, we include recharging services in the multi-UAV systems. This implies that the planning algorithm is required to handle the design of recharging strategies, the decision of station visiting order, and the increased system heterogeneity.
- **Constructing the efficient energy-aware planning method for multi-UAV systems:** With the aim to increase the stability of mission execution and avoid the potential loss of vehicles, the mission planning for multi-UAV systems is expected to be conducted in an energy-aware manner. Considering the large number of iterations between the planner and the energy model in the energy-aware planning, an efficient well-formed UAV energy predictive model is required. In addition, the mission planner should have the ability to take the most of the energy predictive value into its planning.

1.3 Contributions

Towards the aforementioned aim and objectives, this thesis develops and validates a decentralized mission planning paradigm for multi-UAV systems, optimizing and stabilizing the operation of multi-UAV systems in the presence of recharging, winds, and mission constraints.

Contributions in this thesis are highlighted by two decision-making algorithms (i.e., the task allocation algorithm and the route planner) and an energy-aware planning framework. Figure 1.9 shows the relations between the proposed methods for cooperative multi-UAV mission planning in the context of drone deliveries. Tasks are firstly allocated among UAV team members via the proposed auction-based decentralized task coordination strategy (Chapter 2) by communicating the estimated

value of task-UAV pairs with neighbour UAVs. Then, a DRL route planner is established for each individual upon an edge-enhanced attention model (Chapter 3) to map the route policies, producing route candidates by interacting with the energy prediction model. The UAV energy cost is modelled using a mixture density network with respect to wind conditions to enable energy awareness and risk assessment in stochastic planning (Chapter 4). After route candidates are generated via the interaction between the DRL route planner and the energy predictive model, a model-based multi-sampling strategy is employed to select the optimal route with the minimum cost while subjecting to a certain risk threshold (Chapter 4). Finally, the next action of the individual UAV is selected as the first node from the optimal route among all the route candidates.

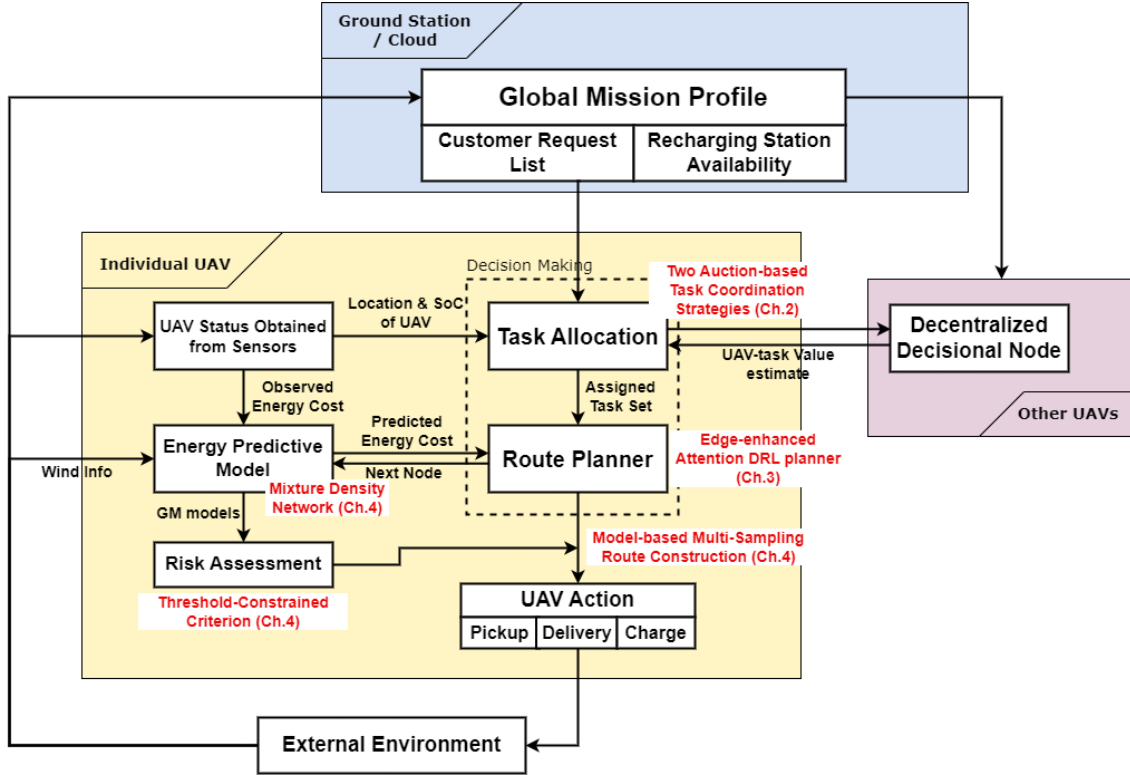


Figure 1.9: Blocked diagram highlighting the relations of proposed algorithms.

The detailed contributions corresponding to each component are:

- **An auction-based coordination strategy for multi-agent stochastic planning with submodular rewards (Chapter 2):** A decentralized auction-based coordination strategy is proposed using a newly formulated score function which is generated by forming the problem into task-constrained MDPs. The proposed method guarantees convergence and at least 50% optimality in

the premise of a submodular reward function. Furthermore, for the implementation on large-scale applications, an approximate variant of the proposed method, namely Deep Auction, is also suggested with the use of neural networks, which is evasive of the troublesome for constructing MDPs. Finally, we demonstrate the performance of the two proposed approaches in the context of drone deliveries, where the stochastic planning for the drone league is cast into a stochastic price-collecting VRP with time windows.

- An efficient planning algorithm for single drone deliveries in the presence of winds and recharging stations (Chapter 3):** An efficient DRL-based planning algorithm is developed to solve the typical drone delivery problem with recharging which is characterized by directional edges and stochastic edge costs subject to wind conditions. The drone delivery is modelled from a practical perspective and is then cast into an MDP over a graph, with the next node chosen according to a stochastic policy based on the evolving observation. An AM-E model is suggested to map the optimal policy via the DRL approach. AM-E comprises a succession of edge-enhanced dot-product attention layers which is designed with the aim of capturing the heterogeneous node relationship for DDP-Rs by incorporating adjacent edge information. In terms of both non-wind and windy cases, extensive simulations demonstrate that the proposed DRL method outperforms state-of-the-art heuristics for solving drone delivery problems, especially at large sizes.
- A model-based energy-aware planning framework for single drones (Chapter 4):** An EaPF framework is constructed to increase the planning robustness against energy consumption uncertainties, which integrates a statistical energy predictive model, a route optimizer, and a risk assessment module. Adaptive to any route optimizer, this framework is demonstrated with the developed DRL planner and produces a more reliable solution for the energy-sensitive application. In terms of energy modelling, an MDN model is established to predict the drone's energy consumption on the premise of wind distribution and load weight. Based on the energy model, a risk-sensitive criterion with respect to violating time windows or the energy constraint is analytically deduced to assess the quality of a route, facilitating solutions of higher robustness. Further, a novel MBMS strategy is proposed to better accommodate the DRL planner into the EaPF, enhancing the planner's robustness, predictability, and reliability. Finally, simulations are conducted in the context of drone delivery problems, respectively validating the effective-

ness of the MDN energy model, the DRL planner, and the EaPF which is incorporated with the MBMS strategy.

1.4 Organization

The rest of the thesis is organised as follows. Chapter 2 presents the developed auction-based multiple-agent coordination strategy, and analyses its optimality guarantee on the premise of submodular rewards. In Chapter 3, an efficient UAV planning algorithm in the presence of winds and recharging stations is proposed and validated in the context of drone delivery scenarios. In Chapter 4, we build a generic energy-aware planning framework, establish a statistical model of drone energy consumption via Gaussian Mixtures, and then validate the framework with the drone delivery case studies. Chapter 5 gives an overall discussion about the proposed planning scheme, that highlights its achievements, benefits and drawbacks. Finally, a conclusion and several future research potentials are discussed in Chapter 6.

1.5 Publications

Published

- [1] Ruifan Liu, Hyo-Sang Shin, and Antonios Tsourdos. Edge-enhanced attentions for drone delivery in presence of winds and recharging stations. *Journal of Aerospace Information Systems*, pages 1-13, 2023. (supporting Chapter 3)
- [2] Ruifan Liu, Hyo-Sang Shin, Min-Guk Seo, and Antonios Tsourdos. Energy-aware route planning for drone delivery systems. In *Proceedings of the 2022 CEAS EuroGNC conference*. Berlin, Germany, May 2022. CEAS-GNC-2022-045. (supporting Chapter 4)
- [3] Ruifan Liu, Hyo-Sang Shin, Min-Guk Seo, and Antonios Tsourdos. Delivery route planning for unmanned aerial systems in the presence of recharging stations. In *AIAA Aviation 2022 Forum*, page 4028, 2022. (supporting Chapter 3)
- [4] Ruifan Liu, Min-Guk Seo, Binbin Yan, and Antonios Tsourdos. Decentralized task allocation for multiple UAVs with task execution uncertainties. In *2020*

International Conference on Unmanned Aircraft Systems (ICUAS), pages 271-278. IEEE, 2020. (supporting Chapter 2)

Preparing

- [1] Ruifan Liu, Hyo-Sang Shin, Binbin Yan, and Antonios Tsourdos, An auction-based coordination strategy for task-constrained multi-agent stochastic planning with submodular rewards. *arXiv preprint arXiv:2212.14624*, 2022. (supporting Chapter 2)
- [2] Ruifan Liu, Hyo-Sang Shin, and Antonios Tsourdos, Energy-aware Drone Delivery Planning: a Model-based Approach with Gaussian Mixtures. (supporting Chapter 4)

Chapter 2

An Auction-Based Task Coordination Strategy

In this chapter, two decentralized auction-based task coordination strategies are presented to address small-scale and large-scale multi-agent stochastic planning problems respectively. Convergence and optimality guarantees are provided for the first strategy on the premise of a submodular reward function. The second coordination strategy, namely Deep Auction, is built as an approximate variant of the first one, extending the solution to the scaling problems. It will be introduced in detail to the neural network architecture, training algorithm, and convergence guarantee. Demonstrations are designed for both strategies via small-scale and large-scale drone delivery scenarios in comparison with other state-of-the-art methods.

2.1 Introduction

Cooperative systems of multiple agents have been of great interest, especially for those operating on unmanned aerial platforms [138]. These systems feature great characteristics including flexible structure, parallel-processing ability, and scalability. Typical implementations have cooperative surveillance, search and rescue [55, 139], border patrolling, etc. Among their various instantiations, there is one specific but widespread category of assigning tasks among team members with a global goal, followed by independent and possibly stochastic task execution. For example, in deliveries of parcels [140], tasks are allocated to individual vehicles, and vehicles deliver their allocated items in sequence, without interference from other

executors. Meanwhile, the delivery may be subject to stochastic travel delays between destinations. Also, in multi-target tracking [60], agents decide the tracking targets and the best action to track based on the estimation of target manoeuvring. The cooperative task assignment and decision-making must be done considering the stochastic state transitions. To summarize, these domains share some common properties: *i)* Agents are functionally independent, except for the task assignment constraint; *ii)* Decisions are made sequentially while taking uncertainty over their consequences into account; *iii)* There is an opportunity for coordination in advance with prior knowledge of task information.

A synergistic combination of two challenges is identified for this type of problem, i.e., task allocation and stochastic planning [141]. Existing approaches are mostly proposed from these two perspectives, either using the deterministic algorithm with greedy selections [142, 143], or modelling problems as resource-constrained multi-agent MDPs [144]. The major drawback of deterministic approaches is that they do not account for transition uncertainties, suffering a performance loss in the face of stochastic operation environments. On the other hand, multi-agent MDPs provide a rich model for representing uncertainties, while not being scalable and hindered by a significantly high computing complexity. Despite the fact that they are still struggling to solve the problem of interest, these approaches have their own merits. Approaches derived from the greedy selection make use of the independence amongst task executions to accomplish the conflict-free assignment efficiently and tractably, while MDPs enhance the results in consideration of transition uncertainties [145, 146]. Motivated by the idea of combining these two methods and built on the previous work in [147], we propose a hybrid algorithm unifying the strengths of these two methods.

To summarize, three key contributions are made by this work to address the above practical task-constrained problems.

- 1) We formulate a generic model for task-constrained multi-agent stochastic planning problems based on MDPs, which explicitly presents task dependencies while taking the stochastic execution process into account.
- 2) A decentralized auction-based coordination strategy is proposed to allocate tasks among team members according to a novel score function defined by the task-constrained MDPs. The proposed coordination strategy is then proved to converge and hold 50% optimality with submodular reward functions.
- 3) An approximated extension to the coordination strategy, namely Deep Auc-

tion, is proposed by utilizing neural networks. The networks get trained by the reinforcement learning algorithm and benefit the computing efficiency facilitating the implementation on large-scale applications.

The remaining part of this chapter is organized as follows: Section 2.2 provides an overview of previous works regarding task allocation and constrained MDPs. Section 2.3 introduces the motivations of research and gives a formal definition of the problem. Section 2.4 presents a detailed explanation of the proposed coordination strategy along with its theoretical analysis. This is followed by an approximate variant of the approach using neural networks in Section 2.5. Section 2.6 compares the proposed approaches with the base methods. Section 2.7 demonstrates the performance of algorithms through simulation in the context of drone deliveries and conducts the comparison with state-of-the-art approaches. Section 2.8 concludes the contribution of this chapter and gives a brief plan for future research.

2.2 Related Works

2.2.1 Task-Constrained Stochastic Planning

Task-constrained stochastic planning, characterized by agent-independent execution and non-overlapping constraints over tasks, can be modelled in a generic form of Dec-MDPs or a specific case of constrained MMDPs [144]. Dec-MDPs provide a standard framework for modelling multi-agent stochastic planning problems, which specify how a stochastic environment behaves when interacting with a set of decision-makers. Despite the powerful modelling ability, the formulation of Dec-MDPs considers all dependencies among agents, which makes the problem NEXP hard and thus significantly impacts the scalability [146]. On the other hand, task/resource-constrained multi-agent MDPs, recognized as weakly-coupled MDPs, can be simplified by decomposing into the individual concurrent process with cooperative actions and merging solutions to form the solution of the original problem [148]. A typical hybrid way to address these resource-constrained MDPs is presented by Meuleau *et al.* [149], who incorporates an online coordination scheme with an offline phase, where a value function is computed to define the expected reward from a specific state. Following a similar idea, Boutilier *et al.* [150] propose the budgeted MDPs defining the value function and policy with a separate budget parameter, and then piece together the solution by solving the budget allocation problem. However, two

drawbacks exist in these approaches: *i)* the amount of computing the value function depends on the dimension of states, which may suffer with a large number of states in the concerning task-constrained problems; *ii)* the online allocation is addressed in a centralized manner with poor scalability and high communication cost. Agrawal *et al.* [141] alternatively formulate task-constrained MDPs in a task-dependent manner, which facilitates its dual decomposition, and then solve it by a greedy heuristic method GAPS. The combination of the greedy heuristic approach and the MDP formalism can also be found in some other applications, such as decentralized planning for non-dedicated agent teams [146]. Its posterior optimality guarantee is exploited with submodular rewards [122].

2.2.2 Multi-Agent Coordination Strategy

In the context of the MDP decomposition, various coordination strategies are implemented to generate a joint policy for a multi-agent system. The strategies take different formulations as the coordination criterion and perform either in a centralized or decentralized manner. Meuleau *et al.* [149] take a centralized coordination strategy, in which actions are sequentially taken in a greedy manner according to the gradient computed by value functions. Boutilier *et al.* [150] instead meets the global resource constraint by solving an MCKP. In this sense, the concerned task constraint can be satisfied by solving a standard task allocation problem.

Task allocation problem extracted from task-constrained stochastic planning is categorized as single-robot multi-task in-schedule assignment problems with task dependencies according to the taxonomy in [90]. It is fundamentally a combinatorial optimization problem and is recognized as NP-hard. One of the most efficient decentralized methods for this problem is proposed in [94], namely CBBA, which adopts a consensus mechanism with the guarantee of at least 50% optimality. As variants of CBBA, a sampling pre-process and a lazy strategy are integrated into the algorithm for better scalability [95, 56]. These methods show a great performance when tackling large-scale deterministic task assignment problems, yet they do not apply to the concerned multi-agent stochastic planning. In an effort to retain the outcome with transition uncertainties, Ponda *et al.* [96] presents a robust extension to CBBA by using the integral of expected margin utility instead. Nevertheless, computing these robust scores takes a huge cost of time, making the allocation process cumbersome and possibly violating the optimality guarantee [96, 99].

2.3 Mathematical Formulation

This section introduces a motivating scenario of the task-constrained multi-agent stochastic planning problem, generalizes it into a generic task-constrained MDP model, and introduces a two-step method to solve it, which includes searching optimal policies for individuals, namely the *Task Routing*, and finding the best collection of non-overlapping task sets, which we refer to as the *Coordination*. Necessary definitions and basic concepts related to the subsequent analyses are also provided in this section.

2.3.1 Problem Definition

Stochastic planning problem for multi-agent systems with mutual exclusive task constraint arises in many life or engineering scenarios. One motivating application is the drone delivery problem where multiple agents (i.e., drones) $i \in Ag$ are sent to accomplish a cooperative mission constituted by a list of heterogeneous tasks (i.e., parcel deliveries) $j \in \mathcal{T}$, and a task assignment scheme $x_{ij} \in \{0, 1\}$, $\forall (i, j) \in Ag \times \mathcal{T}$ is expected to be found optimizing certain mission objectives.

The properties of drone delivery problems align with the three characteristics of the concerning task-constrained stochastic planning problems: 1) Drones are functionally independent except for the task assignment constraint. The delivery uncertainties may affect tasks in the same route sequence, but without any interference from other drones, drones are only confined by task constraints, and otherwise completely independent. 2) Decisions are made taking account of uncertainties. Drones can be severely affected by weather conditions, especially airflow, resulting in an unexpected delay for subsequent delivery tasks. Accommodating this kind of uncertainty in the planning tends to increase the quality of routes by improving customer satisfaction and stabilizing deliveries. 3) Coordination can be made in advance with prior knowledge of tasks. Deliveries always take place after the customer requests are received. The Network topology is assumed to be a fully spanned tree through which consistent decisions are made among the drone fleet prior to the delivery.

Once a problem satisfies the above three properties, it is recognized as a task-constrained multi-agent stochastic planning problem. A hybrid method of the MDP modelling technique and the auction algorithm is proposed to solve it. First, we give an introduction to the proposed generic task-constrained MDP modelling scheme.

2.3.2 Problem Formulation

A task-constrained MDP is defined by the tuple $\langle Ag, \mathcal{T}, \mathcal{K}, \langle \mathbf{M}_i \rangle_{i \in Ag} \rangle$.

- Ag is the set of agents.
- $\mathcal{T}$ is the set of tasks.
- $\mathcal{K} = \{k_i\}$ indicates the capacities of each agent.
- $\mathbf{M}_i$ is the MDP model of agent i which is confined by a global task assignment constraint. It is defined as the tuple $\langle \tau_i, s_i, a_i, p_i, r_i, x_i \rangle$:
 - τ_i indicates the decision epoch. The value of τ_i will add 1 after one decision is made to the agent i .
 - $s_i \in S_i$ is the factored local state for agent i (e.g., the system time, drone's state of charge, location, etc.).
 - $a_i \in A_i$ is the action for agent i , indicating the next task to execute under the current state s_i , which is selected from the available task set for agent i : $a_i = j, j \in \mathcal{T}_i$. $\mathcal{T}_i$ is defaulted to be $\mathcal{T}$.
 - $p_i = p(s'_i | s_i, a_i)$ represents the probability that the current state s_i is converted to the next state s'_i by acting a_i . (e.g., if the state only considers the system time, the next state is then determined by the probability distribution of task duration $t_i(a_i)$: $p(s'_i | s_i, a_i) \sim \Pr(t_i(a_i) | i, a_i)$.)
 - $r_i = r(s_i, a_i, s'_i)$ is the reward function which is defined by an immediate gain after executing the selected task a_i under state s_i .
 - $\mathbf{x}_i = \{x_{ij}, \forall j \in \mathcal{T}\}$ denotes the task accomplishment for agent i , where $x_{ij} : Ag \times \mathcal{T} \rightarrow \mathbb{R}_{\{0,1\}}$ has a similar form as the resource consumption function in [10] but is defined as binary indicating whether task j is accomplished by agent i .

The goal for solving this MDP is to obtain a joint policy π^* that maximizes the expectation of the joint accumulative reward for the agent group:

$$\pi^* = \operatorname{argmax}_{\pi} E \left[\sum_{i \in Ag} \sum_{\tau_i=0}^{\tau_i^f} r_i^{\tau_i} | \pi_i \right] = \operatorname{argmax}_{\pi} \sum_{i \in Ag} V_i(\pi_i, s_0) \quad (2.1)$$

s.t.

$$\begin{aligned} \sum_{i \in Ag} x_{ij} &\leq 1, & \forall j \in \mathcal{T} \\ \sum_{j \in \mathcal{T}} x_{ij} &\leq k_i, & \forall i \in Ag \end{aligned} \quad (2.2)$$

where π_i is the individual policy for agent i , τ_i^f indicates the maximum number of steps in the process, equal to the total task number that is executed by the agent i . The accumulated reward under the policy π_i could also be represented by the expected value $V_i(\pi_i, s_0)$ on the MDP model $\mathbf{M}_i$. The first inequation defines the inter-agent constraint in task allocation, i.e., duplicate selection of tasks is not allowed. The second inequation represents the intra-agent constraint which limits the upper number of tasks executed by agent i to capability k_i .

One interesting observation for this task-constrained MDPs is that if given a collection of non-conflict task sets $\{\mathcal{T}_i, \forall i \in Ag\}$, the global task constraint (i.e., the first inequation) is then satisfied and the individual MDP models $\langle \mathbf{M}_i \rangle_{i \in Ag}$ for each agent are completely independent and decoupled from others. Inspired by this insight, we exploit a decomposition approach by reformulating the constrained optimization problem into

$$\pi_i^*, N_{\mathcal{T}}^* = \operatorname{argmax}_{N_{\mathcal{T}}} \sum_{i \in Ag} \operatorname{argmax}_{\pi_i} \mathbb{E} \left[\sum_{\tau_i=0}^{\tau_i^f} r_i(\tau_i) \mid \pi_i, \mathcal{T}_i \right] \quad (2.3)$$

where $\mathcal{T}_i$ is the task set allocated to agent i . $\mathcal{N}_{\mathcal{T}} = \{\mathcal{T}_i, \forall i \in Ag\}$ is a collection of disjoint task sets that satisfies $|\mathcal{T}_i \cap \mathcal{T}_j| = 0, \forall i, j \in Ag$. The maximum step of the Markov process is $\tau_i^f = \min[k_i, |\mathcal{T}_i|]$.

To find both the optimal policies π_i^* and the task collection $\mathcal{N}_{\mathcal{T}}$, we present a two-step method: 1) Find the optimal policy π_i^* for each agent $i \in Ag$ by solving its individual MDP for any given task set $\mathcal{T}_i$; 2) Given policy π_i^* and the value function $V_i(\pi_i^*, \mathcal{T}_i)$, find the best collection of task sets $N_{\mathcal{T}}^* = \{\mathcal{T}_i, \forall i \in Ag\}$. At last, the policy π_i^* and the allocation $N_{\mathcal{T}}^*$ are merged into the global policy π^* .

Following the above idea, solving task-constrained MDPs is then decomposed into two subproblems: *Task Routing*: to obtain the local policy $\pi_i^* = \operatorname{argmax}_{\pi_i} V_i(\pi_i, s_0; \mathcal{T}_i)$, and *Coordination*: To find the task assignment result $\mathcal{N}_{\mathcal{T}}$ by solving the conventional task allocation problem, which is categorized as single-robot multi-task in-schedule assignment problems with task dependencies.

It should be noted that after decomposition, the optimization problem has been altered from the original one, which might be at the cost of the optimality compromise. Meanwhile, the decomposition benefits the problem solving, which reduces the computational complexity of the problem from NEXP-hard to P-complete.

2.3.3 Preliminary of Submodularity

This part gives some necessary preliminary concepts for the development and analysis of the proposed task allocation method.

Definition 1: (Submodularity) Given a finite set Ω , a set function $f : 2^\Omega \rightarrow \mathbb{R}$, where 2^Ω is the power set of Ω , is a submodular function if, for every $X, Y \subseteq \Omega$ with $X \subseteq Y$ and every $x \in \Omega \setminus Y$ we have

$$f(X \cup \{x\}) - f(X) \geq f(Y \cup \{x\}) - f(Y) \quad (2.4)$$

Equivalently, for every $S, T \subseteq \Omega$ we have

$$f(S) + f(T) \geq f(S \cap T) + f(S \cup T) \quad (2.5)$$

Definition 2: (Monotonicity) Given a set Ω , a set function $f : 2^\Omega \rightarrow \mathbb{R}$, that is defined on Ω , is monotone if, for every $X \subseteq Y \subseteq \Omega$,

$$f(X) \leq f(Y) \quad (2.6)$$

Definition 3: (Matroid) A matroid is a pair of Ω and $\mathcal{I}$, i.e., $\mathcal{M} = (\Omega, \mathcal{I})$, where Ω is a finite set and $\mathcal{I} \subseteq 2^\Omega$ is a collection of independent sets, satisfying:

- $\emptyset \in \mathcal{I}$
- for every $X \subseteq Y, Y \in \mathcal{I} \Rightarrow X \in \mathcal{I}$
- for every $X, Y \in \mathcal{I}, |X| < |Y| \Rightarrow \exists x \in Y \setminus X$ such that $X \cup \{x\} \in \mathcal{I}$.

Definition 4: (Partition Matroid) A partition matroid is a matroid $\mathcal{M} = (\Omega, \mathcal{I})$, where Ω is partitioned into disjoint sets $\Omega_1, \Omega_2, \dots, \Omega_l$ and

$$\mathcal{I} = \{X \in \Omega : |X \cap \Omega_i| \leq k_i \text{ for all } i = 1, \dots, l\} \quad (2.7)$$

for some given parameters $k_1, \dots, k_l$.

In the context of drone deliveries, we consider all task-agent pairs as a ground set, denoted as $\mathcal{N} : \mathcal{T} \times \mathcal{Ag}$, and make partitions each including task-agent pairs with the same task. The solution to drone delivery (conceptualized as a single-robot multi-task task allocation problem) is then to find a set of independent sets subject to the partition matroid, where $k_i = 1$. Therefore, the drone delivery problem modelled in this paper can be approached as a maximization optimization problem subject to a partition matroid constraint [56].

A monotone submodular function has good mathematical characteristics, with which difficulty can be approximated efficiently with a strong quality guarantee [151]. Specifically, a greedy algorithm that incrementally chooses elements by maximizing marginal utility provides the solution with at least $(1 - \frac{1}{e})$ optimality or $1/2$ optimality if there is a matroid constraint [151].

2.4 Auction-based Coordination Strategy

This section presents a decentralized auction-based task allocation strategy to address the *Coordination* phase of the above-described task-constrained MDPs. We first give a brief introduction to the basics of CBBA and its robust variant. Then, making use of the same framework, a new auction-based coordination strategy is suggested for the task-constrained MDPs with a reformulated score function.

2.4.1 CBBA and its Robust Extension

The CBBA, originally proposed by [94], is a widely-known decentralized task allocation method, which resolves conflict-free assignments by implementing the auction algorithm on networked agents and augmenting it with a consensus protocol. The algorithm consists of two phases: 1) the bundle construction phase, where a bundle of tasks is greedily chosen by each UAV, and 2) the task consensus phase, where the conflicts of tasks are resolved through communicating and negotiating with neighbours. The algorithm iterates between these two phases until all tasks are settled in a consistent manner.

Further, the discrepancies between the plan and stochastic real-world operating environments motivated robust extensions to CBBA, where the expected reward is instead maximized [96]. For a formal elaboration, let θ denotes all uncertainty pa-

rameters related to the score calculation, the robust version of the objective function is adapted as

$$\max J_i = E_\theta \left\{ \sum_{j=1}^{|\mathcal{T}|} c_{ij}(\mathbf{x}_i, \theta) \right\} \quad (2.8)$$

where c_{ij} is the marginal score of task j , given tasks that are already allocated to agent i , shown as $\mathbf{x}_i$.

It should be noted that except for the effect on the reward score, uncertainties also influence the optimal task order due to the coupling among tasks. All possible task orders must be taken into account when calculating the score of a task bundle, which should be

$$J_i = \max_{\mathbf{p}_i} J_{\mathbf{p}_i} = \max_{\mathbf{p}_i} E_\theta \left\{ \sum_{j=1}^{|\mathcal{T}|} c_{ij}(\mathbf{p}_i, \theta) x_{ij} \right\} = \max_{\mathbf{p}_i} \int_{\theta \in \Theta} \left\{ \sum_{j=1}^{|\mathcal{T}|} c_{ij}(\mathbf{p}_i, \theta) x_{ij} \right\} P(\theta) d\theta \quad (2.9)$$

where $\mathbf{p}_i$ indicates the task sequence, following which the task will be executed one by one.

By means of extracting uncertainty parameters and optimizing the priori-estimated reward score, this extension indeed enhances the robustness of the CBBA algorithm in the uncertain environment. However, analytically computing this robust score is very difficult due to the integral operation of uncertainty parameters and numerous permutations of the task order. For this reason, some researchers adopt a sampling process to approximate these calculations with the assumption that the order of existing tasks is fixed in order to maintain computational tractability [96]. However, the sampling operation remains cumbersome, and meanwhile, it might also violate the DMG condition of the algorithm convergence. Therefore, a novel task coordination method is proposed in the following section providing a more efficient way to acquire robust scores in the context of the MDP modelling.

2.4.2 The Proposed Coordination Strategy

While adopting the identical consensus stage as CBBA, the proposed task coordination method follows a novel formulation of score functions based on the MDP modelling of the problem. Suppose we now have an intended task set for one agent at some point during the coordination process, its score function is obtained by calculating the MDP's state-value function with respect to the assigned task set $\mathcal{T}$

and state s_τ at time step τ :

$$V_\pi(s_\tau; \mathcal{T}) = E \left[\sum_{t=\tau}^{|\mathcal{T}|} r(s_t, \pi(s_t), s_{t+1}; \mathcal{T}) \right] \quad (2.10)$$

The maximum objective function is then converted to

$$\max J_i(\mathcal{T}_i) \Rightarrow V_\pi(s_\tau; \mathcal{T}_i) \quad (2.11)$$

where $V_\pi(s_\tau; \mathcal{T}_i)$ means the expected accumulative reward of the agent under the state s_τ with the allocated task set $\mathcal{T}_i$. Using the new utility function formulation, the maximum objective is then acquired by solving the value function of MDPs. Given the agent's state s_i , the marginal gain of task j is then calculated by the difference between two value functions:

$$c_{ij} = J^*(\mathcal{T}_i \cup \{j\}) - J^*(\mathcal{T}_i) = V_\pi^*(s_\tau; \mathcal{T}_i) - V_\pi^*(s_\tau; \mathcal{T}_i \cup \{j\}) \quad (2.12)$$

The task bundle for each agent is constructed by greedily selecting the task with the largest marginal gain that is defined by the value function (2.11). The pseudocode of the bundle construction procedure is presented as Algorithm 2.1. Each agent carries three vectors: a winning agent list $\mathbf{z}_i \in \mathcal{I}^{|Ag|}$, a winning bid list $\mathbf{y}_i \in \mathbb{R}_+^{|Ag|}$, and a bundle $\mathbf{b}_i \in (\mathcal{T} \cup \{\emptyset\})^{|k|}$ where k is the bundle depth. These three vectors get updated in the bundle construction procedure. The value for every state in the MDP model is calculated by a policy iteration approach before allocation starts.

Algorithm 2.1 Bundle Construction for agent i

Procedure Update Bundle ($\mathbf{z}_i, \mathbf{y}_i, \mathbf{b}_i$)

```

1: while  $|\mathbf{b}_i| < k$  do
2:   for  $j \in \mathcal{T} \setminus \mathbf{b}_i$  do
3:      $c_{ij} = V_\pi^*(s_i; \mathbf{b}_i \cup \{j\}) - V_\pi^*(s_i; \mathbf{b}_i)$ 
4:      $h_{ij} = \mathbb{I}(c_{ij} > y_{ij})$ 
5:   end for
6:    $j^* \leftarrow \arg \max_{j \notin \mathbf{b}_i} c_{ij} \cdot h_{ij}$ 
7:    $\mathbf{b}_i \leftarrow \mathbf{b}_i \cup \{j^*\}$ 
8:    $y_{ij^*} \leftarrow c_{ij^*}$ 
9:    $z_{ij^*} \leftarrow i$ 
10: end while

```

Another advantage of this MDP-formulated method is that all possible orders of tasks in the bundle will be spontaneously considered when calculating the value

function. Most existing auction-based allocation algorithms operate with an implicit assumption that the order of selected tasks keeps fixed when inserting new tasks into the sequence, for ease of computing. This assumption is regardless of the impact of new tasks on the optimal execution order and thus might violate the underlying conditions for optimality guarantees. The reformulated score function herein is directly assessed on the task list and is not affected by the task construction sequences. In other words, it has no relationship with the order of task sequence and can be determined as long as the task list is given for a vehicle. Therefore, this formulation helps to further improve the solution optimality and truly holds the condition for the following optimality analysis.

2.4.3 Convergence Analysis with Submodular Rewards

Using a similar consensus topology, the sufficient condition of convergence for the proposed method stays the same as CBBA, i.e., the score function must satisfy the submodularity [94]. The submodularity proof of value function is given by *Theorem1*. Prior to the proof, we first define a preliminary condition, termed *Independent Extensibility*.

Definition 4: (Independent Extensibility) Let Ω be a finite set. θ defines all transition parameters in a stochastic process $\{s_t\}$ on the set Ω . The stochastic process $\{s_t\}$ is independently extensible, if, for every $X \subseteq Y \subseteq \Omega$, it satisfies

- $\theta(X) \subseteq \theta(Y)$
- $P[\theta(Y)] = P[\theta(X)] * P[\theta(Y) \setminus \theta(X)]$

where $P[\theta]$ is the cumulative distributed function of the variable set θ .

The property of independent extensibility can be found in many applications. Using VRPs to exemplify, driving between two pairs of locations is regarded as independent stochastic processes. In this case, the stochastic process regarding city x and city y follow the time distribution on its arc $t(c_{xy})$. When the set is extended to have city z , we got two more variables on two extra edges, $t(c_{xz})$ and $t(c_{yz})$, which, however, are independent from $t(c_{xy})$. Thus we have $\theta([x, y]) \subseteq \theta([x, y, z])$ and $P(\theta([x, y, z])) = P[t(c_{xy})] * P[t(c_{xz}), t(c_{yz})]$, the stochastic process concerned by VRPs satisfies independent extensibility.

Theorem1: Presuming an independently extensible stochastic process $\{s_t\}$, if its reward function R is submodular, its state-value function $V_\pi(s_\tau)$ is also submodular.

Preliminary: First, let us parameterize the state transition of MDPs on the premise of task set $\mathcal{T}$ as $\boldsymbol{\theta}_{\mathcal{T}}$. The value function is then presented as

$$V_{\pi}^*(s_{\tau}; \mathcal{T}) = E_{\boldsymbol{\theta}_{\mathcal{T}}} [R_{\pi^*}(s_{\tau}; \mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})] \quad (2.13)$$

where $R_{\pi^*}(s_{\tau}, \mathcal{T}; \boldsymbol{\theta}_{\mathcal{T}}) = \sum_{t=\tau}^{\tau+|\mathcal{T}|} r(s_{\tau}, \pi^*(s_t), s_{t+1}; \mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})$ is the cumulative reward for executing task set $\mathcal{T}$ under the transition parameters $\boldsymbol{\theta}_{\mathcal{T}}$. It is noted that given the optimal policy π^* and state s_{τ} , $R_{\pi^*}(s_{\tau}, \mathcal{T}; \boldsymbol{\theta}_{\mathcal{T}})$ is also equivalent to the deterministic reward function with respect to the task set $\mathcal{T}$ on the condition of the parameter instance $\boldsymbol{\theta}_{\mathcal{T}}$. Further, assume that the policy π^* and state s_{τ} stay unchanged during the task coordination, we can get a simplified notation of the value function:

$$V(\mathcal{T}) = E_{\boldsymbol{\theta}_{\mathcal{T}}} [R(\mathcal{T}; \boldsymbol{\theta}_{\mathcal{T}})] \Leftarrow V_{\pi}^*(s_{\tau}; \mathcal{T}) \quad (2.14)$$

As we assume that the process is independently extensible, according to *Definition 4*, the distribution of parameter $\boldsymbol{\theta}_{\mathcal{T}}$ is not affected by any additional parameters introduced by extra tasks. Considering a superset $\mathcal{T}'$, $\mathcal{T} \subseteq \mathcal{T}'$, the expectancy of $R(\mathcal{T})$ on $\boldsymbol{\theta}_{\mathcal{T}}$ equals to its expectancy on $\boldsymbol{\theta}_{\mathcal{T}'}$ as the additional parameter $\boldsymbol{\theta}_{\mathcal{T}' \setminus \mathcal{T}}$ is independent of $\boldsymbol{\theta}_{\mathcal{T}}$, which is formally presented as

$$E_{\boldsymbol{\theta}_{\mathcal{T}}} [R(\mathcal{T}; \boldsymbol{\theta}_{\mathcal{T}})] = E_{\boldsymbol{\theta}_{\mathcal{T}' \setminus \mathcal{T}}} E_{\boldsymbol{\theta}_{\mathcal{T}}} [R(\mathcal{T}; \boldsymbol{\theta}_{\mathcal{T}'})] = E_{\boldsymbol{\theta}_{\mathcal{T}'}} [R(\mathcal{T}; \boldsymbol{\theta}_{\mathcal{T}'}), \quad \forall \mathcal{T}' \supseteq \mathcal{T} \quad (2.15)$$

Via this equation, we further obtain a connection between the marginal value function $V(\mathcal{T}') - V(\mathcal{T})$ and the marginal reward function $R(\mathcal{T}') - R(\mathcal{T})$:

$$\begin{aligned} V(\mathcal{T}') - V(\mathcal{T}) &= E_{\boldsymbol{\theta}_{\mathcal{T}'}} [R(\mathcal{T}'; \boldsymbol{\theta}_{\mathcal{T}'})] - E_{\boldsymbol{\theta}_{\mathcal{T}}} [R(\mathcal{T}; \boldsymbol{\theta}_{\mathcal{T}})] \\ &= E_{\boldsymbol{\theta}_{\mathcal{T}'}} [R(\mathcal{T}'; \boldsymbol{\theta}_{\mathcal{T}'})] - E_{\boldsymbol{\theta}_{\mathcal{T}'}} [R(\mathcal{T}; \boldsymbol{\theta}_{\mathcal{T}'})] \\ &= E_{\boldsymbol{\theta}_{\mathcal{T}'}} [R(\mathcal{T}'; \boldsymbol{\theta}_{\mathcal{T}'} - R(\mathcal{T}; \boldsymbol{\theta}_{\mathcal{T}'}))] \end{aligned} \quad (2.16)$$

It implies that just like the state-value function defines the the expectation of cumulative rewards on uncertainties, the marginal value function is also the expectation of the marginal reward function on the joint uncertainty parameters $\boldsymbol{\theta}_{\mathcal{T}'}$.

Proof: Now let us consider two available task sets $\mathcal{T}_m$ and $\mathcal{T}_n$, $\mathcal{T}_m \subseteq \mathcal{T}_n$. Given the optimal policy π^* and the state s , the marginal gain of task j with the prior of the task set $\mathcal{T}_m$, according to the expanded expectation formulation of the value

function (2.16), is presented as follows:

$$\begin{aligned} V(\mathcal{T}_m \cup j) - V(\mathcal{T}_m) &= E_{\boldsymbol{\theta}_{\mathcal{T}_m \cup j}} [R(\mathcal{T}_m \cup j; \boldsymbol{\theta}_{\mathcal{T}_m \cup j}) - R(\mathcal{T}_m; \boldsymbol{\theta}_{\mathcal{T}_m \cup j})] \\ &= E_{\boldsymbol{\theta}_{\mathcal{T}_n \cup j}} [R(\mathcal{T}_m \cup j; \boldsymbol{\theta}_{\mathcal{T}_n \cup j}) - R(\mathcal{T}_m; \boldsymbol{\theta}_{\mathcal{T}_n \cup j})] \end{aligned} \quad (2.17)$$

Similarly, the marginal gain with respect to the task set $\mathcal{T}_n$ is presented as:

$$V(\mathcal{T}_n \cup j) - V(\mathcal{T}_n) = E_{\boldsymbol{\theta}_{\mathcal{T}_n \cup j}} [R(\mathcal{T}_n \cup j; \boldsymbol{\theta}_{\mathcal{T}_n \cup j}) - R(\mathcal{T}_n; \boldsymbol{\theta}_{\mathcal{T}_n \cup j})] \quad (2.18)$$

As R is assumed to be submodular, i.e., $R(\mathcal{T}_n \cup j) - R(\mathcal{T}_n) \leq R(\mathcal{T}_m \cup j) - R(\mathcal{T}_m)$, its expectation on the same parameter distributions is going to be submodular, thus:

$$V(\mathcal{T}_n \cup j) - V(\mathcal{T}_n) \leq V(\mathcal{T}_m \cup j) - V(\mathcal{T}_m) \quad (2.19)$$

■

Similarly, we deduced the *Theorem2* for the monotonicity of state-value functions.

Theorem2: Presuming an independently extensible stochastic process $\{s_t\}$, if its reward function R is monotone, its state-value function $V_\pi(s_\tau)$ is also monotone.

2.4.4 Optimality Analysis

According to [152], for problems that maximize a monotone submodular function subject to a matroid constraint, the greedy algorithm is guaranteed to produce a solution that is bigger than 50% of the optimal solution.

Theorem1 has indicated the submodularity of the objective function with the assumption of the submodular deterministic reward function and the independent extensibility. The score function is also shown to be monotone if the reward function is monotone and the stochastic process is independently extensible, according to *Theorem2*. The non-overlapping task constraint could be considered as a partition matroid where all task-agent pairs are defined as a ground set $N : \mathcal{T} \times Ag$ and the collection of subsets, which is limited to at most one element from the same subset. Overall, the concerned task coordination problem satisfies the conditions listed in [152] for the optimality guarantee with the greedy algorithm. The CBBA holds the identical solution with the greedy algorithm when the reward function is submodular and monotone. Accordingly, the proposed task coordination strategy retains the guarantee of 50% optimality. Note that the optimality guarantee refers

to the optimal expected reward rather than actual execution rewards as uncertainty parameters will not be known until tasks are executed.

In summary, a new utility function is designed in the proposed coordination algorithm aiming to optimise the expected accumulative rewards. The algorithm is expected to enhance the robustness under uncertain task execution while maintaining the convergence, as well as the prior optimal guarantee of 50%.

2.5 Approximation via Neural Networks

While the proposed coordination strategy provides an efficient solution for task-constrained planning, solving MDPs is still subject to the curse of dimensionality, i.e., the state-action space grows exponentially with the problem scale. In this section, we consider an approximation method via neural networks to generate the planning policy and value function for large-scale task-constrained MDPs.

2.5.1 Deep Auction

Deep Auction, an approximated variant of the proposed task coordination method, employs two neural networks to approximate the individual policies and value functions respectively. It is inspired by the well-known actor-critic architecture where a policy network maps the action probability and a critic baseline is used to calculate the advantage value for the benefit to reduce variances.

To elaborate on how the Deep Auction method works, Figure 2.1 and Figure 2.2 summarize its working mechanism in a sequence of a training phase and an implementation phase. The framework of the approximator training, shown in Figure 2.1, inherits the learning paradigm of REINFORCE algorithm with baselines, where two neural networks are updated using the *Advantage* error defined by the difference between the accumulative reward from the environment and the value evaluated by the critic. After training, the *Actor* and the *Critic*, both modelled by a *Transformer* network, are able to map the policy π^* and value function V^* given a task set $\mathcal{T}_i$ and the agent status s_i observed from the environment.

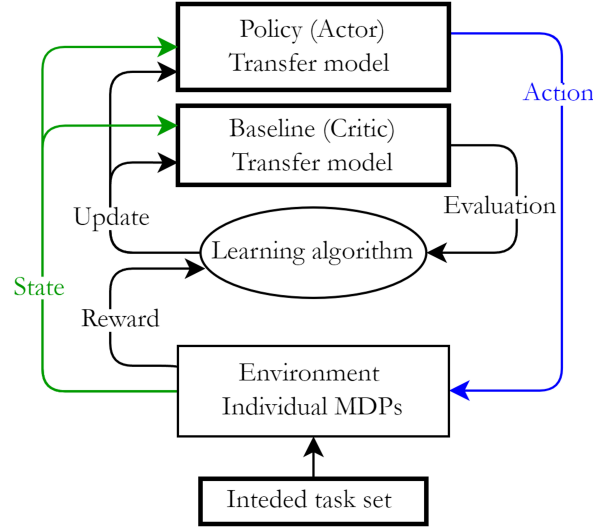


Figure 2.1: The training phase of the Deep Auction method

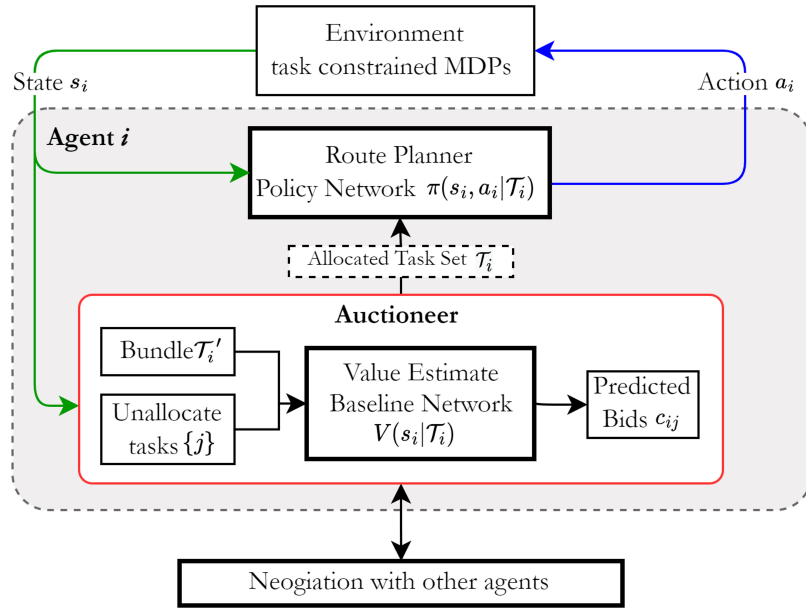


Figure 2.2: The implementation phase of the Deep Auction method

In terms of the implementation, seen in Figure 2.2, the trained critic baseline constitutes an auctioneer that negotiates with other agents using the coordination strategy proposed in Section 2.4. Coordination bids c_{ij} are calculated by the marginal score obtained using the value estimate network. By the end of the auctioning process, a consensus on the task assignment is acknowledged by all agents with a task set $\mathcal{T}_i$ allocated to the current agent i and then passed to a route planner. The route planner greedily chooses the next action a_i from the intended task set according to

the action probability $\pi(s_i, a_i | \mathcal{T}_i)$ mapped by the policy network till all tasks have been accomplished or another task coordination is invoked.

In the proposed deep auctioned coordination method, the two decomposed problems for solving task-constrained MDPs, i.e., the *Coordination* and the *Task Routing*, as stated in Section 2.3.2, are addressed by introducing the policy network and the critic baseline in an approximate scalable way.

2.5.2 Network Architecture

The *Transformer* model [153], recognized as a powerful work to address the combinatorial problem with unordered inputs, is adopted for both the policy network and critic network to map the action probability and value function respectively.

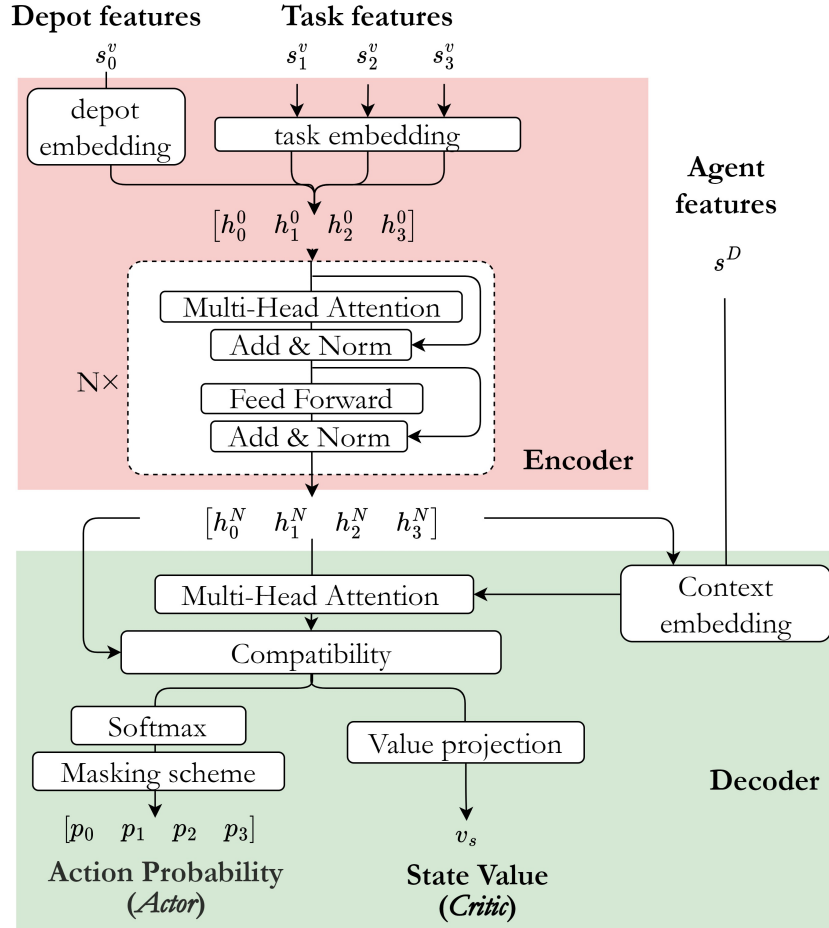


Figure 2.3: The neural network architecture. The critic network and actor network share the same architecture except for the output layer for the corresponding output format.

Figure 2.3 depicts the architecture of networks, which takes advantage of the encoder-decoder architecture in the Transformer model. The network architecture follows the *Transformer* model presented in [154] and varies slightly in the output layer between the actor network and the critic network. For the actor i.e., the policy network, the output is the probability for all unmasked actions by passing the *compatibility* vector through a Softmax layer and a Masking scheme. For critic i.e., the baseline, the output is the state value corresponding to the current mission state and agent state, which is decoded by a $N \times 1$ linear layer, where N is the number of tasks.

2.5.3 Training Algorithm

Networks get trained by the REINFORCE algorithm integrated with the critic baseline. In detail, at the beginning of each episode, a batch of initial states $\{s_0\}_{b \in B}$ and corresponding task sets $\{\mathcal{T}\}_{b \in B}$ are stochastically sampled. Tasks are then sequentially selected from the set according to the probability output of the policy network $\pi(s; \theta)$ and rewards $r(s, a)$ are meanwhile collected. By the end of each episode, we calculated a policy loss $\mathcal{L}$ to train the policy model, which is defined as the *Advantage* error averaged over the batch B :

$$\mathcal{L}(B; \theta) = \frac{1}{|B|} \sum_{b=1}^{|B|} A(s^b; \pi) = \frac{1}{|B|} \sum_{b=1}^{|B|} (R(s^b; \pi) - bl(s^b)) \quad (2.20)$$

The *Advantage* error is used for variance reduction and is defined by the difference between the accumulative reward and a baseline $bl(s)$. The baseline is calculated by the critic network, which estimates the state-value function, denoted as $\hat{v}(s; \mathbf{w})$ parameterized by $\mathbf{w}$:

$$\mathcal{L}(B; \theta) = \frac{1}{|B|} \sum_{b=1}^{|B|} (R(s^b; \pi) - \hat{v}(s^b; \mathbf{w})) \quad (2.21)$$

The function $\hat{v}$ can be learned from the observation $R(s, \pi)$, which is the Monte Carlo estimate of the true value function. Thus, we define a value loss based on the mean square error between the estimate function and observations concerning parameters $\mathbf{w}$:

$$\mathcal{L}_v(B; \mathbf{w}) = \frac{1}{|B|} \sum_{b=1}^{|B|} (R(s^b) - \hat{v}(s^b; \mathbf{w}))^2 \quad (2.22)$$

Standard back-propagation is then adopted to update the policy network and the critic network using the gradient descent method. Beneficial from more efficient training, parameters are shared among these two networks except those of the output projection layer. However, in this case, the learning for the critic network might get distracted by policy errors, as the shared parameters are impacted by the variance of the policy loss due to the greedy sampling method. This does not matter in previous studies, as the critic network is only used to facilitate the training of the actor and will be cut off in the implementation. However, in our study, we are looking for good performance of both two networks. To that end, the training is divided into two phases:

1) Generic Training: All parameters $(\boldsymbol{\theta}, \boldsymbol{w})$ in both networks get trained. Joint parameters are updated using combined gradients:

$$\begin{aligned} d\boldsymbol{\theta} &= \frac{\partial \mathcal{L}(B; \boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = \frac{1}{|B|} \sum_{b=1}^{|B|} (R(s^b) - \hat{v}(s^b)) \nabla_{\boldsymbol{\theta}} \log \pi_{\boldsymbol{\theta}}(s^b) \\ d\boldsymbol{w} &= \frac{\partial \mathcal{L}_v(B; \boldsymbol{w})}{\partial \boldsymbol{w}} = \frac{1}{|B|} \sum_{b=1}^{|B|} \frac{\partial (R(s^b) - \hat{v}(s^b; \boldsymbol{w}))^2}{\partial \boldsymbol{w}} \end{aligned} \quad (2.23)$$

2) One-hot Critic Training: In this phase, the value network is detached from the policy network and updated for a more delicate adjustment on $\boldsymbol{w}$, with parameters in the policy network fixed.

2.5.4 Convergence Guarantee

As mentioned in Section 2.4.3, the sufficient convergence condition for the decentralized auction-based coordination framework is that the *bids* must satisfy submodularity. However, the network-approximated score function might violate the property of submodularity due to the inevitable approximating error using neural networks, even though its MDP form has been proven submodular by *Theorem1*. To resolve this, we implement a *bid wrapping function* proposed by [97] that disguises the actual task scores with submodular *bids* when sharing between agents.

The basic idea behind this wrapping function stems from the key insight that the *bids* agents share must satisfy submodularity rather than its actual task scores. Thus, when calculating the marginal scores for all available tasks, $c_{ij} = V(\mathbf{b}_i \cup \{j\}) -$

$V(\mathbf{b}_i)$, $\forall j \in \mathcal{T} \setminus \mathbf{b}_i$, each of them is wrapped with the function:

$$c'_{ij} = \min(c_{ij}, y_{ik}), \quad \forall k \in \mathbf{b}_i \quad (2.24)$$

where c_{ij} is the actual marginal gain, c'_{ij} is the wrapped coordination gain, and y_{ik} is the existing bid for all tasks in its bundle. The bid wrapping function ensures that the new bids are not greater than the bids already in the bundle, and thus will not affect their allocations. For further details and proofs of convergence, please refer to [97].

2.6 Comparison with Base Methods

The proposed work builds on multiple base research, featuring with advantages on the trade-off question between computational expenses and robustness against uncertainties. In addition to Section 2.4 where we have demonstrated the improved performance of the mission planning algorithm targeting the listed objectives, in this section we will mainly focus on its optimality, and time and space complexity, in comparison with the CBBA and the centralized Multi-agent Reinforcement Learning algorithm.

First, the proposed task coordination method features a reduction of computing amount in the bundle construction phase compared to CBBA, which benefits from the new-formulated non-ordered score function. In the original CBBA, tasks are sequentially inserted into the path with the best location gained by a trial loop of all possible positions [94]:

$$c_{ij}[b_i] = \max_{n \leq |p_i|} S_i^{p_i \oplus n\{j\}} - S_i^{p_i}, \quad \forall j \notin b_i \quad (2.25)$$

where the trial number $|p_i|$ depends on the length of the path, increasing linearly with problem dimensions. In contrast, in the proposed method, bids are calculated using a value function defined on the non-ordered task set, thus exempted from the need to find the best insertion positions. Formally, given the total number of tasks n , the depth of the bundle k , and the number of agents a , we summarize the total amount of score calculation for using the proposed auction method, CBBA, robust CBBA in Table 2.1.

Table 2.1: Complexity comparison

Method	Individual complexity	Total complexity	Individual complexity ($k = n$)	Total complexity ($k = n$)
Proposed method	$O(nk)$	$O(ank)$	$O(n^2)$	$O(an^2)$
CBBA	$O(nk^2)$	$O(ank^2)$	$O(n^3)$	$O(an^3)$
Robust CBBA	$O(rnk^2)$	$O(arnk^2)$	$O(rn^3)$	$O(arn^3)$

where r is the sampling number of robust CBBA.

An improved optimality guarantee is also facilitated by the proposed non-ordered score function. As we previously noted, there is an unnoticeable assumption in the CBBA process and most of its variations for computational tractability, that the path is always fixed when additional tasks are added to the sequence. However, for some mission profiles, especially those with uncertain executions, the optimal order of actions may alter. Ignoring the influence of additional tasks on optimum execution order reduces solution optimality, resulting in a 50% guarantee relative to a weakened optimal value. Instead of defining the score on a path, the suggested method calculates the marginal gain directly on the task bundle, taking into account all permutations of the task bundle, truly holding 50% optimality for the problem.

The decomposition of Multi-agent MDPs reduces the complexity of solving M-MDPs by using a predefined task bundle depth k . In the proposed work, M-MDPs is fractioned by individuals and utilized to compute bids for auction-based coordinates. Due to that, the constructed MDPs only attempt to solve the problem of dimensions up to the depth of the bundle k instead of the total problem size n . For example, if there are $a = 10$ agents serving $n = 50$ customers, for a relatively even allocation the task number for each agent is restricted to $k = 10$. In this case, the dimension of intended problems is $1 \sim 10$, instead of 50. This characteristic is imperative for the implementation of large-scale applications. In the integrated M-MDPs, the dimensionalities of the joint action and state space increase exponentially as the problem size increases, though its worst-case complexity is P-complete [117]. It is reported in [155] that one training episode takes 3 mins for solving VRPs of 10 customers while 22 mins/epoch is required for 50 customers. In contrast, the proposed decomposition strategy will keep the training time constant up to 3 mins/epoch regardless of the overall issue size.

2.7 Case Study on Drone Deliveries

To test the performance and expected properties of the proposed auction-based coordination method, we implement it in the context of the drone delivery problems, which is cast into stochastic price-collecting VRPs with time windows operated on the unmanned aerial platform. In this section, we start by introducing the definition of this problem and how we generate its mission instances. Then we discuss the performances of using the proposed auction-based coordination method in comparison to state-of-the-art approaches. Demonstrations are conducted in terms of the exact method for small-scale problems and the approximate extension to larger problems.

2.7.1 Drone Delivery Problem with Time Windows

Suppose a drone delivery scenario where there is a list of customer requests $j \in \mathcal{T}$ and a fleet of drones $i \in \mathcal{Ag}$ is sent from the depot (x^0, y^0) to serve the customers and finally returns to the depot after visiting all the tasks. The corresponding price r_j is collected from the customer j if any of the drones arrive at its location (x^j, y^j) during the time window $[t_r^j, t_d^j]$, otherwise a pending cost c_{pen} is imposed. In this logistic problem, optimizing the delivery flow indicates maximizing the rewards collected from customer requests while minimizing the loss of failed tasks.

According to the described delivery scenario, we independently generate sets of instances for different problem dimensions using the mission parameters shown in Table 2.2 and Table 2.3 for performance evaluation. Some of the mission settings are adapted from [155].

Due to the impact of airflows, the time cost on arcades between customers is shown to be stochastic considering the wind-sensitive property of drones. We simulate the stochastic travel times by sampling the drones' speed from a normal distribution, shown in Table 2.4.

2.7.2 Simulations with Small-Scale Problems

In the first simulation, we test the coordination algorithm proposed in Section 2.4 on a set of small-size drone delivery problems (task number is $2 \sim 5$). The simulation result is compared to the original CBBA and the sampling robust extension of CBBA [96]. A Sample process of $N = 1000$ is used in the sampling robust CBBA

Table 2.2: Generic parameter values or distributions used to generate instances of drone delivery. N_{small}, M_{small} are sets of customer counts and drone counts for small-scale simulations. N_{large}, M_{large} are sets of customer counts and drone counts for large-scale simulations. $\mathcal{U}(a, b)$ denotes a discrete uniform distribution on integer interval $[a, b]$.

Parameter notation	Value or distribution
Planning horizon (min)	$H = 480$
Number of customers	$N_{small} = [2, 3, 4, 5]$ $N_{large} = [10, 20, 50]$
Number of drones	$M_{small} = 2$ $M_{large} = \lceil \frac{N}{5} \rceil$
Expected drone flight speed (km/min)	$V = 1$
Depot location (km)	$x^0, y^0 \sim \mathcal{U}(0, 100)$
Customer locations (km)	$x^j, y^j \sim \mathcal{U}(0, 100)$
Customer price	$r^j = 1$
Penalty for failed customer	$c_{pen} = 1$

to approximate the expected score function. A submodular wrapper is also adopted in the robust CBBA method for convergence.

Suppose that there is a list of $2 \sim 5$ customer requests and a group of 2 drones is sent from one common depot to these requests for the delivery mission. For the evaluation purpose, we randomly generate 100 mission instances of each problem dimension with parameters following Table 2.2 and Table 2.3 and then solve it using the above three coordination methods. The assignment results are validated with the environment that subjects to stochastic travel times following the distribution shown in Table 2.4. Validation is conducted for 100 rounds. The expected cumulative rewards and validation rewards of all three methods are presented in Table 2.5. Computing times are shown in Table 2.6.

Table 2.5 demonstrates better robustness of the proposed Auction-based method and the robust sampling CBBA against stochastic edge costs. As indicated in the table, the original CBBA has the highest expectation for planning results, even 100% accomplish rates for instances of $N = 3, 4, 5$, but suffers a performance degradation (to about 97% success rate) when executed in stochastic environments. In contrast, the proposed auction method and sampling CBBA coordinate which use more precise scores though in two distinct ways, both result in a greater completion rate (about

Table 2.3: Parameter distributions related to time windows. $\mathcal{U}(a, b)$ denotes a discrete uniform distribution on integer interval $[a, b]$; $\mathcal{U}\{x_1, \dots, x_n\}$ denotes a discrete uniform distribution on finite set $\{x_1, \dots, x_n\}$; and $\mathcal{B}(p)$ is a Bernoulli distribution of parameter p .

Parameter notation	Value or distribution
Is customer constrained by TM?	$\delta_{TW}^j = \mathcal{B}(p_{TW})$ where $p_{TW} \sim \mathcal{U}(0.25, 0.5, 0.75, 1.0)$
Customer ready time (min)	$t_r^j \sim \mathcal{U}(0, t_{\max}^j)$ where $t_{\max}^j = H - \frac{\text{dist}(0, j)}{V} - \tau^j$
Customer time window width (min)	$w_{TW}^j \sim \mathcal{U}(30, 90)$
Customer service duration (min)	$\tau^j \sim \mathcal{U}(10, 30)$
Customer due time (min)	$t_d^j = t_r^j + w_{TW}^j$

Table 2.4: The distribution of drone speed

Parameter notation	Distribution
Drone flight speed (km/min)	$V \sim \mathcal{N}(\mu_V, \sigma_V^2)$ where $\mu_V = 1, \sigma_V^2 = [0, 0.05, 0.1, 0.2]$

1% increase) and a reduced deviation from predicted performance.

While the sampling CBBA performs comparatively to the proposed method, it costs much longer time due to the sampling process, as shown in Table 2.6. Further, as seen in Fig. 2.4, the running times for CBBA and robust CBBA present a trend of exponential growth as the customer count increases, while the proposed auction method increases linearly with respect to the problem size, implying better efficiency on large-scale problems.

2.7.3 Deep Auction Training Result

To train the two neural networks presented in Deep Auction, we built a simulated environment by which we can evolve the mission state and vehicle state, and get rewards back to update the policy network and the critic network. Instances are generated for training and testing following the parameter scheme in Table 2.2 and Table 2.3.

Training over varied dimensions: It is noted that for a generalization ability

Table 2.5: Total task scores: Expected vs. Actual

Mission setting \ Method		Proposed auction method		CBBA		Robust CBBA	
Dro. count	Cust. count	Exp. re-ward	Act. reward (finish rate)	Exp. re-ward	Act. reward (finish rate)	Exp. re-ward	Act. reward (finish rate)
2	2	1.90	1.90(97.5%)	1.8	1.86(96.6%)	1.86	1.86(96.65%)
2	3	2.84	2.84(97.38%)	3.0	2.77(96.2%)	2.88	2.81(96.93%)
2	4	3.93	4.0 (100.0%)	4.0	3.82(97.08%)	4.0	4.0 (100.0%)
2	5	4.93	4.92(99.28%)	5.0	4.84(98.44%)	4.97	4.94(99.47%)

Table 2.6: Running time of three methods for solving 1000 validation instances of each problem size

Method \ Cust. Count	2	3	4	5
Proposed Auction	1"	2"	4"	5"
CBBA	4"	8"	17"	25"
Sampling CBBA	3'35"	8'57"	18'50"	29'33"

over problem dimensions in value estimate and route planning, datasets for training and testing are constituted by instances of varied dimensions $N \sim \mathcal{U}(1, k)$, where $k = 10$ is the bundle depth defined by the proposed auction-based coordination method.

Network setup: For the structure of *Transformer* networks, the depot embedding and task embedding are both one-layer element-wise linear projections with dimension 128. The multi-head attention network consists of 8 heads computing key vectors and value vectors of dimension $d_k = d_v = 16$. There are $N = 3$ multi-attention modules in a row for the encoder network. In the feed-forward layer, the node features are passed through node-wise projections of one hidden sublayer with the ReLu activation and 512 hidden neurons. The Adam Optimizer is used to train networks with a constant learning rate $\alpha_\theta = 10^{-4}$ for the policy network and a constant learning rate $\alpha_w = 10^{-5}$ for the critic network.

We run training for 100 epochs for the generic policy training and an additional 50 epochs for one-hot value training. In one epoch, we process 1.28 million instances in

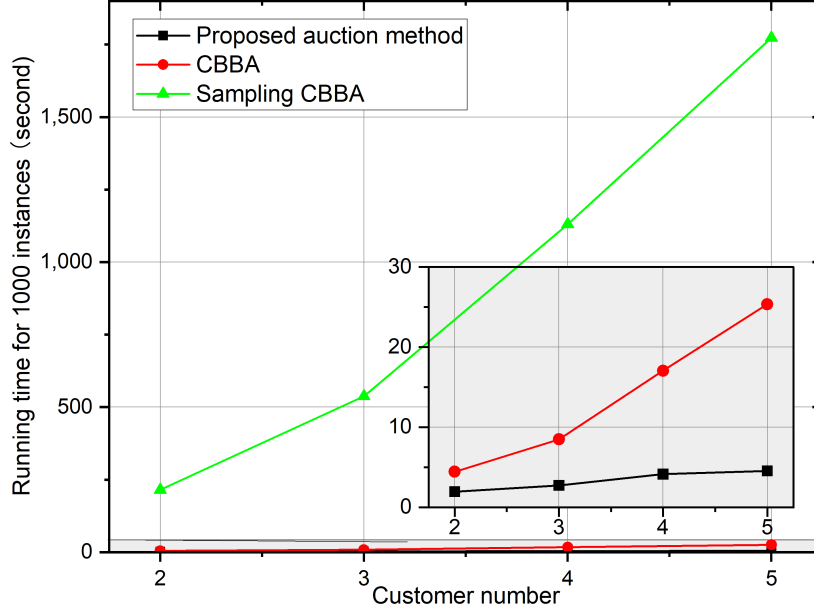


Figure 2.4: Running time for all three methods as the customer number changes.

2,500 iterations with a batch size of 512. Each epoch takes up to 1 minute using the Tesla A100 GPU. Learning curves are presented in Figure 2.5. All curves integrate data from two learning phases: generic learning for the first 100 epochs and value learning for the 100-150 epochs.

One interesting point is observed from the Figure 2.5. The cumulative reward obtained from the test dataset gives a better performance than those from the training dataset before the 100th epoch. This phenomenon is reasonable, as the agent stochastically selects its action according to the policy probability at the training phase for the sample diversity while at the testing, actions are chosen greedily for better performance. After 100 epochs, when we fix the policy parameters and use a greedy decoder strategy for both testing and training, the cumulative rewards match rather well even without the overfitting issue. Nevertheless, all curves shown in Figure 2.5 demonstrate a convergent training process finalized with stable performance regarding the cumulative reward, actor loss and critic loss.

2.7.4 Comparative Study with Deep Auction

With the well-trained networks, the proposed deep auction-based approach is evaluated over 3 different mission dimensions $N = [10, 20, 50]$ in comparison with three

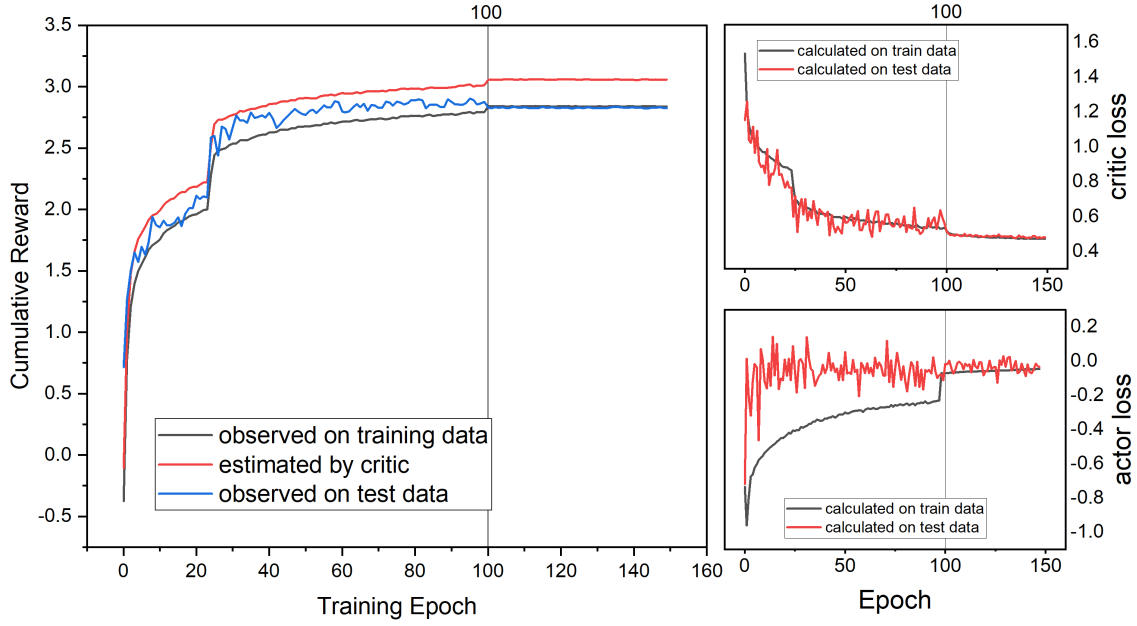


Figure 2.5: Learning curves of training actor and critic networks: (left) cumulative reward on training and test datasets per epoch; (right-top) critic loss on training and test datasets per epoch; (right-bottom) policy loss on training and test datasets per epoch.

baselines:

- The first baseline is the original CBBA, which is decentralized deployed as our method but uses deterministic reward functions.
- The second baseline considers a hybrid approach that integrates the CBBA task coordination method and the proposed DRL route planner, partially implementing the approximators present in Section 2.5, termed CBBA-DRL.
- A centralized DRL-based approach, namely MARDAM, proposed in [155], is also included in the comparison. To the best of our knowledge, there is no decentralized instantiation of DRL on routing problems yet due to its computational complexity, so we refer to this centralized approach as a representative of the DRL domain.

Simulation results for solving deterministic instances are shown in Table 2.7, where we assume that delivery processes are identical to the ones planned, where the drone keeps a constant speed, and the time cost on transport is exactly as expected. In these cases, among the decentralized approaches, including Deep Auction, CBBA, and CBBA-DRL, the CBBA provides the best planning solutions for each problem dimension. When using the proposed Deep Auction method, there is a performance

decrease of about 2%. Among all methods under consideration, the centralized MARL, i.e., MARDAM, provides the highest quality of solution, which benefits from its global scope of planning.

Table 2.7: Comparison of cumulative rewards for solving deterministic drone delivery problems with time windows of different dimensions. The value is averaged over 1000 randomly generated instances.

Mission setting \ Method		Deep Auction		CBBA		CBBA-DRL	MARDAM
Dro. count	Cust. count	Exp. reward	Act. reward	Exp. reward	Act. reward	Act. reward	Act. reward
2	10	8.5477	8.42	8.55	8.55	8.28	9.020
4	20	18.396	18.04	18.78	18.78	17.21	19.080
10	50	48.108	46.93	48.24	48.24	41.29	48.070

Table 2.8: Comparison of the total coordinating time required by the proposed method and the original CBBA for solving 1000 validation instances of drone delivery problems of different dimensions.

Dim. \ Method	Cust. Count = 10	Cust. Count = 20	Cust. Count = 50
Deep Auction	2'23"	12'49"	2h2'32"
CBBA	5'24"	54'43"	16h28'32"

Table 2.8 compares the computing time of the proposed Deep Auction method and the CBBA for solving 1000 validation instances of each dimension. As indicated from the table, CBBA consumes a longer time to solve the problems than the proposed Deep Auction, and it is noted that the differences become larger as the dimension increases, as shown in Figure 2.6. The proposed Deep Auction shows a more moderate growth than the CBBA does, which is consistent with the complexity analysis in Section 2.6 and further exemplifies the better scalability of the proposed approach.

The second series of simulations is conducted regarding the performance in a stochastic environment. Variances of flight speed $\sigma_v^2 = 0.05, 0.1, 0.2$ are added to the environment, indicating different levels of uncertainties. Simulation results are depicted in Table 2.9.

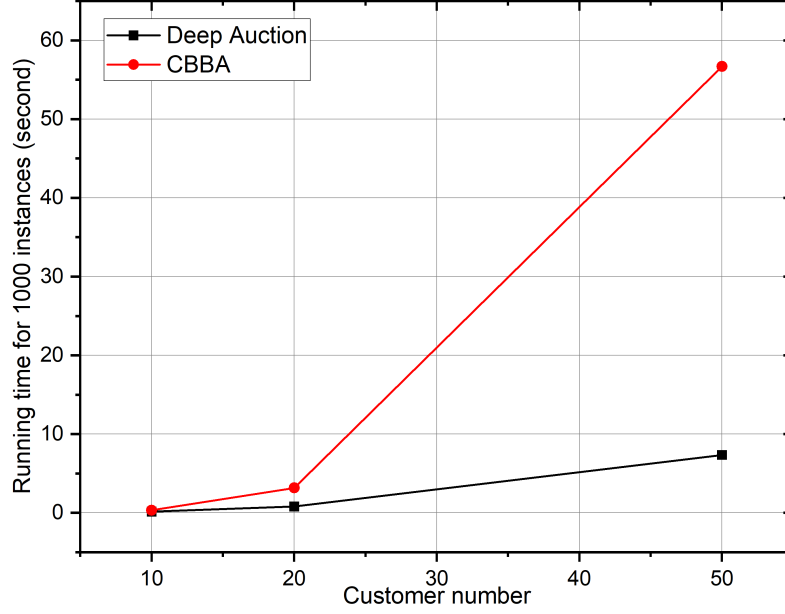


Figure 2.6: Running time for Deep Auction and CBBA as the customer number changes.

Table 2.9: Comparison of cumulative rewards for solving stochastic drone delivery problems with time windows of different dimensions. The stochastic feature of deliveries is emulated by sampling the drone speed from a normal distribution defined by different variances. Larger variances indicate higher randomness. The value is averaged over 1000 randomly generated instances.

Mission setting \ Method			Deep Auction		CBBA		CBBA-DRL	MARDAM
Dro. count	Cust. count	σ_V^2	Exp. reward	Act. reward	Exp. reward	Act. reward	Act. reward	Act. reward
2	10	0.05		8.499		8.216 (-3.33%)	7.987	9.043 (+6.40%)
		0.10	8.291	8.448	8.55	7.795 (-7.73%)	7.689	8.992 (+6.44%)
		0.20		8.165		6.758 (-17.23%)	7.434	8.678 (+6.28%)
4	20	0.05		17.923		17.007 (-1.24%)	16.448	18.864 (+5.25%)
		0.10	17.437	17.805	18.78	16.603 (-6.75%)	16.109	18.761 (+5.37%)
		0.20		17.181		13.936 (-18.89%)	15.027	18.366 (+6.90%)
10	50	0.05		46.886		44.799 (-4.45%)	39.629	47.746 (+1.83%)
		0.10	46.072	46.676	48.24	41.558 (-10.96%)	38.765	47.582 (+1.30%)
		0.20		45.466		34.275 (-24.61%)	35.967	46.909 (+3.17%)

If we vertically compare rewards in Table 2.9, the DRL-series methods perform

stably against different speed variances, in contrast with the significant performance degradation using deterministic CBBA as the variance increases. We attribute the increased robustness to the newly formulated score function in the Deep Auction and the observation evolvment in MARDAM.

Second, comparing the rewards horizontally, the proposed Deep Auction gains a better performance compared to the CBBA, while being inferior to the centralized MARDAM method. The inferior performance is within expectation, as centralized methods have global situation awareness and are more likely to choose better actions than decentralized approaches. However, as previously noted, centralized methods have inherent limitations, due to the central-node dependency and the high communication requirement, as well as the poor scalability which is also demonstrated by the simulation results: the MARDAM loses its superiority incrementally as the problem size increases.

At last, in terms of CBBA-DRL, which is an additional reference incorporating the CBBA for task coordination and a DRL planner for task routing, its performance is poorer than the original CBBA with small variances of 0.05 and 0.1, and surpasses the CBBA with $\sigma_V^2 = 0.2$. This demonstrates the robustness of the DRL planner under uncertainties, though there is a significant negative impact due to inconsistency between the coordination method and the route planner.

2.8 Conclusion

This chapter presents two decentralized auction-based approaches addressing the task coordination problem of multi-agent systems while considering the possible operation uncertainties. Through casting the problem as task-constrained MDPs, the task dependency due to an exclusive constraint is despatched from M-MDPs and then resolved by adopting the auction-based coordination method. For these multi-agent stochastic planning problems, the suggested technique resolves the trade-off concern between computational tractability and solution quality. Further, this algorithm is proven to be convergent and can reach a prior optimality guarantee of at least 50% with submodular reward functions. The second algorithm, namely Deep Auction, is an approximate modification of the first suggested auction-based coordination system that includes two neural network approximators to facilitate large-scale implementations. By theoretical analysis, these two proposed algorithms achieve more robustness and less computing complexity, compared to the original

CBBA with respect to the score function calculation and to the centralized MARL in terms of the problem dimensions. Finally, a case study of drone delivery with time windows is studied. The simulation results demonstrate the theoretical benefits of the recommended methodologies.

Chapter 3

Edge-enhanced Attentions for Route Planning

In this chapter, a specific DDP-R will be presented, and an AM-E model is proposed based on the DRL method to address the DDP-R. A stochastic drone energy cost model is also established for validation. Extensive simulations are conducted to demonstrate the performance of the proposed AM-E in comparison to other state-of-the-art approaches in both non-wind and windy cases.

3.1 Introduction

Rapid developments in drone technology opened the way for a wide range of civilian applications, e.g., traffic surveillance, medical delivery, and general warehouse, which comprise an important component of the *Intelligent Transport System*. Meanwhile, worldwide authorities began to authorize flights beyond the visual line of sight to test the safety of opening up the technology to the wider industry. Integrating drones into ITS offers increased flexibility and efficiency to transportation, which brings high societal and economic benefits.

Despite their advantages, the introduction of drones as new couriers into the logistics market implies new challenges in terms of logistics planning due to their limited onboard battery and load capacity. The VRP is a generic optimization class for modelling multi-vehicle logistics with the objective of seeking the best route for a fleet of vehicles to serve a set of customers [156]. To avoid exceeding the limited flight time of Li-Po battery-equipped drone couriers, an MTVRP for drone deliv-

ery are proposed in [48], which compensates by making multiple trips to the depot while subjecting to a budget constraint. However, in MTVRP, the serving range of delivery is still restricted, and drones continuously return the depot resulting in low efficiency. Motivated by this, a new area of research is arising that investigates the placement of recharging stations to extend drone flight distance and maximize covered demand with the least amount of infrastructure investment [157, 158], which benefits drone delivery with greater coverage, lower costs, and higher quality delivery services, and is also eco-friendly by reducing unnecessary return flights. The routing problem considering recharging is termed as the EVRP [159], also known as the *Green Vehicle Routing Problem*, and was originally proposed for electric cars similarly suffering from limited battery capacity. Furthermore, in addition to the Amazon ‘beehive’ delivery pattern, where all drones pick up parcels from a shared giant centre, a delivery request is likely to have its own pickup and delivery locations, such as those in a medical delivery service [138]. This type of route planning problem can be described as PDP, another variant of VRP characterized by pairing and precedence relationships. To address the unmanned aerial logistics while both considering recharging facilities and heterogeneous pickup and delivery requirements, a combination of EVRP and PDP is investigated in this paper. Moreover, compared to ground-based vehicles, UAVs are severely affected by weather conditions, especially airflow, introducing stochastic factors to the execution environment. Accommodating this uncertainty in planning tends to increase the quality of routes by reducing risks and stabilizing deliveries.

To summarize, the DDP-R considered in this paper is essentially an optimization problem that finds the best routes for serving pickup and delivery tasks while taking into account load and battery constraints as well as the randomness introduced by weather. Due to its NP-hard and stochastic nature, conventional methods including exact and heuristic algorithms struggle to efficiently address this routing problem [160]. Recently, there has been increasing attention on DRL for solving combinatorial optimization problems, which has delivered promising results on the basic VRP and some of its variants [154, 161]. To this end, this paper explores learning techniques for solving the drone delivery problem considered.

Since most DRL-based approaches are proposed for typical VRPs and have few actual applications for stochastic environments, two key challenges are accordingly identified on the above-mentioned drone delivery problem. Firstly, nodes in the DDP-R involve more heterogeneous roles, consisting of pickup tasks, delivery tasks, charging stations, and the depot. Current network architectures are not intended

to capture their sophisticated characteristics and connections. Secondly, routing problems with drones are vulnerable to weather. Performance degradation is likely to happen due to discrepancies between the deterministic planning model and the actual stochastic system in the presence of wind.

To address these challenges, we developed a routing simulator for DDP-Rs in the presence or absence of winds and propose a novel neural network architecture with a stronger ability of description. To summarize, the main contributions are as follows:

- 1) To the best of our knowledge, this paper is the first attempt to solve the DDP-Rs via the RL method. A good policy is sought by devising the evolution of the environment, masking scheme, and reward function. Through the comparison of state-of-the-art algorithms, the proposed DRL-based planning method is shown to provide high-quality solutions (within a 5% gap to optimal solutions in terms of both TSPs and PDPs) and demonstrates robustness in the presence of stochastic winds.
- 2) A policy neural network called AM-E is proposed for solving DDP-Rs, where an edge feature matrix is additionally considered to offer the network greater description power in terms of elaborate node connectivity extensively without increasing its computational complexity.
- 3) In contrast to existing studies, this paper also considers the pickup-delivery task configuration, the effect of wind variations, and the presence of charging stations in formulating the drone delivery problem for the most practical purposes. The newly formulated problem has a stochastic nature and complex node relationships that pose difficulties for traditional methods. Therefore, we further describe the problem in the form of an MDP [145], which facilitates the generation of AM-E policies.

The rest of the paper is organized as follows: Section 3.2 provides an overview of related works based on conventional methods and learning techniques. Section 3.3 gives an explicit description of the routing problem of concern, including the drones' energy consumption model. This is followed by the proposed DRL-based in Section 3.4. Section 3.5 presents details of the demonstration experiments, in which the experimental settings are described and the evaluation results under deterministic and stochastic cases are presented. Section 3.6 offers conclusions of this study and outlines a brief plan for future research.

3.2 Related Works

In this section, we introduce readers to existing works that have addressed vehicle routing problems using conventional methods or using learning-based methods, with a focus on its two variants: pickup and delivery problems with recharging and stochastic routing problems.

3.2.1 Exact and Heuristic Methods

Since the VRP is an NP-hard problem, either EVRP or PDP is a generalization of the VRP, thus are equally considered NP-hard in the strong sense [156]. Due to the complexity of the problem, studies using exact approaches are rarely found in the literature, and most of them adopt branch-and-bound algorithms or their variants. A branch and cut algorithm is proposed in [162] to solve the EVRP. Similarly, a branch-price-and-cut algorithm was used in [163] to solve the EVRP with time windows. Various branch-and-price algorithms were designed, targeting specific extended versions of EVRP, e.g., with heterogeneous stations, partial charging policy, and non-linear charging rate [164]. In terms of PDP, extra bounding procedures or column generation schemes are introduced to the basic branch-and-bound algorithm to meet the pickup-delivery constraints [165, 166]. It is worth mentioning that based on these methods, some commercial solvers, such as CPLEX and GUROBI, are broadly employed to find the optimal solution for small-sized VRPs and their variants. However, for large-scale problems, exact methods suffer from the computational complexity of the problem.

One trend in solving VRPs is to use metaheuristics, with the purpose of tackling the problem approximately but efficiently. Typical metaheuristics include neighbourhood search, simulated annealing [167], tabu search, evolutionary metaheuristics [168] and genetic algorithms [169]. Rastani *et al.* [170] integrated an optimal repair procedure in a large neighbourhood search heuristic method for solving the load-dependent EVRP with time windows. Another adaptive large neighbourhood search method was proposed in [171] to solve PDP based on the destroy and re-create principle. Particularly, stated as the first solution to PDP with electric vehicles, a granular tabu search algorithm is developed in [172] by restricting the neighbourhood of search by creating a promising arc. Although these heuristic methodologies perform well in offline situations, most of them cannot be directly employed in real-time or dynamic cases, since they fail to efficiently deal with unforeseen changes,

i.e., incoming requests. In terms of stochastic routing problems, the problem is modelled as a stochastic VRP, which is then handled by simulation-based optimization techniques for maximizing the expected objective function. However, the increase in complexity due to stochastic factors makes the problem even harder to solve in real time.

3.2.2 Learning-Based Approaches

In recent years, great potential has been found to employ DRL to resolve routing problems. To use DRL, the route is constructed by appending next visit nodes, formulated as a sequential Markov decision-making process. Graph-based policy networks and sequence-to-sequence networks are found in the literature to map the state space to action probabilities.

The Pointer Network is the first seminar work to cope with the routing problem via learning, which solves TSP via recurrent neural networks [173]. The PtrNet is firstly implemented in a supervised manner and later extended into an RL framework [174]. Furthermore, the work in [175] generalizes the application of PtrNet to a wider range of CO such as CVRP, where element embeddings could be dynamic. After that, more effective DL architectures are proposed to represent problem statements by combining the transformer-based attention model [154, 155] or graph embedded structure [161, 176], showing outperforming results in comparison with heuristic methods. A stage has arrived where deep neural networks can effectively extract useful information from customer configurations and obtain high-quality policies for typical routing problems through reinforcement learning.

Extending to variants of the vehicle routing problem, such as the PDP or EVRP, the above-mentioned networks could be directly employed with a redesigned reward scheme and mask policy. However, this is less effective for distinguishing different types of nodes and identifying their relationship with specific variants. To that end, Li *et al.* [177] propose a heterogeneous attention model for PDP, in which seven types of attention layers are sophisticatedly designed to consider different roles played by nodes while taking the precedence constraint into account. Lin *et al.* [176] incorporate the model of [175] with a graph-embedding component to yield the global information of the graph for EVRP with time windows. In addition to these efforts, another direction of enhancement is to integrate edge features in the graph neural networks. Instead of the one-dimensional binary adjacent matrix, Gong *et al.* [178] made efforts to exploit more complex edge features leading to a generic

edge-integrated framework. Despite its purpose to boost performance, edge-featured models are not always superior, especially for some node-sensitive tasks [179], which has actually motivated our proposal of an AM-E network.

On the other hand, the routing problem with stochastic model parameters has received increasing attention. Instead of assuming cost and customers are static and known a priori, taking into account the randomness of planning parameters leads to a higher quality of the solution. Since RL provides promising tools to optimize policies with stochastic state transitions, Bono *et al.* [180] developed an online representation of problem states enabling real-time planning based on the latest observation of vehicles. Machine learning techniques have also been used to build a probabilistic energy consumption model [181], achieving more energy-saving and reliability for routes.

3.3 Problem Statement

This section describes the DDP-R problem, presents a stochastic drone energy consumption model in consideration of varying wind conditions, and then formulates DDP-Rs in the form of MDPs [182].

3.3.1 Drone Delivery Problem with Recharging

Suppose a drone delivery scenario where there are n customer requests and each of them is decomposed into a pickup task $i \in \mathbb{P}$ and a delivery task $i + n \in \mathbb{D}$. A drone is sent from the depot point with a fully charged battery and finally returns to the depot after accomplishing all tasks. During mission execution, the drone can get charged at any recharging station $j \in \mathbb{C}$. The aim of solving the drone delivery problem is to find a route that minimizes the energy cost while subjecting to constraints regarding the delivery mission and drone properties. Further assumptions used in this study are listed below:

- Each customer request is labelled with a predefined load weight.
- The drone only processes one request at a time.
- Only fully recharging is considered in this study.
- The recharge stations could be visited infinite times.

- The depot also serves as the recharging station during mission execution.

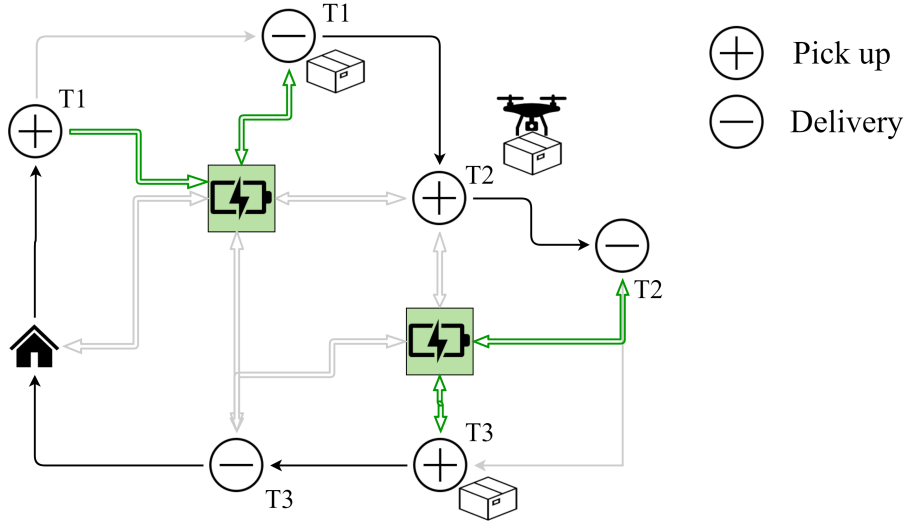


Figure 3.1: Drone delivery with recharging

3.3.2 Drone's Energy Consumption Model with the Wind Effect

For drone delivery, estimating energy requirements performs an essential role in route planning. Unlike ground vehicles, the drone's energy cost can be strongly affected by weather conditions, especially the wind speed and direction as they directly impact the flight kinematic model [36, 183]. Nonetheless, most route optimization models only incorporate energy consumption implicitly via a pre-defined limited drone duration or range. With the aim of better emulating the real-world drone delivery scenarios, we build a stochastic energy consumption model considering varying wind conditions, inspired by the work in [184] and [185].

Energy Consumption without Wind

The total aerodynamic power required for drone flight comprises four parts: parasite power, induced power, profile power and power required to climb [185]:

$$\begin{aligned}
P_{aero} &= P_{parasite} + \kappa_{ind} P_{induced} + P_{profile} + P_{gravity} \\
&= D_{body} v_a + \kappa_{ind} T w + \rho A v_T^3 \left(1 + 3 \left(\frac{v_a}{v_T} \right)^2 \right) \frac{\sigma c_{bd}}{8} + m g v_a \sin \gamma
\end{aligned} \tag{3.1}$$

where v_a is the airspeed, γ is the flight angle, κ_{ind} is the induced factor (usually equal to 1.15, see [185]), body drag force $D_{body} = q S C_{D_{body}} v_a^2$, thrust

$T = \sqrt{m^2 g^2 + D_{body}^2 + 2 D_{body} m g \sin \gamma}$, w is the downwash coefficient, A is the rotor dis area, v_T is the blade tip speed, σ is the rotor solidity ratio and c_{bd} is the blade drag coefficient. Details about determining w , A , v_T , σ and c_{bd} are referring to [185].

The overall power during flight includes aerodynamic power (conditioned by the power efficiency η) P_{aero} and the hotel power (the power required to supply internal electronics) P_{hotel} [185]:

$$P = \frac{P_{aero}}{\eta} + P_{hotel} \tag{3.2}$$

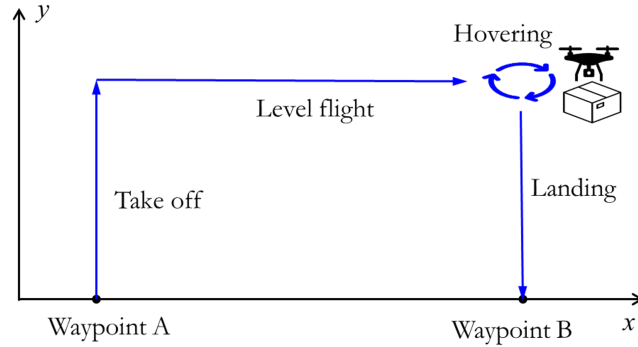


Figure 3.2: The flight profile for the VTOL drone delivery, consisting of four phases: takeoff, level flight, hovering and landing.

For delivery problems, we identify the flight process as having four phases: take-off, level flight, hovering and landing. A typical flight profile is shown in Fig 2. To estimate the total energy demand, the power demand for each phase is weighted with an associated duration:

$$\begin{aligned}
E &= \frac{1}{\eta} \left(t_{tof} P_{aero} (m, v_a^{tof}, \gamma_{tof}) + t_{lf} P_{aero} (m, v_a^{lf}, 0^\circ) \right. \\
&\quad \left. + t_{hover} P_{aero} (m, |v_w|, 0^\circ) + t_{land} P_{aero} (m, v_a^{land}, \gamma_{land}) + t_{total} P_{hotel} \right)
\end{aligned} \tag{3.3}$$

Where $|v_w|$ is the wind speed, t_{tof} , t_{land} indicates the time for takeoff and landing, t_{hover} is time for hovering, t_{lf} is time for level flight, and t_{total} is the total duration.

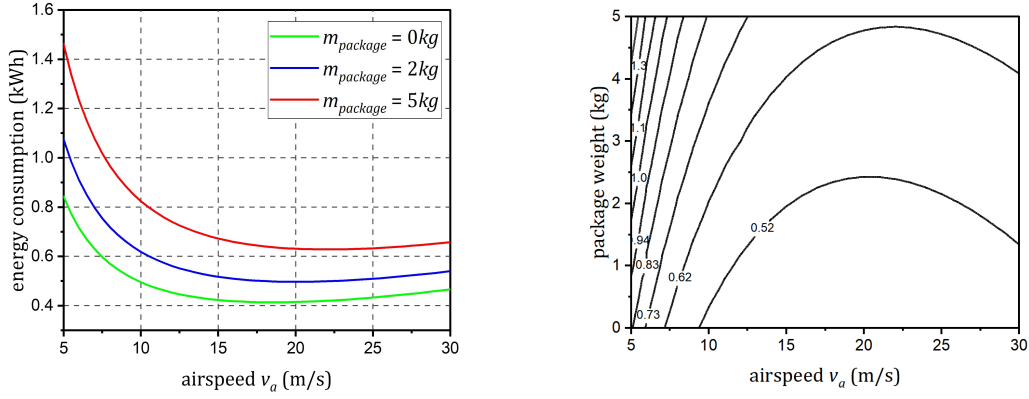


Figure 3.3: Energy consumptions for a 10km flight with airspeed and package weight varying. In the first figure, three cases are considered with package weight $m_{load} = 0, 2\text{kg}, 5\text{kg}$ respectively. In the second figure, a contour map of energy consumption is depicted with regard to airspeed and package weight. In simulations, no wind is considered, altitude for level flight $a = 100\text{m}$, hovering time $t_{hover} = 5\text{s}$, and vertical speed is assumed as a constant $v_v = 3\frac{\text{m}}{\text{s}}$.

In this paper, we consider the drone with vertical take-off and landing, indicating that the ascent angle $\gamma_{tof} = 90^\circ$ and the descent angle $t_{land} = -90^\circ$.

Given the altitude of level flight a and the UAV ground speed v_g ($v_g = v_a$ with no wind), the time for level flight is $t_{lf} = \frac{d}{v_g}$, and the time for takeoff and landing is $t_{tof} = t_{land} = \frac{a}{v_v}$, where v_v is the vertical speed. Figure 3.3 depicts the energy consumption curves for a 10 km delivery flight with three different package weights and airspeed varying (left), and its contour map with respect to both airspeed and package weight (right).

Energy Consumption in Presence of Wind

According to the above model, the energy cost for a specific delivery is defined by three variables: package weight m_{pack} , airspeed v_a , and flight time t . Further, assume that the drone is directed by a constant ground speed command, the airspeed is then obtained by integrating the velocity of wind and the ground speed command, which is derived from the drone kinematic model [186]:

$$\begin{aligned}\dot{x} &= v_g \cos \chi = v_a \cos \psi + v_w \cos \psi_w \\ \dot{z} &= v_g \sin \chi = v_a \sin \psi + v_w \sin \psi_w\end{aligned}\tag{3.4}$$

where v_g , v_a and v_w indicate the ground speed, airspeed and wind speed respectively, $\chi = \tan^{-1} \left(\frac{z}{x} \right)$ is the course angle, ψ is the heading angle of the UAV, and ψ_w is the wind course angle. Due to the constant ground speed, the airspeed could be derived as

$$v_a = \sqrt{2v_g^2 + 2v_w^2 - 2v_g v_w \cos(\psi_w - \chi)} \quad (3.5)$$

In terms of wind modelling, we establish a stochastic wind model in which a constant wind is combined with a random turbulence flow. The constant wind speed is randomly generated via a Weibull distribution, which is widely used to describe wind variation [187]. The scale parameter and shape parameter in wind Weibull distributions indicate the mean value of wind speed and distribution shape, respectively. The direction of the constant wind is assumed to follow a uniform distribution from $\psi_w = 0^\circ$ to $\psi_w = 360^\circ$. The turbulence wind component is generated through the well-known Von Karman turbulence model [188].

To analyze the distribution of energy consumption under varying wind conditions, we conduct a series of Monte Carlo simulations given a single-way waypoint task, where the distance to the target position is 10km , and the flight is unloaded. After running simulations of 10000 rounds each with different airspeeds and average wind speed, the statistical figure of the energy consumption is depicted in Figure 3.4.

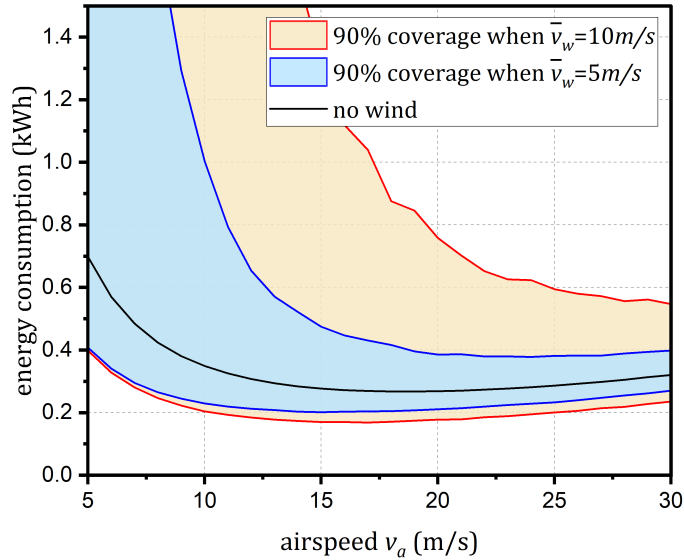


Figure 3.4: Statistical distribution of energy consumption for a 10km flight with the wind. In Monte Carlo simulations, the shape parameter of the wind Weibull distribution is set as $k = 2$.

The 90% confidence intervals of energy consumption are calculated and depicted as the shaded areas in Figure 3.4, where the energy cost value will fall inside with the probability of 90% when the average wind speed is 5m/s or 10m/s . As shown in

the figure, these two shaded areas are both unignorable. We can also find out that when airspeed increases, the UAV could resist the wind to some degree, leading to a smaller variance. However, even with $20m/s$ airspeed, winds having an average speed of $5m/s$ could also cause a difference in the energy cost up to $0.2kWh$ for a $10km$ path segment, and the deviation even becomes larger as wind speed increases. This implies that ignoring variances in energy consumption caused by random wind conditions will degrade the performance of planned routes and even lead to task failures. Meanwhile, in many practical applications, such as drones operating in built-up urban areas or in windy outdoor areas with complex terrain, it is always impossible to obtain accurate wind models. This emphasizes the importance of introducing the stochastic energy consumption model in planning to make the results with better wind resistance.

3.3.3 Markov Decision Process for DDP-Rs

For the purpose of handling randomness, we model the stochastic DDP-R as sequential MDPs. This idea, borrowed from path planning algorithms, is naturally well suited for stochastic problems, enabling us to update the route based on the evolution of available information.

A generic MDP consists of four components: state space S , action space A , immediate reward function $r(s, a)$ and state transition probability $p(s'|s, a)$. Specifically, for the DDP-R described above, these components are defined as follows:

- *State*: The global state $s_i \in S$ is factored into the drone state s_i^D (location, remaining energy) and the mission state s_i^M including locations and status of customer requests, recharging stations and the depot. The mission state is further decomposed into the vertex features $\mathbf{s}^v$ (location) and the edge features $\mathbf{s}^e$ (distance, connectivity).

$$\mathbf{s} = (\mathbf{s}^D, \mathbf{s}^M = (\mathbf{s}^v, \mathbf{s}^e)) \in S \quad (3.6)$$

where

$$\begin{aligned} \mathbf{s}^D &= [x_d, y_d, soc] \\ \mathbf{s}_j^v &= [x_j, y_j, w_j] \\ \mathbf{s}_{ij}^e &= [d_{ij}, a_{ij}] \end{aligned} \quad (3.7)$$

where (x_d, y_d) and (x_j, y_j) is the coordinate of the drone and the vertex j , soc is the state of charge level of the drone, w_j is the weight of the package, d_{ij} is the distance between vertex i and j , and a_{ij} is a binary variable, indicating directional connectivity from vertex i to vertex j .

It is noted that there are two possible route ends:

- a. All customer requests have been served.
- b. Planning steps reach the maximum step limitation.

When the route comes to an end, the drone is forced to go back to the depot and the state arrives at the goal state $s_f \in S_f$.

- *Action*: The action indicates the next movement of the UAV, represented by the corresponding vertex: $a = j, j \in \mathbb{N}$. It could be the depot, a task or a recharge station. With the purpose of improving the exploration efficiency, the set of actions is partitioned into valid and invalid sets and a masking policy is employed to mask invalid actions. The details of the action masking policy will be introduced in Section 5.3.
- *Reward*: The reward function determines the return value obtained from the environment after the system acts. The design of the reward mechanism is critical, which directly guides the training of the RL agent. In the DDP-R problem, our aim is to minimize the overall energy consumed, which accumulates the energy cost on each path edge. We define an energy cost function ϵ for the drone traversing an edge e , which depends on 1) the payload l , 3) wind conditions w . Thus, the reward associated with each travel is set as the negative energy cost of the travel under the configuration (l, w) :

$$r(s, a) = -\epsilon(e; l, w) \quad (3.8)$$

However, with a negative reward function like this, there is an incentive for the UAV to stay at the depot forever to avoid all costs. To encourage the UAV to serve more tasks, an additional negative reward is imposed for tasks left pending when the drone is back at the depot. Specifically, a constant penalty variable C_{pen} is imposed for every task that is left unfinished at the end of the episode, which is denoted as a terminal reward R^g :

$$R^g(s_t) = \begin{cases} -C_{pen}n_{pen} & \text{if } s_t \in S_f \\ 0 & \text{otherwise} \end{cases} \quad (3.9)$$

where n_{pen} counts the number of pending tasks when the state arrives at s_f .

- *Transition:* A transition function describes how states are updated, represented by the probability $p(s'|s, a)$ that the current state s is converted to the next state s' by taking the action a . When the UAV travels to the vertex j after choosing action $a = j$, the drone's state and the mission state get updated according to the type of node the drone visited.
 - 1) If the targeted vertex is a task, the status of the drone is updated according to its new location and the energy consumption for the journey, using the model described in Section 3.2.
 - 2) If the targeted vertex is a recharge station or the depot, the mission status remains unchanged, while the location of the drone moves to the targeted vertex and the energy level of the UAV returns to full.

3.4 Reinforcement Learning Model

This section describes details about the design of a DRL-based algorithm to solve the problem formulated, including the architecture of the policy network, the masking policy, and the implemented RL training method.

3.4.1 Edge-Enhanced Attention Model

The policy network adapts the original AM model [154] for a better description of the considered DDP-Rs using a *dot-product attention layer with edges*, which is inspired by embedding techniques in *Graph Neural Network*. Most policy networks, as discussed in Section 2, were designed for homo-vertex problems such as classical TSPs and CVRPs, with only one depot serving as the starting and terminal points. However, in DDP-Rs, there are four different types of nodes (i.e., *pickup tasks*, *delivery tasks*, *depot*, and *recharging stations*), coupled with heterogeneous connections subject to the drone's payload limitation and the pickup-delivery precedence constraint. To describe the connectivity status of nodes, we build up an adjacent matrix, which together with the distance matrix forms edge features. The adjacent matrix is constituted of binary element e_{ij} which describes the connectivity from the node i to the node j . For further elaboration, a toy example is given in Figure 3.5, which contains 2 delivery tasks, 1 depot and 1 charging station. If we define the node order as the depot, stations, pickup tasks and delivery tasks, its adjacent

matrix will be constructed as the binary matrix in Figure 3.5. For instance, the first row in the matrix indicates that departing from the depot, the UAV is only expected to pick up parcels or get recharged for long-distance cruises instead of flying back or going to delivery destinations, while the first column indicates the vehicle can only return back to the depot from the delivery destinations and recharging stations. In terms of recharging stations, they are indirectly connected with all other nodes except from themselves. For pickup nodes, they are only connected to their corresponding delivery locations and recharging facilities, while for delivery tasks, they could direct to any other pickup positions with the empty vehicle except for its corresponding pickup task since the parcel has just been delivered.

To summarize, the logic behind the connectivity matrix comprises: the edge between paired requests is a one-way flight from pickup locations to delivery locations, the drone is not able to load another parcel while loaded, and the empty drone is not expected to fly to any delivery destination.

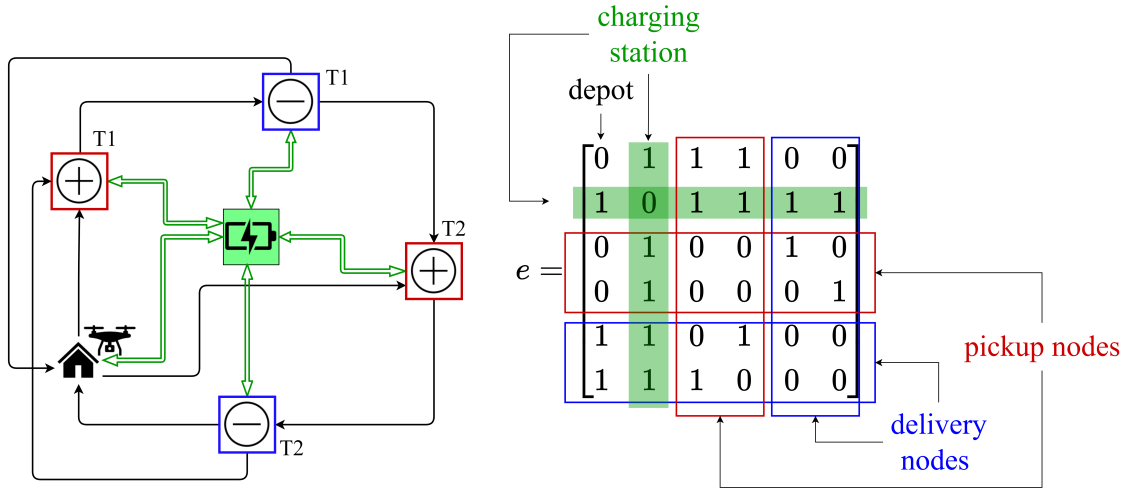


Figure 3.5: An example of the adjacent matrix in DDP-Rs. According to delivery logic and payload constraint, the connectivity matrix is defined as shown in the figure.

With the aim to efficiently capture the complex node relationship in DDP-Rs and enhance the network's description ability, the establishment of an adjacent matrix could be regarded as a manual process of feature extraction. To adapt edge features into a policy neural network model, we develop an *edge-enhanced dot-product attention layer* (shown in Figure 3.7) and integrate it into the Attention Model.

Edge-Enhanced Dot-product Attention Layer

Exploring edge information in the graph neural network, this paper is not the first trial. Gong *et al.* [178] have built a generic framework to sufficiently exploit edge features. The framework can consolidate both GCN and GAT, extending the one-dimensional adjacent matrix to multi-dimensional edge features. Based on the idea of [178], Wang *et al.* [179] and Hussain *et al.* [189] instantiates the edge augmentation with GAT and the transformer network respectively.

Though the intention of using extra edge features is to enhance performance, these edge-integration networks do not always perform superiorly. Simulations in [179] revealed that EGAT even has a slight performance degradation compared to GAT when executing node-sensitive tasks (e.g., *Cora*, *Citeseer*) though achieving higher accuracy on edge-sensitive tasks (e.g., trading network classifications). As analyzed in [179], this is due to the fact that in EGAT, edge features are updated by integrating their adjacent edges. These features, however, are most likely to be useless in node-sensitive tasks and interferences may occur because of the intensive updating. To avoid interference from other adjacent edges, we adopt another architecture of edge integration for the concerned routing problem which is identified as node-sensitive. The difference is illustrated in Figure 3.6. The EGAT model in [179] coupling embeds the node features H and the edge feature E by each GAT layer, while the proposed EGAT only updates the node features retaining the edge features the same for each layer. To implement the model, we propose an *edge-enhanced dot-product attention layer* which facilitates the network model to use edge features without updating it frequently.

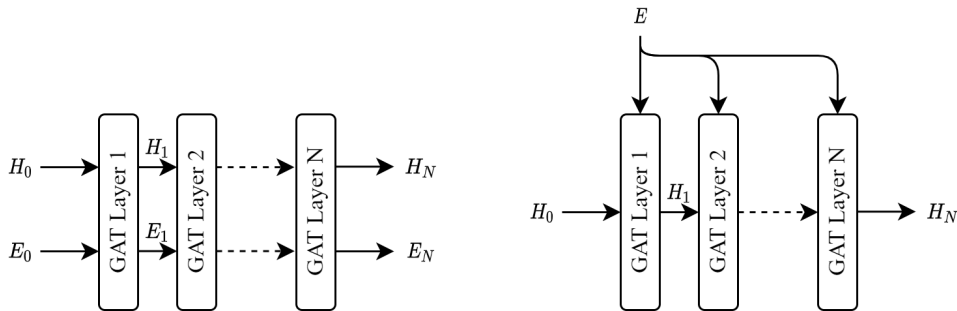


Figure 3.6: Two edge-enhanced GAT architectures. In the first architecture, both of node features and edge features are evolved for each attention layer, while the second one only updates the node features.

The attention mechanism performs a message-passing process over nodes of a graph, during which weights are added when integrating neighbourhood elements. In the

proposed edge-enhanced attention layer, edge features are taken into account when calculating weights and merging values. Specifically, an attention layer originally integrates the *value* of the node's neighbours v_i , and weights them using the *compatibility* of its *query* q_i with the *key* of the neighbour k_i . To take advantage of edge information, we integrate the embedded edge feature into the *compatibility* and also merge it into the *value* of neighbours, as illustrated in Figure 3.7.

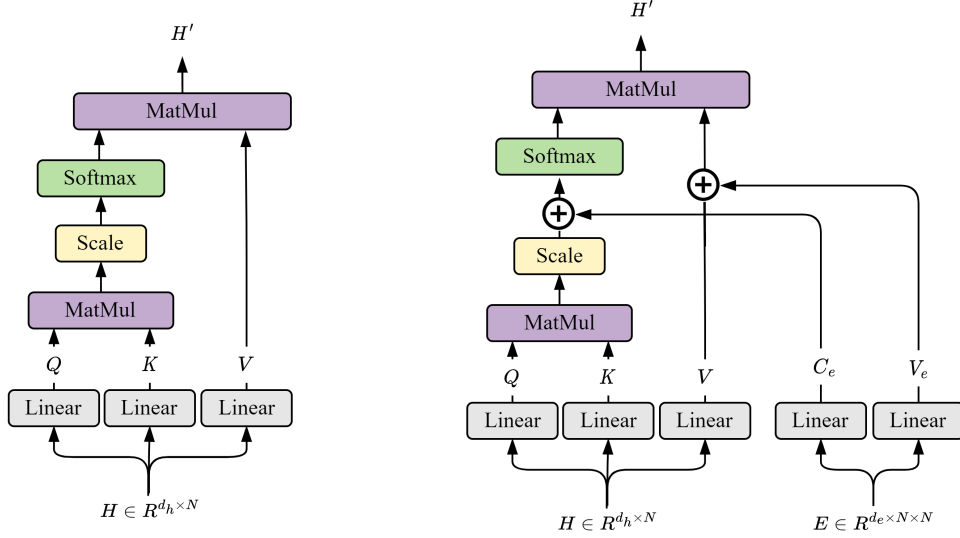


Figure 3.7: (left) Scaled dot-product attention. (right) Edge-enhanced scaled dot-product attention. In edge-enhanced scaled dot-product attention, edge features are projected and then merged into the original *value* vector and the *compatibility* vector, i.e., the scaled dot-product of *query* and *key* vectors.

Formally, the *key*, *value* and *query* for each node are computed by projecting node embedding h_i :

$$q_i = W^Q h_i \quad k_i = W^K h_i \quad v_i = W^V h_i \quad (3.10)$$

where $W^Q, W^K \in \mathbb{R}^{d_k \times d_h}$ and $W^V \in \mathbb{R}^{d_v \times d_h}$, d_k and d_v are designable dimensions.

In addition to node projection, edge embedding e_{ij} are also linearly projected, constructing the edge component for the *compatibility* and the *value* of each edge:

$$c_{ij}^e = W^{C_e} e_{ij} \quad v_{ij}^e = W^{V_e} e_{ij} \quad (3.11)$$

Here learnable parameter $W^{C_e} \in \mathbb{R}^{1 \times d_h}$ is used to generate a 1-dimensional *edge compatibility* matrix c_{ij}^e , and $W^{V_e} \in \mathbb{R}^{d_v \times d_h}$ has the same dimensions as the node weight W^V .

The *compatibility* $u_{ij} \in \mathbb{R}$ is then calculated as the sum of its edge component and

the normalized dot-product of the *query* from node i and the *key* from node j :

$$u_{ij} = \frac{q_i^T k_j}{\sqrt{d_k}} + c_{ij}^e \quad (3.12)$$

From the compatibilities, we compute the attention weights $a_{ij} \in [0, 1]$ using a softmax function:

$$a_{ij} = \frac{e^{u_{ij}}}{\sum_{j'} e^{u_{ij'}}} \quad (3.13)$$

Finally, the vector h'_i that is received by node i is the weighted sum of the *node value* v_i and the *edge value* v_{ij}^e :

$$h'_i = \sum_j a_{ij} (v_j + v_{ij}^e) \quad (3.14)$$

To enhance the capability of expression and also benefit the stability of the attention learning process, the single-layer attention is then extended to the *multi-head attention network*. Specifically, we compute M attention layers with independent parameters, using $d_k = d_v = \frac{d_h}{M}$, and then concatenate their outputs to the single d_h -dimensional vector. The final multi-head attention value for the node is:

$$h'_i = \parallel_{m=1}^M \sum_j a_{ij}^{(m)} (v_j^{(m)} + v_{ij}^{e(m)}) \quad (3.15)$$

where $\parallel$ represents concatenation, and superscript (m) indicates parameters obtained by the m th attention mechanism.

The Encoder-Decoder Architecture

Figure 3.8 depicts the overall architecture of our policy network, called *Attention Model with Edges* (AM-E). AM-E inherits the encoder-decoder structure of the Transformer model, which is recognized as the most competitive neural sequence transduction model [153]. Taking a toy instance (i.e., two deliver requests, one recharging station, and one depot) for example, the vertex features $[s_0^v \ \dots \ s_5^v]$ and edge features s^e are first passed to the dash line-blocked attention module after being initially embedded. Then the encoded features are recursively updated by the attention module N times until we get the output of the encoder: the embedded node

features $[h_0^N \dots h_5^N]$. Afterwards, using drone features s^D as the *query* vector and masking unavailable nodes, we could finally get the policy output $[p_0 \dots p_5]$, which is the policy probability of each node.

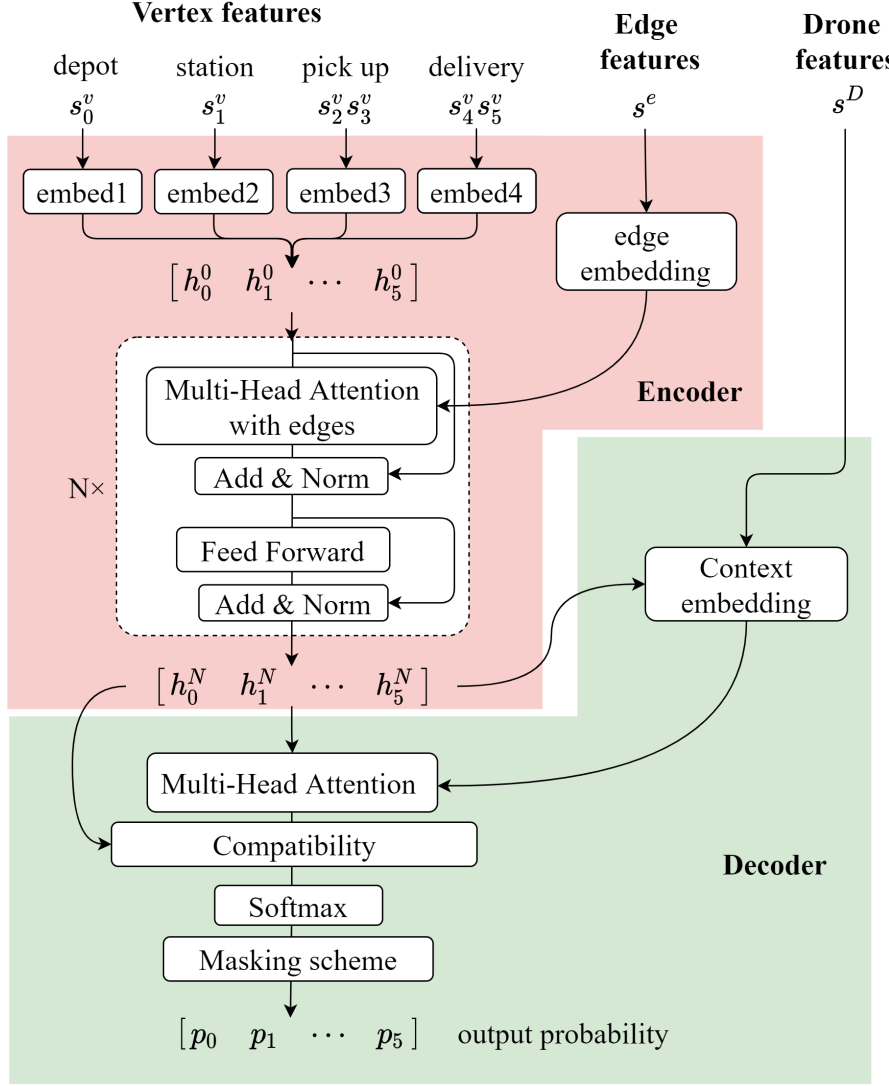


Figure 3.8: AM-E policy network.

Regarding the encoder, a similar structure to the Transformer [153] is adopted, which provides a mapping from nodes' raw features to a richer embedding space, through which the node's own features, the features of its neighbours, and the features of the edge connected with neighbours are all represented. Specifically, mission features $[s^v, s^e]$ are initially embedded to a larger dimension d_h via the node-wise linear projection. The projection parameters are only shared among features of the same category (pickup nodes, delivery nodes, recharging nodes and the depot). Then the embeddings $h_i^{(0)}$ are recursively updated by N multi-head attention layers, each

consisting of three operations: multi-head attention, feed-forward and normalization (see [154] for details on the rest of the encoder).

The decoder of the policy network leverages the node embedding from the encoder and generates a probability vector $\mathbf{p}$ for selecting nodes at each step. To achieve this, the graph is augmented with a special *context node* (c) to represent the decoding context. Herein, the context of the DDP-R consists of the embedding of the graph and the current state of the drone:

$$h_{(c)}^{(N)} = \left[h_{(g)}^{(N)}, h_{(d)} \right] \quad (3.16)$$

where the context $h_{(c)}$ is the concatenation of an aggregated node embedding $h_{(g)}^{(N)}$:

$$h_{(g)}^{(N)} = \frac{1}{n} \sum_{i=1}^n h_i^{(N)} \quad (3.17)$$

and $h_{(d)}$ is the linear projection of the drone state s^D :

$$h_{(d)} = W^{(d)} s^D + b^{(d)} \quad (3.18)$$

Then, the context embedding is computed using a multi-head attention mechanism, with a single query $q_{(c)}$ from the context node, and the keys and values from the node embedding of the encoder:

$$h_{(c)}^{(N+1)} = \text{MultiHead} \left(W_c^Q h_{(c)}, W_c^K h^{(N)}, W_c^V h^{(N)} \right) \quad (3.19)$$

Given the context embedding output $h_{(c)}^{(N+1)}$, we add one final attention layer to compute the output probabilities, for which we only compute the compatibilities with $q_{(c)} = W_c^Q h_{(c)}^{(N+1)}$ and $k_i = W_c^K h_i^{(N)}$. The compatibility is then clipped within $[-C, C]$ using a $\tanh$, which is followed by the masking policy described in Section 3.4.2:

$$u_{(c)j} = \begin{cases} C \cdot \tanh \left(\frac{q_{(c)}^T k_j}{\sqrt{d_k}} \right) & \text{if } j \text{ is valid} \\ -\infty & \text{otherwise} \end{cases} \quad (3.20)$$

Finally, after a softmax function, each node is scored with a probability as the final output, given as,

$$p_i = p_\theta(\pi_i | s) = \frac{e^{u_{(c)i}}}{\sum_j e^{u_{(c)j}}} \quad (3.21)$$

3.4.2 Masking Scheme

With the purpose to improve exploration efficiency and ensure the feasibility of solutions, the set of actions is masked by a designed masking scheme to exclude infeasible routes. According to the battery capacity constraint and the pickup-delivery precedence constraint, an action $a = v_j$ is labelled as valid only if all of the following conditions are satisfied:

1. If v_j represents a pickup or delivery task, the task must not have been served, and the UAV has enough energy to cover the trip to the vertex v_j and return to the closest charging station.
2. If v_j represents a charging station (or the depot), it should be reachable from the vertex where the UAV is currently located within its remaining battery capacity.
3. If the previous node visited is a pickup task, all other nodes except charging stations, the depot, and the corresponding delivery tasks are masked.
4. If the previous node visited is a delivery task, or if the UAV just departs from the depot, all delivery tasks would be masked.
5. If the last node visited is a charging station (or the depot), all station nodes and the depot node would be masked. Regarding task nodes, it depends on what type of node was visited before the charging station (or the depot) and then execute masking according to point 3 and point 4.

It is noted that even for stochastic cases in the presence of winds, the energy amount for paths is still calculated using non-wind energy prediction model as the wind information is assumed to be unknown prior to planning.

3.4.3 Training Method

Shown as Algorithm 3.1, we train the policy network using the REINFORCE algorithm with baselines. The baseline is chosen as rollout or critic according to whether the problem is deterministic or stochastic. In detail, at the beginning of each episode, one batch of new samples is stochastically generated. Then routes are sequentially sampled according to the probability output from the policy network and rewards are collected. Moreover, we get the expected reward $v(s_0^b)$ from the

Algorithm 3.1 REINFORCE learning algorithm**Input:** number of epochs N , batch size B **Initialize:** policy parameter θ , rollout baseline parameter ϕ and paired t-test threshold α if *rollout baseline*, critic baseline parameter θ_v if *critic baseline***for** each training *epoch* $= 1, 2, \dots, N$ **do**Generate instances of batch size B **for** each instance $b = 1, 2, \dots, B$ **do** /* ran in parallel */sample trajectory using policy π_θ receive the accumulative reward of the trajectory R^b **if** *rollout baseline* **then**receive baseline reward $v(s_0^b)$ using GreedyRollout policy π_ϕ **else if** *critic baseline* **then**estimate values of initial states using critic $v(s_0^b)$ **end if****end for**

estimate mean Policy Gradient on the batch of trajectories:

$$d\theta = \frac{1}{B} \sum_{b=1}^B (R^b - v(s_0^b)) \nabla_\theta \log \pi_\theta(\pi_b | s_0)$$

if *rollout baseline* **then**update policy parameter θ **if** *OneSidedPairedTest* $(\pi_\theta, \pi_\phi) < \alpha$ **then**replace ϕ using θ **end if****else if** *critic baseline* **then**

estimate the mean gradient of the MSE of critic on the batch of trajectories:

$$d\theta_v = \frac{1}{B} \sum_{b=1}^B \frac{\partial (R^b - v(s_0^b))^2}{\partial \theta_v}$$

update policy parameters θ and critic parameters θ_v **end if****end for**

critic network or the rollout baseline network, which is used to calculate the advantage component of the gradients. Backpropagation is then adopted to update the policy network and the critic network for the critic baseline. Regarding the use of rollout baseline, at the end of each episode, the parameter of the baseline policy network will be replaced by that of the policy network if the performance of the latter is significantly superior.

3.5 Numerical Simulations

To verify the effectiveness and evaluate the performance of the proposed AM-E model, we conduct simulations for solving DDP-R problems in the presence and absence of winds. In this section, we will introduce our experimental setup, and

investigate the performance of the proposed method in comparison to the original AM and other state-of-the-art solutions.

All experiments are carried out on a 16-core Intel E5-2620 v4 CPU, a Tesla K80 GPU, or a Tesla V100 GPU.

3.5.1 Experimental Setup

1) *Simulated Environment:* A simulated environment is built to imitate missions of the DDP-R, from which we can sample trajectories and get rewards back to train networks or evaluate the planning models. For initialization, pickup tasks, delivery tasks, charging stations, and the depot are randomly located on a $10km \times 10km$ mission area. The distance between every two nodes is set as the Euclidean distance. In simulations, the airspeed of the UAV is set to a constant $20m/s$ and a battery capacity of $1.5kWh$ is assumed. The energy cost on each edge is decided by the energy consumption model presented in Section 3.2 and normalized by the maximum battery capacity. The constant penalty for each pending task is $C_{pen} = 1$. Parameters of the constant wind component (i.e., direction and speed) are generated following the distribution model in Section 3.2 and assumed to remain unchanged for each path segment.

2) *Network Structure:* For the proposed AM-E model, the node embedding, edge embedding and context embedding are all one-layer element-wise liner projections with dimension 128. The multi-head graph attention network consists of $M = 8$ heads computing *key* vectors and *value* vectors of dimension $d_k = d_v = 16$. The number of sequential multi-attention modules is 3. In the feed-forward layer, the node features are passed through node-wise projections of one hidden sublayer with the ReLu activation and 512 hidden units.

3) *Training Parameters:* We adopt the Adam Optimizer to train the network with a constant learning rate $\alpha = 10^{-4}$. We run training for 100 epochs for each problem. In one epoch, we process 1.28 million instances in 2500 iterations with a batch size of 512. Each epoch takes around 15 min for training DDP20-R3 using a Tesla K80 GPU, 42 min for DDP40- R3 using a Tesla K80 GPU, and 28 min for DDP80-R3 using a Tesla V100 GPU.

3.5.2 Ablation Study of Attention Model with Edges

3.5.3 Evaluation Against Other Models

To justify the choice of the attention model as the base policy network, different models are evaluated on solving TSP. Models used in the comparison are Pointer Network (offline) [173], Pointer Network (A2C) [174], Graph embedding model [190], and Graph neural network [191]. A test dataset of TSP with 10,000 instances is used to validate models, with coordinates generated uniformly in the unit square. All results are compared against the optimal solution obtained by solving the mixed integer programming formulation using the Gurobi Software.

Results are presented in Table 3.1. As seen from the table, the attention model demonstrates the best performance across all three problem dimensions. Those results offer evidence that the attention model provides a structure that can effectively be learned and used to address combinatorial optimization problems.

Table 3.1: Results for solving TSPs using various models. Results are cited from each reference. Note that for the Graph embedding model, results are for 15-20, 40-50, and 50-100 node graphs.

Method	TSP20	TSP50	TSP100
Optimal	3.83	5.69	7.76
Pointer Network (offline)	3.88 (1.25%)	7.66 (34.56%)	-
Pointer Network (A2C)	3.89 (1.51%)	5.95 (4.59%)	8.30 (6.89%)
Graph Embedding Network	3.89 (1.47%)	5.99 (5.33%)	8.31 (7.01%)
Graph Neural Network	3.93 (2.7%)	-	-
Attention Model	3.84 (0.33%)	5.82 (2.28%)	8.19 (5.49%)

3.5.4 Validation of Edge Feature Enhancement

To test the efficacy of edge enhancement for solving DDP-Rs, we carried out a comparison study among three neural network models: the proposed AM-E model, the original AM [154], and the heterogeneous attention model [177]. As mentioned, the heterogeneous attention model is designed for PDP featuring a better performance by considering heterogeneous roles in PDP. It does not exactly match the DDP-R studied in this paper because of the presence of recharging stations, so an extension

is made that two more types of attention layers are added regarding pickup-recharge relations and delivery-recharge relations respectively.

The learning curves of the above three models are depicted in Figure 3.9. The total number of trainable parameters and the training time for one episode are summarized in Table 3.2. Simulation results in terms of the basic TSPs and the concerned DDP-Rs are shown in Table 3.3.

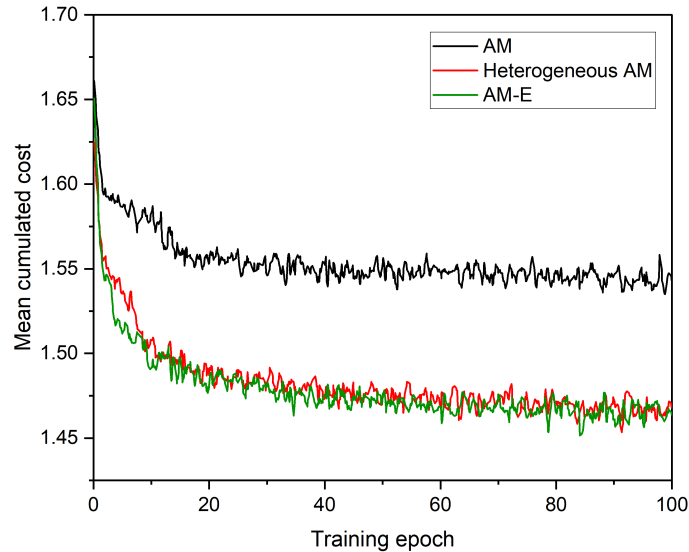


Figure 3.9: Learning curves for solving the DDP-R with 10 delivery requests and 3 recharge stations.

Table 3.2: Total trainable parameters of models and running time for one epoch*.

	AM	Heterogeneous AM	AM-E
Trainable parameters	693,632	1,086,848	746,112
Running time	4min 50s	8min 18s	6min 07s

*All Programs ran in Tesla V100, and time was measured over the entire training set and averaged.

From Table 3.3, in the case that all settings are identical except for model architectures, the AM-E and heterogeneous attention model show superiority over AM in solving DDP-Rs, with no significant difference is shown for TSPs with homogeneous and fully connected nodes. It should be noted that the heterogeneous attention model degenerates to the AM in solving the TSP, so we refer to the same results in the table.

Table 3.3: Comparison among AM, AM-E and heterogeneous AM*.

Task	Optimal	AM	Heterogeneous AM	AM-E
TSP20	3.83*	3.84 (0.33%)*	3.84 (0.33%)*	3.84 (0.33%)
TSP50	5.69*	5.82 (2.28%)*	5.82 (2.28%)*	5.81 (2.11%)
TSP100	7.76*	8.19 (5.49%)*	8.19 (5.49%)*	8.08 (4.12%)
DDP10-R3	1.483	1.590 (7.36%)	1.527 (2.97%)	1.526 (2.90%)
DDP20-R3	2.688	2.864 (6.55%)	2.788 (3.72%)	2.790 (3.79%)
DDP40-R3	-	5.286 (2.66)	5.170 (0.41%)	5.149 (0.00%)

*Results reported in Ref. [154].

For further comparison, the proposed AM-E model achieves comparable and even slightly better performance than the heterogeneous attention model with about a 32% reduction in parameters. In the heterogeneous attention model, 8 extra types of attention layers are added to the original one for solving DDP-Rs. Even though parameters of all *keys* and *values* are shared, the *query* matrices of attention layers are kept independently, which introduces a large number of extra trainable parameters to the model and almost doubles the running time for one episode. More trainable parameters always imply more computational amounts and larger memory requirements. In the proposed AM-E model, we only use one extra edge matrix to capture the node relationships, which requires much fewer trainable parameters and avoids complex network formulation while retaining comparable performance.

3.5.5 DDP-Rs without Winds

In the first experiment, we test the performance of the proposed AM-E model for solving deterministic DDP-R without considering the wind. The DRL agents are trained and tested with 3 different dimensions respectively (i.e., the number of delivery requests $N = 10, 20$ and 40 while the number of charging stations is fixed to 3) and compared to the original AM model and deterministic baseline methods.

The first baseline in comparison is obtained by Gurobi, a mathematical optimization solver, to provide the exact optimal solution for evaluation. To avoid unaffordable computing time, we set a 100-second time limit for the Gurobi. Under these limitations, Gurobi sometimes fails to obtain a feasible solution, so we also count the success rate for the Gurobi baseline. The second baseline comes from

Google’s Or-tools backed by a CP-SAT solver, which is usually seen as the most advanced heuristic solver. We gather Or-tools solutions of different qualities using the 1-second, 10-second, and 50-second solving time respectively.

Regarding the DRL implementations, we introduce the original AM model as another baseline. For the heterogeneous AM model, since the proposed AM-E model outperforms it in terms of both solution qualities and computing efficiencies as demonstrated by simulations in the last section, so we remove the heterogeneous AM model from the following comparison studies. With regard to the proposed AM-E model and the baseline AM model, we apply two decoding strategies for evaluation: 1) *Greedy*, always select the action with maximum probability at each step; 2) *Sampling*, sample $N = 1280$ routes for each instance according to the probability distribution and choose the one with minimum cost. The simulation results are summarized in Table 3.4.

Table 3.4: Costs and computing time for solving deterministic DDP-Rs

Method	DDP10-R3			DDP20-R3			DDP40-R3		
	Cost	Gap,%	Time,s	Cost	Gap,%	Time,s	Cost	Gap,%	Time,s
Gurobi (optimal)	1.483 (99.88%)	0.13	2.436	2.688 (98.90%)	0.00	4.306	4.925 (88.00%)	-	46.064
Or-tools (1s)	1.493	0.81	1.0	3.438	27.90	1.0	-	-	-
Or-tools (10s)	1.481	0.00	10.0	2.752	2.38	10.0	10.007	98.59	10.0
Or-tools (50s)	-	-	-	2.726	1.41	50.0	6.452	28.04	50.0
AM (greedy)	1.590	7.36	0.109	2.864	6.55	0.181	5.286	4.90	0.280
AM (sample1280)	1.521	2.70	0.141	2.741	1.97	0.251	5.094	1.09	0.380
AM-E (greedy)	1.530	3.04	0.063	2.776	3.79	0.121	5.131	2.18	0.416
AM-E (sample1280)	1.487	0.41	0.131	2.709	0.79	0.286	5.039	0.00	0.465

From Table 3, Gurobi fails to obtain a feasible solution in some instances within 100 seconds, and the proportion of these cases becomes larger as the problem size increases. Or-tools obtains high-quality solutions to small-size problems but scales poorly since the time required to search a high-quality route explodes as the number of delivery requests increases. Comparing these two DRL-based methods, the proposed AM-E model shows a performance improvement over its basic form, the AM model, both for greedy decoding and sampling decoding. Overall, AM-E has better scalability and efficiently produces high-quality routes for all three dimensions.

3.5.6 DDP-Rs in Presence of Winds

Next, the robustness of the proposed model is tested against stochastic edge cost in DDP-Rs under varying wind conditions. Regarding stochastic energy costs, if the drone's onboard battery is used up before it gets recharged, a penalty $p = 1.0$ will be imposed on the overall cost. The model is trained in a stochastic environment with the mean wind value $\bar{v}_w = 5m/s$, and then tested for robustness with $\bar{v}_w = 2m/s$, $5m/s$, $10m/s$. Since the DRL agent performs as an online policy, it is expected to continuously adjust the route according to its current battery level. However, the Gurobi and Or-tools, as offline planning methods, use predefined routes with less flexibility. Thus, in order to prevent battery exhaustion, a margin of safety is implemented to these baselines, i.e., the amount of available battery capacity used in planning is reduced by a specific percentage.

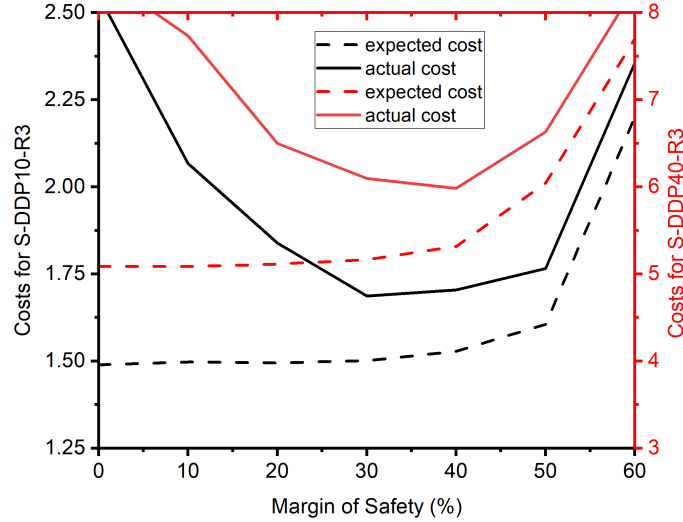


Figure 3.10: Expected costs and actual costs with different margins of safety.

To ascertain the best value of the margin of safety, we obtain the expected costs and actual costs via Or-tools using different margins for solving S-DDP10-R3 and S-DDP40-R3, presented in Figure 3.10. As seen in Figure 3.10, small margins turn out to be risky, resulting in large biases between expected and actual costs, whereas overly large margins are conservative, resulting in higher costs for frequent charging station visits. A margin of safety of around 20-40% provides the best planning results with the lowest costs in actual executions.

Table 3.5: Costs and computing time for solving stochastic DDP-Rs

$\overline{v_w}$	Method	S-DDP10-R3		S-DDP20-R3		S-DDP40-R3	
		Cost (Solvable rate)	Gap, %	Cost (Solvable rate)	Gap, %	Cost (Solvable rate)	Gap, %
$5m/s$	Gurobi (mgn. 20%)	1.783 ± 0.816 (99.90%)	5.82	3.161 ± 1.185 (97.80%)	4.81	6.423 ± 2.674 (41%)	-
	Gurobi (mgn. 30%)	1.716 ± 0.600 (99.89%)	1.84	3.145 ± 1.138 (98.30%)	4.28	6.209 ± 2.418 (52%)	-
	Gurobi (mgn. 40%)	1.685 ± 0.369 (98.91%)	0.00	3.016 ± 0.518 (99.30%)	0.00	5.456 ± 0.895 (49%)	-
	Or-tools (mgn. 20%)	1.839 ± 0.672	9.14	3.418 ± 1.581	13.33	6.495 ± 3.431	11.98
	Or-tools (mgn. 30%)	1.687 ± 0.300	0.12	3.341 ± 1.255	10.78	6.097 ± 3.045	5.12
	Or-tools (mgn. 40%)	1.703 ± 0.355	1.07	3.530 ± 1.478	17.04	5.983 ± 1.512	3.15
$2m/s$	AM (Greedy)	1.781 ± 0.007	5.70	3.234 ± 0.012	17.04	5.937 ± 0.019	2.36
	AM-E (Greedy)	1.744 ± 0.009	3.50	3.118 ± 0.011	3.38	5.800 ± 0.019	0.00
	Or-tools (mgn. 20%)	1.528 ± 0.211	0.00	2.806 ± 0.282	0.00	5.396 ± 1.089	0.00
	Or-tools (mgn. 30%)	1.544 ± 0.262	1.05	2.822 ± 0.327	0.57	5.782 ± 2.065	7.15
	Or-tools (mgn. 40%)	1.566 ± 0.333	2.48	2.829 ± 0.296	0.81	6.744 ± 3.673	24.98
	AM (Greedy)	1.657 ± 0.017	8.44	2.962 ± 0.021	5.56	5.511 ± 0.033	2.13
$10m/s$	AM-E (greedy)	1.584 ± 0.013	3.66	2.880 ± 0.019	2.64	5.458 ± 0.033	1.15
	Or-tools (mgn. 20%)	3.485 ± 2.237	64.54	6.482 ± 3.878	61.40	13.690 ± 8.028	71.17
	Or-tools (mgn. 30%)	2.874 ± 1.653	35.69	5.160 ± 2.648	28.48	11.038 ± 5.666	38.01
	Or-tools (mgn. 40%)	2.303 ± 0.962	8.73	4.246 ± 1.600	5.73	9.557 ± 4.559	19.49
	AM (greedy)	2.379 ± 0.051	12.32	4.216 ± 0.076	4.98	8.005 ± 0.137	0.08
	AM-E (greedy)	2.118 ± 0.032	0.00	4.016 ± 0.068	0.00	7.998 ± 0.143	0.00

mgn. = battery safety margin.

Table 3.5 compares the proposed AM-E model, AM model and offline planning baselines with the 20%, 30% and 40% margin of safety. It is noted that the results of Gurobi are excluded from the comparison for solving S-DDP40-R3 due to its high failure rate. Like deterministic cases, Or-tools gains solutions of lower costs for small-size deliveries while DRL methods perform better with scenarios of 20 requests or 40 requests. Setting certain margins for battery usage is the strategy mostly used in industrial applications, and it successfully guarantees a certain degree of robustness for these deterministic baselines, as shown in Table 3.5. However, how to define the margin value while operating in varying wind fields is a tricky question. It might either become risky or conservative. As an alternative solution, the DRL approaches to avoid this problem by continuously adjusting the route according to the UAV’s remaining energy.

Concerning the comparison with varying mean wind values, Or-tools obtains the best results with the wind $\bar{v}_w = 2m/s$ while AM-E shows better robustness when planning with windier scenarios. We also notice that or-tools, as an offline planner causes large variances when executing with $\bar{v}_w = 10m/s$, indicating that the drone fails to return to stations for some instances. At the same time, even though the AM-E policy was also trained with a milder wind field, it still remains a good performance and small variances with wind fields of $\bar{v}_w = 10m/s$ showing a certain transferring ability.

Overall, the proposed method obtains high-quality routes and arrives at the solutions in a very short time. The improvement becomes more significant when handling larger scales, demonstrating the AM-E with better scalability than Gurobi or Or-tools, as also seen by its stable performance and quick processing times. Again, AM-E offers better solutions than AM without consuming more time. Moreover, since the DRL-based method possesses the ability to respond to environmental changes, it is expected to perform better in dynamic scenarios compared to the deterministic approaches which require replanning of the whole routes.

3.5.7 Generalization with Dimensions

In all the above simulations, AM-E was trained and tested separately on each problem dimension, such that model weights are specialized on these dimensions. Despite that a route planner could transfer across these models according to current task requirements, we are still expecting a general model that performs well in all cases to facilitate future implementations. To that end, the proposed model has been

designed in a dimension-independent way, decorated with a node-wise embedding layer and a mean-reduction value layer for the critic baseline. Models are able to solve problems of other dimensions without parameter structure inconsistency.

Figure 3.11 and Table 3.6 record the intersecting validation results of models for solving delivery problems with $N = 10$, $N = 20$, $N = 40$, where the bottom axis in the figure refers to the problem size of the model learned, and the top axis refers to the problems to be solved. We observed that though the corresponding model achieves the smallest cost in comparison with those learned for other sizes, the differences across them are very small for all dimensions. More interestingly, we notice that even though the policy for DDP10 is only trained with the problems of small sizes, its degradation in solving DDP40 is still acceptable, which is quite meaningful in real-life situations as generalizing to larger problem sizes can save tremendous time and computing resources for large applications.

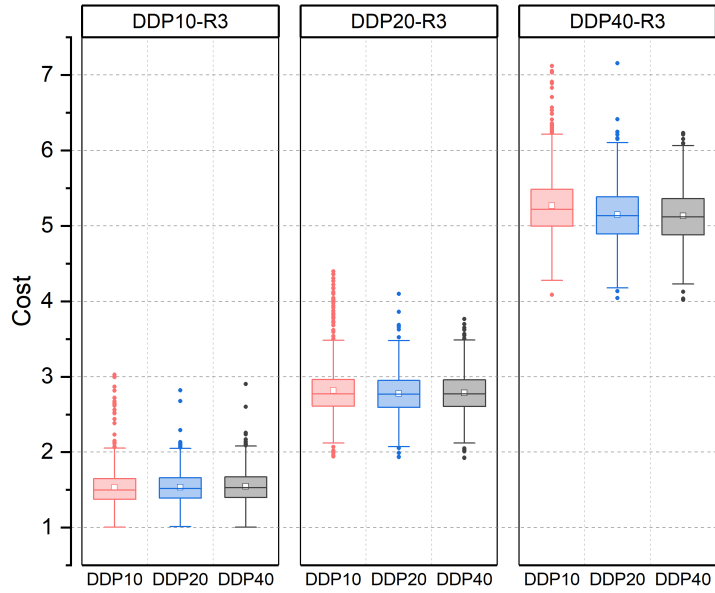


Figure 3.11: Generalization performance of AM-E.

Table 3.6: Costs using AM-E models trained with different dimensions

Method	DDP10-R3		DDP20-R3		DDP40-R3	
	Cost	Gap, %	Cost	Gap, %	Cost	Gap, %
AM-E (greedy) (DDP10)	1.530 ± 0.015	0.00	2.816 ± 0.022	1.44	5.266 ± 0.027	2.63
AM-E (greedy) (DDP20)	1.532 ± 0.013	0.13	2.776 ± 0.017	0.00	5.147 ± 0.023	0.31
AM-E (greedy) (DDP40)	1.544 ± 0.013	0.92	2.783 ± 0.017	0.25	5.131 ± 0.023	0.00

3.6 Conclusion

In this chapter, a new variant of the vehicle routing problem has been investigated, which is proposed to target the drone application for parcel delivery. The routing model includes recharging operations and considers the effect of varying winds. Specifically, we have built a simulated environment with a stochastic energy consumption model under varying winds to imitate real-world delivery scenarios, in which the energy-constrained UAV could be recharged in the middle of mission execution. To address the delivery problem efficiently, a DRL-based method with an AM-E network model has been presented and demonstrated with high-quality solutions. AM-E model is composed of a novel edge-enhanced dot-product attention layer via merging edge features into the information propagation, offering the network a stronger ability to describe the complex relationship among heterogeneous nodes. With the designed masking policy and reward function, the AM-E model has been trained and evaluated via the simulated environment. Results showed the AM-E is fast, robust and has better scalability compared with three baseline heuristic methods and a state-of-the-art deep learning model.

Chapter 4

A Model-based Energy-aware Route Planning Framework

This chapter introduces an integrated mission planning framework, namely EaPF. The framework aims to resolve the stochastic route optimization problem for drone delivery, with a focus on optimizing service qualities while considering possible time-window violations and battery depletion. Three main components of the EaPF will be presented, including a statistical energy predictive model based on an MDN, a DRL route optimizer, and a risk assessment module. Besides, an MBMS route construction strategy is also proposed to better accommodate the DRL planner within EaPF and further improve its solution quality and robustness of planning. Simulations are carried out to demonstrate the performance of the proposed EaPF in the context of drone deliveries.

4.1 Introduction

With the rapid development of drone technologies, using drones to transport goods has gained increasing attention. Online retailers such as Walmart, Google, and Amazon are exploring the potential of UAVs to enhance delivery time in last-mile delivery while being eco-friendly and cost-efficient [192, 193, 194]. Zipline, who has delivered over five million items globally, highlighted that the speed, reliability, and efficiencies have been brought by those aerial couriers to medical distribution with reduced expiration rates by 67%, reduced maternal mortality by 88%, and reduced CO2 emissions by 97% as compared to ground deliveries today [195, 196].

Despite benefits, delivering goods with drones comes with a very limited battery and load capacity when compared to conventional ground transportation. To compensate, multiple trips are made subject to a budget constraint, resulting in multi-trip VRPs [197]. However, frequently returning to the depot results in low efficiency of time and energy, and the serving range of delivery is still restricted. The deployment of recharging stations is thus motivated and investigated in [198, 199] on the subject of deciding the station placement to maximize the coverage demand with the least amount of infrastructure investment. Hybrid truck-drone systems are also proposed to utilize ground vehicles to refuel drones [200]. Considering recharging services in aerial logistics tends to optimize the overall travelling distance and quality of services. It forms a more complicated planning problem, where in-advance scheduling of station visits is desirable. Further, in unmanned aerial systems, energy risk control is critical for avoiding the potential loss of vehicles. It involves the process of pinpointing, evaluating, and prioritizing risks associated with uncertainties in drones' energy consumption. Overall, with the aim of reducing the travelling distance, improving the service quality, and controlling the energy risk, route optimization for drone delivery is expected to be conducted in an energy-aware manner.

While estimating the energy requirement plays an essential role in energy-aware route planning, it faces challenges in terms of prediction precision and planning efficiency. First, unlike ground vehicles, drones' energy consumption can be considerably affected by factors such as prevailing wind conditions, air density, payload, drone design, and airspeed [201]. Developing a precise energy prediction model is of utmost importance to prevent the loss of UAVs, and to avoid overly cautious and inefficient solutions resulting from overestimation of energy costs. Second, energy-aware planning involves a large number of iterations between the route planner and the drone energy prediction model, which is mostly established on the simulation of aerodynamics and power estimations. This causes a huge increase in computation amounts and a dramatic decrease in planning efficiency. Due to this reason, most route optimization models only incorporate energy consumption implicitly via a pre-defined limited drone duration or a linear function of payload weight [197, 202, 203]. This simplification may benefit computational tractability but does not reflect reality. To enable efficient and effective energy-aware planning, an efficient well-formed energy model is desirable for producing more reliable and robust solutions for route optimization.

This chapter investigates the stochastic route optimization problem for drone deliveries and proposes an efficient energy-aware planning scheme that accounts for

stochastic wind conditions and aims to enhance the robustness of route solutions. The contribution of this study is four-fold:

- 1) An energy-aware planning framework, namely EaPF, is proposed, which integrates a statistical energy predictive model, a route optimizer, and a risk assessment module. Adaptive to any route optimizer, this framework is demonstrated with a DRL planner herein and produces a more reliable solution for the energy-sensitive application.
- 2) A MDN model is developed for predicting drones' energy consumption with respect to wind distributions and load weights. This statistical model estimates a posterior distribution of the energy cost without the need for an erroneous and time-intensive sampling process.
- 3) A risk-sensitive criterion is established based on the statistical energy model, where the possibility of violating time windows and exceeding the energy budget constraint are both considered. The criterion is used to assess the quality of a route, facilitating solution production of higher robustness.
- 4) A novel MBMS route construction strategy is proposed. This strategy accommodates the DRL planner into the EaPF, enhancing the planner's robustness, predictability and reliability.

The rest of the chapter is organized as follows. Section 4.2 provides an overview of previous works with regard to energy-aware planning and simulation-based optimization. Section 4.3 gives a conceptual definition of green logistics systems, introduces the route optimization problem for the system, and provides its Markov modelling. This is followed by the proposed Energy-aware planning framework in Section 4.4. Section 4.5 and Section 4.6 respectively present the details about the implementation of the MDN energy predictive model and DRL route optimizer. The validations and discussion in terms of energy modelling and the EaPF are provided in Section ???. Section 4.9 concludes this study and outlines a brief plan for future research.

4.2 Related Works

4.2.1 Drone Energy-aware Planning

A variety of energy consumption models have been established to predict the energy demand for drones, either based on the dynamics of vehicles [204, 205, 197] or via fitting the practical experiment dataset. Langelaan *et al.* [206] investigate the flight power demands of rotary-wing vehicles, considering both hover and forward flight. Based on this study, Kirschstein *et al.* [207] propose an energy consumption model focusing on the drone delivery task constituted by four flight patterns. Some other energy models were established upon regression techniques and machine learning. After partitioning the flight of the UAV into segments of different movements, Prasetia *et al.* [208] establish energy regression models for these movements separately. Nguyen *et al.* [209] model the energy consumption as a black box and train it by comparing a set of predicted destinations and the truly reachable set to extend the range of delivery. Choudhry *et al.* [210] develop a deep energy consumption model with temporal convolutional networks, which is trained on a real-world dataset [211] and shows a 29% improvement in power predictions compared to the analytic energy model proposed in [204]. Nonetheless, most existing drone energy predictive models are deterministic, assumed to operate in a static and pre-known environment, failing to reflect the randomness in real-world scenarios.

On the basis of energy cost predictive models, studies have been conducted to take into account energy consumption in drones' planning, routing and scheduling [212]. On applications of data gathering, the energy minimization problem of multiple-drone systems is investigated in [213] and [214], where mixed-integer non-linear programming problems are formulated and solved based on the propulsion and communication energy model. To address the drone delivery problem under varying wind conditions, Sorbeliili *et al.* [49] define a specific variant of the VRP, namely the mission feasibility problem. In the mission feasibility problem, future energy consumption is continuously updated using the latest observation of wind control units and reflected by dynamic edge costs of the routing graph. Different from our work, the MFP focuses on finding feasible routes for a single delivery cycle, where the drone serves only one customer and returns to the depot. A multi-drone multi-delivery system dealing with winds, obstacle avoidance, and energy budget is investigated in [215]. Given a precise weather forecast, the time horizon is subdivided into weather time windows. A set of admissible submissions are created regarding each time win-

dow and arrive at the final mission plan via a depth-first search strategy. This work is then extended in [50] with regard to task clusters. In both works, wind conditions are assumed to be known and unchanged during each mission. However, the wind is by its nature uncertain and unpredictable. Especially when drones are operating in built-up urban areas or in the windy outdoors with complex terrain, it is always impossible to obtain accurate wind predictions.

Energy-risk assessment is an important component in energy-aware planning, which refers to the evaluation of the potential risk that a UAV may exceed its energy limitation. It is a relatively new research domain and stems from the difficulty in identifying and quantifying various factors that affect UAV energy consumption. Choudhry *et al.* [210] employ Conditional-Value-at-Risk [216] to quantify the risk by simulating the energy cost of a flight path through Monte Carlo forward simulation. Though the Conditional-Value-at-Risk provides a good assessment tool for the energy risk on a given trajectory, the Monte Carlo simulation is not efficient enough for risk-aware planning, as it requires a large number of iterations between the planner and the risk assessment module. Alternatively, Lau *et al.* [217] describe the travel time of edges as pre-known gamma distributions, and optimize a stochastic orienteering problem while considering the possibility of not completing the intended plan. In [217], a risk-aware criterion is proposed in addition to the reward-collecting objective, taking into account the probability of violating deadlines, which is then used in a local search method for solving the problem. Distributions of uncertainties in this model are assumed pre-known, which is not always true in reality.

To sum up, there is still a lack of research on energy risk assessment and energy-aware planning for UAVs, and the main challenge lies in the prediction of flight energy costs.

4.2.2 Simulation-based Optimization for Routing Problem

Simulation-based optimization extends deterministic techniques to deal with stochastic optimization problems. One typical application, referred to as the term *sim-heuristics*, is combined with metaheuristic methods. By integrating simulations into an existing metaheuristic framework, sim-heuristic allows the handling of uncertainties. Sim-heuristics also facilitate the introduction of risks and reliability analysis/assessment of solutions in stochastic combinatorial problems. Juan *et al.* [218] provides a general scheme of sim-heuristics for solving stochastic combinatorial optimization problems. For its specific application on routing problems, Mazzuco *et*

al. [219] investigates the problem of daily good deliveries and presents the first outcomes of a simulation-based optimization approach for vehicle routing and transport scheduling. In terms of the stochastic character of real-world VRPs, Mayer *et al.* [220] offer a rich VRP simulation model based on the DES workflow. The simulation model is then used to evaluate the solution candidates for metaheuristics.

In addition to combinations with metaheuristics, the combined use of system simulation and machine learning was also proposed in a few studies, aiming to enhance the quality of solutions generated via intelligent agents. A trial model of applying RL in the AnyLogic model is investigated in [221]. They demonstrate how to implement RL in the AnyLogic simulation model using the Pathmind Library. The combination of DES and deep RL has also been studied on large logistics networks [222], for which the authors first employ meta-heuristic approaches and then replace them with a reinforcement learning agent to find the most promising action sets.

4.3 Energy-constrained Drone Delivery

This section offers a generic concept of drone delivery systems including its essential system components and the definition of its route optimization problems. The route optimization problem is further modelled as a Markov decision process, providing the preliminary of the approach proposed in the following sections.

4.3.1 Route Optimization for Drone Delivery

System Components

Referring to the architecture of the green logistics system in [223], we list the five basic components for drone delivery systems, namely the transportation network, vehicles, logistics requests, renewable generations, and depots.

- *Aerial Transportation Network:* We define the transportation network on an undirected graph $\mathcal{G} = \langle V, E \rangle$. Each vertex $v \in V$ stands for a point of interest, which could be a task request, a recharging facility or the depot. Each edge $e \in E$ represents the path segment linking a pair of vertices. Vertices are connected with the corresponding logistics components and each edge is associated with

a series of waypoints, including the take-off, landing and predefined mid-way route points.

- *Autonomous Aerial Vehicles*: We denote aerial vehicles in the system as $k \in \mathcal{K}$. Each of them is attached with an independent battery capacity B_k and SoC c_k . Drones are normally sent out from the depot with fully recharged batteries and are expected to get recharged before the battery is used up. It is assumed that the logistics capacity of drones is 1, i.e., it only processes one request at a time.
- *Logistics Requests*: Customer requests are denoted as $q \in \mathcal{Q}$, which consist of pickup requests $q_p \in \mathcal{Q}_p$ and delivery requests $q_d \in \mathcal{Q}_d$. Each pair of pickup and delivery requests is bound together and constrained by a precedence relationship. They share the same load weight w_q , task value r_q , appearance time a_q , and execution deadline d_q .
- *Renewable Generation*: Renewable facilities $g \in \mathcal{G}$ are deployed to extend the range of services and improve environmental-friendliness. The batteries of drones get recharged or swapped in these locations. If it is recharged, the recharging rate is denoted as r_k . Additionally, we consider at most Ω_g drones could be charged at point g at the same time.
- *Depots*: Depots $d \in \mathcal{D}$ refer to those places where drones take off and are recycled at the end of the route. In the concerned drone delivery system, the depots also serve to charge drones.

Mission planning for drone delivery systems involves two subproblems, i.e., the allocation of logistics requests among the UAV fleet, and the generation of an optimal route for each team member. In this study, we mainly focus on the latter problem, which can be classified as a dynamic and stochastic green vehicle routing problem.

Route Optimization

In logistics systems, optimizing the delivery flow indicates maximizing the rewards collected from requests while minimizing the loss if failing to arrive at one task within its time window or exceeding the energy budget. For easier expression, we denote the status of the vehicle at vertex j by visiting time t_j and SoC c_j , which respectively indicate the time and battery level of the drone when it arrives at vertex j .

In drone delivery systems, we consider a practical objective from four aspects: 1) maximizing the sum of collected rewards; 2) minimizing the penalty of missing delivery time windows; 3) minimizing the penalty of battery depletion; and 4) minimizing the overall energy cost. Regarding a route $\mathcal{P} = [v_1, v_2, \dots, v_n]$, using the above definitions, the objective function of the drone delivery is formally given by

$$\max f(\mathcal{P}) = \sum_{j=1}^n r(v_j) - p_e(\mathcal{P}) - \sum_{j=1}^{n-1} e(v_j, v_{j+1}) \quad (4.1)$$

where $r(v_j)$ is the task-oriented reward function, which returns a positive value when the execution deadline is met, otherwise returns a negative value $-C_{pen_j}^t$, $e(v_j, v_{j+1})$ indicates the energy cost flying from v_j to v_{j+1} , and $p_e(\mathcal{P})$ is the penalty function that a constant penalty variable C_{pen}^e will be imposed to the system if an energy depletion happens during any point of the delivery process. The formal definitions of $r(v_j)$ and $p_e(\mathcal{P})$ are given as

$$r(v_j) = \begin{cases} 0 & \text{if } v_j \notin \mathcal{Q}_d \\ r_j & \text{if } v_j \in \mathcal{Q}_d \text{ and } t_j \leq d_j \\ -C_{pen_j}^t & \text{if } v_j \in \mathcal{Q}_d \text{ and } t_j > d_j \end{cases} \quad (4.2)$$

$$p_e(\mathcal{P}) = \begin{cases} 0 & \text{if } c_j > 0, \forall v_j \in \mathcal{P} \\ C_{pen}^e & \text{otherwise} \end{cases} \quad (4.3)$$

It is noted that adding the penalty of energy depletion is to avoid adverse objective value when an algorithm performs well at searching the optimal path but is bad at energy management, which is not how the energy constraint is controlled. The energy constraint is guaranteed in other ways, i.e., energy usage margin and the action masking in the DRL planner. This will be expanded later in this chapter.

4.3.2 Markov Modelling for Drone Delivery

The MDP model of the drone delivery problem is defined as the tuple $\langle s_i, a_i, r_i, p_i \rangle$.

- $s_i \in \mathcal{S}$ is the system state, constituted by the drone state $s^D = [x_d, y_d, t, c]$ indicating the drone's coordination, system time and state of the charge, and the mission state $s^M = \{[(x_j, y_j), w_j, r_j, (a_j, d_j)], j \in \mathcal{T}\}$ representing the co-

ordination, weight, reward, and time window of the customer request.

- $a_i \in \mathcal{A}$ is the agent action, indicating the next movement of the drone, represented by the corresponding vertex $a = v_j, j \in \mathcal{T}$. The whole path is generated by successively selecting actions, denoted as $\mathcal{P} = [a_0, a_1, a_2, \dots]$. An action could be the depot, a task or a recharge station. With the purpose of exploration efficiency, actions that tend to exceed the energy budget will be masked by a masking policy.
- $r_i = r(s_i, a_i, s'_i)$ is the reward obtained from the environment after the drone executes the selected action a_i under state s_i . The calculation of reward follows the objective function defined in Section ??.
- $p_i = p(s'_i | s_i, a_i)$ represents the transition probability that the current state s_i is converted to the next states s'_i by taking action a_i . When the drone travels to the vertex j , the drone state and mission state get updated according to the type of the node
 - *i*) if the targeted vertex is a task, the drone moves to the location of the destination, the state of the charge is reduced by the edge cost, and the time is evolved with the flight duration.
 - *ii*) if the destination is a recharge station or the depot, the mission state remains unchanged, while the location of the drone moves to the targeted vertex, the energy level of the drone returns to full, and the time clock moves forward counting in both the flight time and recharge time.

The goal for solving this MDP is to obtain a policy π^* that maximizes the expected accumulative reward within a finite time range:

$$\pi^* = \underset{\pi}{\operatorname{argmin}} E \left[\sum_{\tau=1}^{\infty} r_{\tau} | \pi_{\tau} \right] \quad (4.4)$$

4.4 Energy-Aware Planning Framework

To resolve the described delivery problem, we propose an EaPF as depicted in Figure 4.1. The main goal of EaPF is to produce robust routes considering the possible violation of task deadlines and manage energy risks in the face of poorly known uncertainties. It mainly consists of three modules: an energy cost model, a route optimizer and a risk assessment block.

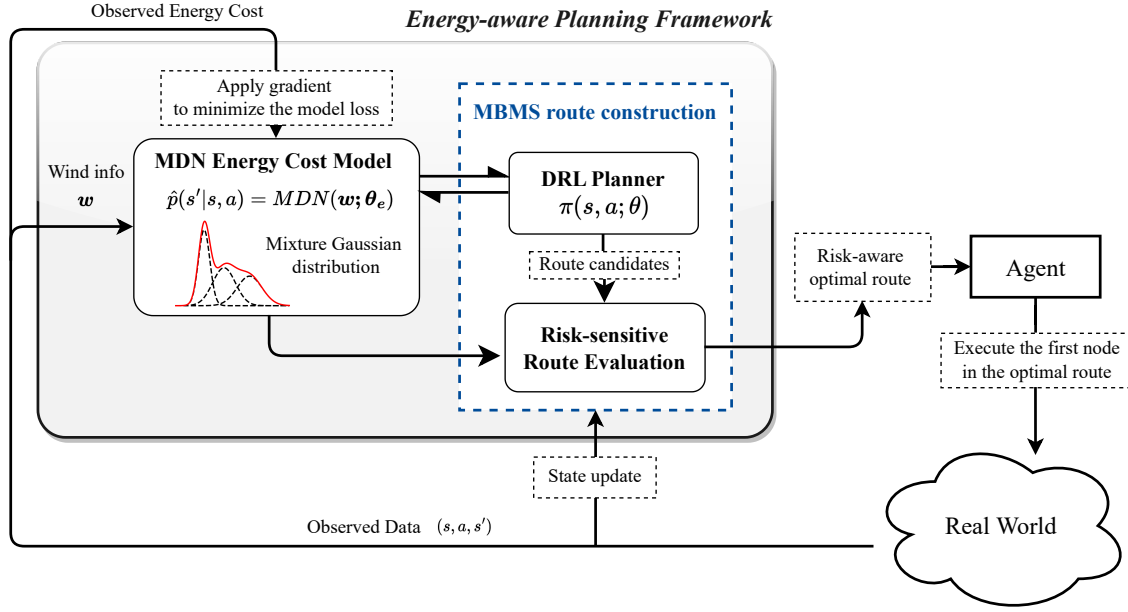


Figure 4.1: Overview of the Energy-aware Planning Framework. It integrates an MDN energy cost model, a DRL route optimizer, and a risk assessment block. The DRL optimizer is adapted with an MBMS route construction strategy with the help of the energy model and the risk assessment.

First, a routing environment is imitated by using an energy cost model built on an MDN. The MDN of the energy cost is pre-trained by historical data and refined on-line using empirical observations to reduce the discrepancies between the simulator and real-world transition probabilities. Mixture Gaussian distributions of energy and time along a given route are then calculated by integrating the MDNs of path segments. The distributions are then used to estimate the risk of task failure and battery depletion, constituting the risk assessment function of the EaPF. For route planning, a pre-trained DRL planner operates as the route optimizer which constructs the route by appending the next action to the existing partial path based on the evolved observation from the outer environment. To enhance its robustness, reliability and predictability, the MBMS strategy is incorporated with the DRL planner, where route candidates are constructed, evaluated and executed in a Model Predict Control and Simulation-based Optimization way.

4.4.1 MBMS Route Construction Strategy

In terms of the route optimizer, we implement a DRL method for its natural ability to handle uncertainties and embed it with an MBMS route construction strategy as

shown in Figure 4.2. Instead of choosing the next action following the greatest probability from the output of the policy network (i.e., the *greedy strategy*), the MBMS strategy enables the DRL planner to perform in a *sim-heuristic* way. It enhances the performance of the original DRL planner to deal with stochastic optimization problems and also facilitates the introduction of risks and reliability through assessing solutions via simulators.

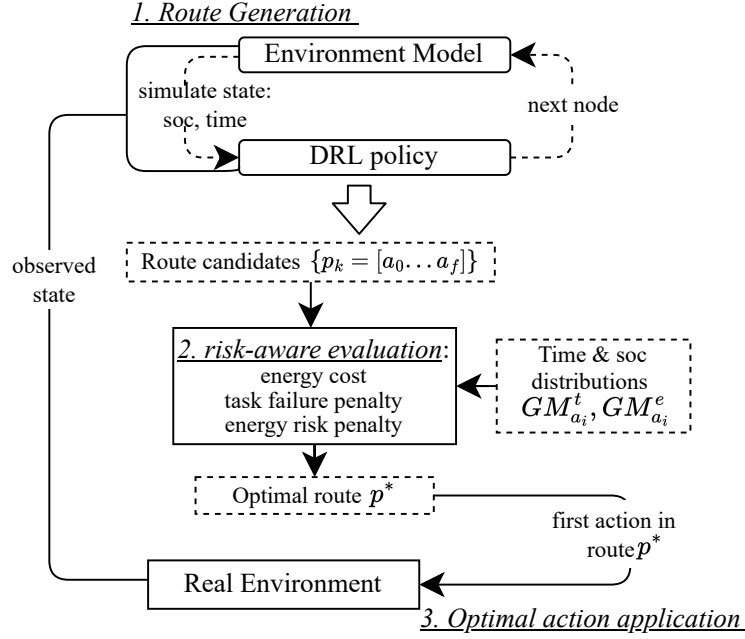


Figure 4.2: Three steps for the Model-based Multi-Sampling Route Construction Strategy.

The proposed MBMS strategy is implemented in three steps: route generation, risk-aware evaluation, and optimal action application.

- 1) *Route generation:* Given the initial state observed from the outer environment, the DRL planner samples the next action by the sampling number N based on the probability vector yielded by the policy network. Instead of executing the action in the real world, the agent applies these actions to the environment model and receives the estimated state from the model. The process is repeated till a batch of routes is fully constructed.
- 2) *Risk-aware evaluation:* Time-window exceeding risks and energy depletion risks are both assessed along the generated routes via applying energy MDN models for each path segment and integrating the distributions prior to each task or recharging way-point. A risk-aware criterion is built on account of these risks and used to evaluate all the route candidates.

- 3) *Optimal action application:* Only the first action in the best route sequence will be taken in the real environment. The planning problem is solved successively with the latest observation from the real world.

The incorporation of the MBMS strategy is expected to enhance the robustness of the DRL planner by sampling a batch of route trajectories and filtering the candidates using risk-aware criteria. In previous studies, researchers have proposed various attempts to improve the robustness of the DRL method, with most of the focus on modifying the structure of policy neural networks. For example, LSTM has been used to capture the history of state transitions while making decisions. While these attempts have achieved considerable progress, the approximate nature of neural networks limits their efficacy, and sometimes a sophisticated design is less beneficial than simply adding a sample decoding scheme, as demonstrated in [5]. However, little attention has been paid to the solution construction phase after the policy model is trained, and difficulties arise in evaluating the quality of solutions in stochastic cases, which hinders further solution searching. The MBMS strategy is proposed to address these challenges.

4.5 Energy Modelling and Risk-Aware Criterion

4.5.1 Energy Prediction Model via MDN

We use an MDN to estimate the energy cost of drones with respect to the wind and the payload weight. Researchers have adopted regression techniques to predict the energy cost, the goal of which is to find a single-value prediction with maximum likelihood or a Gaussian conditional distribution $p(e|x)$, where e is the energy cost and x indicates the influential variables. However, under varying wind conditions, the drone energy cost for a given path is distributed across a large range and is shown to be non-Gaussian [36]. To that end, we adopt the MDN in this study for a more general formulation and more precise prediction.

MDN model

The MDN approximates conditional density functions by mixing several component densities with mixing coefficients. Both densities and coefficients are flexible

functions of the input vector x . Here components are chosen as GMs:

$$p(t|x) = \sum_{k=1}^K \pi_k(x) \mathcal{N}(t|\mu_k(x), \sigma_k^2(x)) \quad (4.5)$$

where the decision variables x on drone energy consumption is selected as $x = [\mu_{v_w}, \sigma_{v_w}^2, \mu_{\theta_w}, \sigma_{\theta_w}^2, w]$, containing the mean and variance of both wind speed v_w and direction θ_w , and payload weight w .

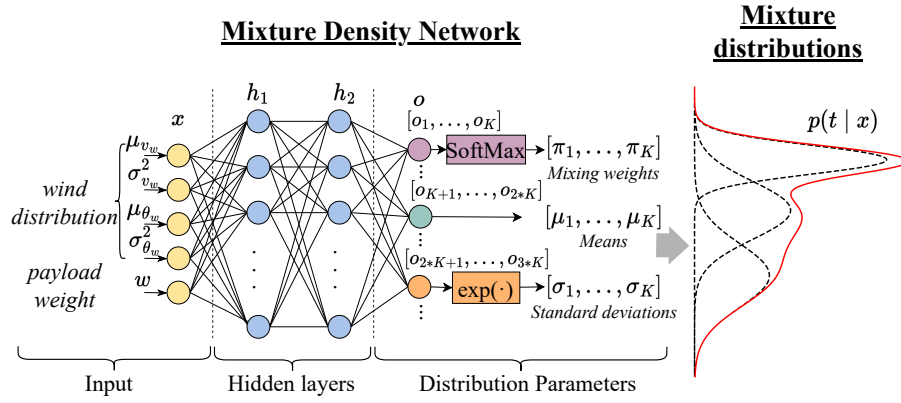


Figure 4.3: The mixture density network for predicting the distribution of the drone's energy cost, given the input including the wind distributions and the payload weight.

Parameters in the above Gaussian mixtures are governed by the outputs of a conventional neural network that takes x as its input. If there are K components in the mixture model, the network will have $3 \cdot K$ outputs, representing mixing coefficients $\pi_k(x)$, means $\mu_k(x)$ and variances $\sigma_k^2(x)$ respectively. Notably, since the sum of mixing coefficients must be 1, an extra SoftMax is added to coefficient outputs, and an exponential operation is added to the variance outputs to satisfy the positivity of variances.

Parameter Update

Ahead of the mission, the MDN network forms the prior distribution of energy cost $p(t|\theta_i)$, which is parameterized by θ_i and trained using historical and simulated data. Once practical flights launch, the network is then modified onboard with the latest empirical results (x_i, y_i) , producing a posterior distribution $p(t|\theta_i, x_i, y_i)$.

In terms of the parameter update, the trainable parameters θ of this density network, including weights and biases in the neural network, are set by maximizing

the likelihood, or equivalently by minimizing a loss function defined as the negative logarithm of the likelihood. The loss function takes the form

$$L(\theta) = - \sum_{n=1}^N \ln \left\{ \sum_{k=1}^K \pi_k(x_n, \theta) \mathcal{N}(t_n | \mu_k(x_n, \theta), \sigma_k^2(x_n, \theta)) \right\} \quad (4.6)$$

To minimize the loss function, we calculate the derivatives of the loss $L(\theta)$ with respect to the parameters θ and update the network via the standard back-propagation procedure. Unlike the fixed energy prediction model, the MDN is updated continuously, beneficial to long-term performance in real-world applications

4.5.2 Risk-Aware Criterion

Expectation of Objectives

The risk-aware criterion takes the expected form of the deterministic objective function defined in Section 3.B. Total energy cost is directly governed by the GM model of drone energy consumption. The reward function and task lateness penalty are decided by arriving times, which implicitly depend on energy cost as it affects charging time if ground speed is assumed to be constant. Accordingly, the expected objective function with respect to the GM model of energy cost is formulated as

$$\max E(f(\mathcal{P} | \theta)) = \sum_{j=1}^n E(r(v_j) | \theta) - \sum_{j=1}^{n-1} E(e(v_j, v_{j+1}) | \theta) \quad (4.7)$$

where the expected objective components are given by

$$E(r(v_j) | \theta) = P(t_j \leq d_j | \theta) r_j - P(t_j > d_j | \theta) C_{pen}^t \quad (4.8)$$

$$E(e(v_j, v_{j+1}) | \theta) = \mu(e(v_j, v_{j+1}) | \theta) \quad (4.9)$$

where $P(\bullet | \theta)$ denotes the probability defined by uncertainty parameters θ .

Regarding the objective of not depleting the battery, we first build up a risk profile for a given route. Then, a chance-constrained route filter is applied to the candidate route to guarantee that the energy constraint is satisfied within a certain risk threshold. Precisely, a route candidate $p = [v_1, v_2, \dots, v_k]$ can only be denoted as

valid if it satisfies the following change-constrained metric:

$$\mathcal{P}^* = \{p : P(c(v_j) < B_r \mid \theta) \geq 1 - \epsilon, v_j \in p\} \quad (4.10)$$

where B_r is the drone's remaining budget. If the agent chooses a route from $\mathcal{P}^*$, it can be guaranteed to finish the route within the allowable risk threshold specified by ϵ .

Joint Energy Distribution

The probability distribution of energy cost for flying over a mission route is calculated by integrating all the probability density functions of its path segments. The MDN energy cost predictions are applied on each edge segment rather than overall route solutions for the purpose of flexibility and computational efficiency. The path partition is under the assumption of independent energy costs. As a simple illustration, suppose there is a route from waypoint A to waypoint C passing through waypoint B , the probability function of the route is then given by

$$e_{ac} = e_{ab} + e_{bc} \sim \sum_i \sum_j \pi_i \phi_j \mathcal{N}(\mu_i + \mu_j, \sigma_i^2 + \sigma_j^2) \quad (4.11)$$

when energy costs for both path segments follow GM distributions:

$$e_{ab} \sim \sum_{i=1}^n \pi_i \mathcal{N}(\mu_i, \sigma_i^2), \quad e_{bc} \sim \sum_{j=1}^m \phi_j \mathcal{N}(\mu_j, \sigma_j^2) \quad (4.12)$$

Mixture Reduction

We employ a mixture weight Pruning technique to retain the number of GM components within a reasonable range for the energy cost MDN model. The component number of the energy prediction model increases exponentially when integrating GMs of all path segments. Thus, we resort to mixture reduction techniques, which attempt to reduce the number of components in a mixture through emerging or pruning. Here we primarily employ the emerging operation, which is inspired by [224].

The emerging operation replaces a pair of components in a mixture density with a single component of the same type. The KL divergence is used as the criterion for selecting the emerging pair to minimize the divergence between the emerged and

original components. Formally, the KL divergence criterion $B(i, j)$ is calculated as

$$B(i, j) = w_i D_{KL} \left(p(x; \mu_i, \sigma_i^2) \| p(x; \mu_{ij}, \sigma_{ij}^2) \right) \quad (4.13)$$

$$+ w_j D_{KL} \left(p(x; \mu_j, \sigma_j^2) \| p(x; \mu_{ij}, \sigma_{ij}^2) \right) \quad (4.14)$$

where merged parameters are obtained by moment matching:

$$\mu_{ij} = \frac{w_i \mu_i + w_j \mu_j}{w_i + w_j} \quad (4.15)$$

$$\sigma_{ij}^2 = \frac{(\sigma_i^2 + \mu_i^2)w_i + (\sigma_j^2 + \mu_j^2)w_j}{w_i + w_j} - \mu_{ij}^2 \quad (4.16)$$

Merging operations are implemented in a greedy manner. The pair of components with the minimum global KL divergence is recursively selected to be merged until the target component number is reached.

4.6 DRL Route Optimizer

4.6.1 Policy Network and Masking Scheme

We use the DRL approach to address the formulated MDPs by adapting the AM-E neural network model, proposed in Section 3.4.1, to map the best policies.

With the purpose of improving exploration efficiency and ensuring the feasibility of solutions, the set of actions is masked by a designed masking scheme to exclude infeasible routes. According to the battery capacity constraint and the pickup-delivery precedence constraint, an action $a = v_j$ is labelled as valid only if all of the conditions listed in Section 3.4.2 are satisfied.

4.6.2 Training of the Planning Network

The policy network is trained using the REINFORCE algorithm with baselines, shown as Algorithm 4.1. A Critic baseline is chosen to reduce the variance of gradients. In detail, at the beginning of each episode, one batch of new missions is stochastically generated. Routes are sequentially sampled according to the probability output from the policy network while rewards are collected from the environ-

ment. Meanwhile, we calculated the expected reward $v(s_0^b)$ via the critic baseline network, which is used to calculate the advantage component of the gradients. Back-propagation is then adopted to update the policy network and the critic network for the critic baseline.

Algorithm 4.1 Training of the route planning network

Initialize: policy parameters θ , critic baseline parameter θ_v

Input: number of epochs N , batch size B , learning rate α

```

for each training  $epoch = 1, 2, \dots, N$  do
  generate instances of batch size  $B$ ;
  for each instance  $b = 1, 2, \dots, B$  do
    \*run in parallel* \
    sample trajectory according to policy  $\pi_\theta$ ;
    receive the accumulative reward of the trajectory  $R^b$ ;
    estimate the value of initial state using the critic  $v(s_0^b)$ ;
  end for
  estimate the mean Policy Gradient on the batch of trajectories:
   $d\theta = \frac{1}{B} \sum_{b=1}^B (R^b - v(s_0^b)) \nabla_\theta \log \pi_\theta(\pi_b | s_0)$ 
  estimate the mean gradient of the mean squared error of the critic on the batch
  of trajectories:
   $d\theta_v = \frac{1}{B} \sum_{b=1}^B \partial (R^b - v(s_0^b))^2 / \partial \theta_v$ 
  update policy parameters  $\theta = \theta + \alpha d\theta$ 
  update critic parameters  $\theta_v = \theta_v + \alpha d\theta_v$ 
end for

```

4.7 Validation of the MDN Energy Model

A power consumption model from [204] is used to obtain the reference energy cost for training and validating the proposed MDN energy model. Parameters in the baseline power model are identified via an energy usage dataset collected from 100-hour real-world flights [211]. Drone's parameters are shown in Table 4.1.

Table 4.1: Drone specifications

Parameter	Value
weight (including battery)	$3.71kg$
airspeed command	$10m/s$
cruise altitude	$20m$
battery capacity	$369.36kJ$

The process of dataset generation is depicted in Figure 4.4. Inputs to the training and validation datasets consist of wind parameters and payload weights, which are randomly generated and cover the full range of scenarios for a feasible UAV flight. To improve the training stability, the input data and output data are both pre-processed by feature-wise normalization. We apply the Adam optimizer to train the network with a learning rate of 0.001. The component number of the Gaussian Mixture model is set as $K = 5$. Key parameters of the MDN model and datasets are summarized in Table 4.2.

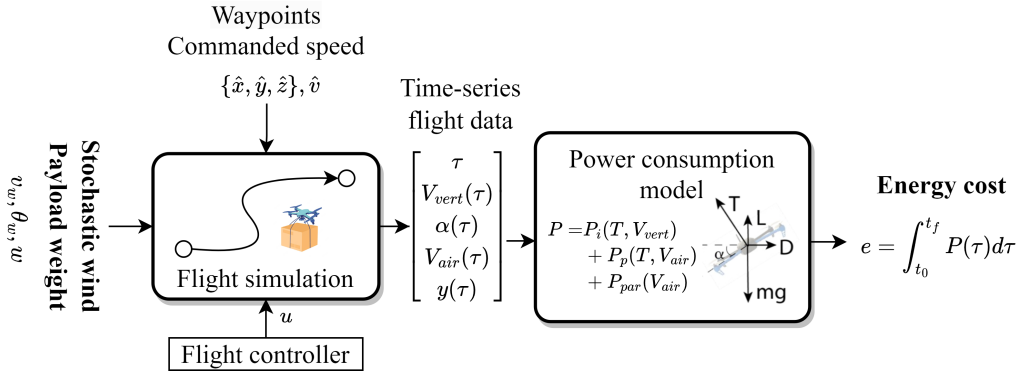


Figure 4.4: Energy consumption dataset generation.

After training for 100 epochs, the average likelihood on the validation dataset between the MDN model and the Monte Carlo simulation arrives at 8.0159. The likelihood function is defined by

$$\text{Likelihood}(\text{MDN}(\theta) \mid x_n) = \sum_{k=1}^K \pi_k(x_n, \theta) \mathcal{N}(t_n \mid \mu_k(x_n, \theta), \sigma_k^2(x_n, \theta)) \quad (4.17)$$

Table 4.3: Running time for the MDN energy predict model and Monte Carlo simulations

Method	Running time
MDN energy cost model	0.026''
1000 Monte Carlo simulation	8'45''

The predicted distributions of energy costs via MDN for four typical flight cases are depicted in Figure 4.5.

Table 4.2: Key parameters for MDN model and dataset. $\mathcal{U}(a, b)$ denotes a uniform distribution on interval $[a, b]$.

Parameter notation	Value or distribution
Wind speed mean (m/s)	$\mu_{v_w} \sim \mathcal{U}(0, 10)$
Wind speed std (m/s)	$\sigma_{v_w}^2 \sim \mathcal{U}(0, 5)$
Wind direction mean (deg)	$\mu_{\theta_w} \sim \mathcal{U}(0, 360)$
Wind direction std (deg)	$\sigma_{\theta_w}^2 \sim \mathcal{U}(0, 60)$
payload weight (kg)	$w \sim \mathcal{U}(0, 1)$
Training/Validation	80%/20%
The size of dataset	10000
Neural network architecture	Dense
Input number of network	5
Output number of network	15
Hidden layer number	2
Neuron of hidden layer	256
Learning rate	0.001
Maximum training epochs	100

Table 4.3 compares the average computing time of the MDN model and the Monte Carlo simulation. The results show that to predict the distribution of the energy consumption considering winds, the MDN energy cost model only requires 10^{-4} running time compared to Monte Carlo simulations.

In summary, the simulation outcomes demonstrate that the MDN model is proficient at accurately estimating the distribution of drone energy cost with respect to wind conditions and payload weight. Furthermore, it significantly decreases the computational time required for optimizing flight routes.

4.8 Case Study on Drone Deliveries

To test the performance of the proposed EaPF, we implement it in the context of drone delivery problems, which have been cast as stochastic greedy vehicle rout-

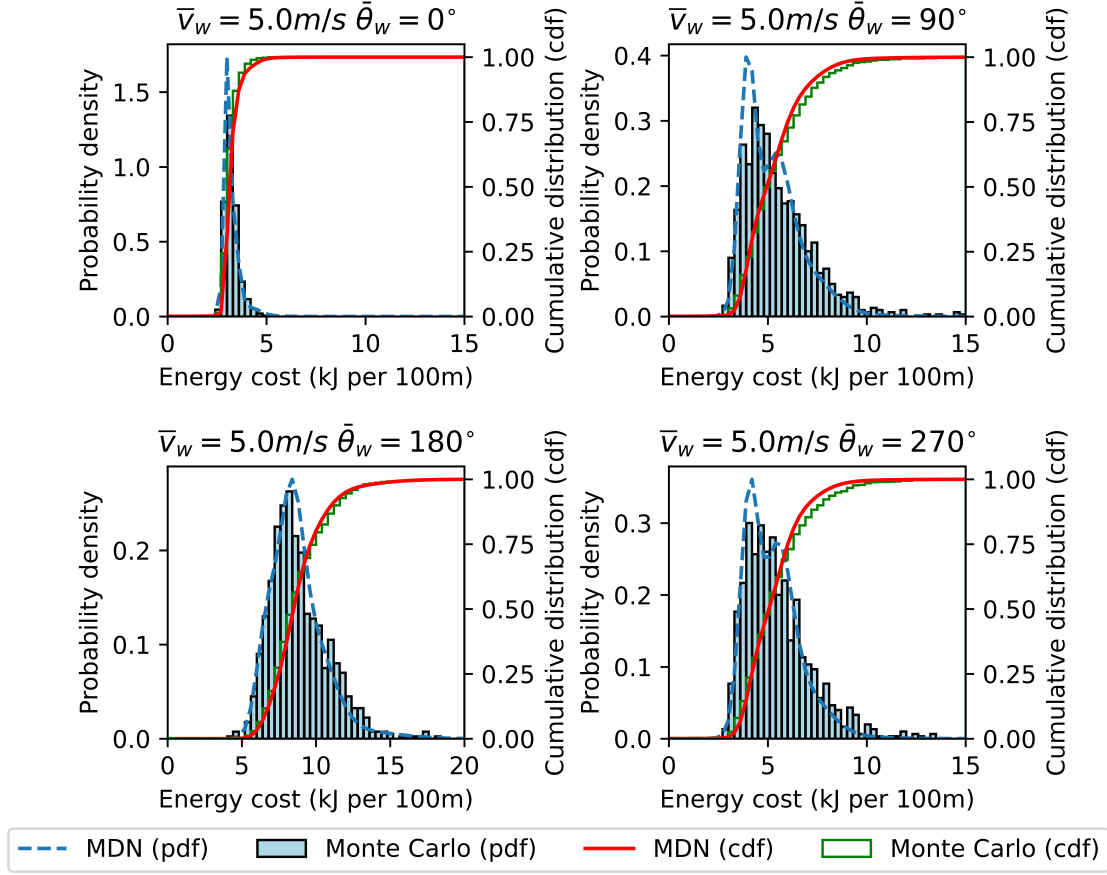


Figure 4.5: Energy prediction via MDN for four typical cases. Results obtained by MDN and Monte Carlo simulations are plotted in the form of probability densities functions and cumulative densities respectively. Four cases are simulated in the presence of winds from different directions: $0deg$ (tailwind), $90deg$, $180deg$ (headwind), $270deg$. The wind velocity v_w and direction θ_w follow: $v_w \sim \mathcal{N}(5m/s, 1m/s)$, $\theta_w \sim \mathcal{N}(\bar{\theta}_w, 30deg)$

ing problems with time windows (S-GVRP-TW) operated on the unmanned aerial platform.

4.8.1 Experimental Setup

Delivery Mission

To train and validate the performance of the EaPF, a simulated environment is set up to initialize the system and evolve its state updates. As stated in Section 4.3.1, the drone delivery problem considers the scenarios where a list of customer requests j , consisting of the pickup location (x_j^p, y_j^p) and the delivery location (x_j^d, y_j^d)

Table 4.4: Parameter settings for drone delivery instances. $\mathcal{U}(a, b)$ denotes a uniform distribution on interval $[a, b]$; $\mathcal{U}\{x_1, \dots, x_n\}$ denotes a discrete uniform distribution on finite set $\{x_1, \dots, x_n\}$.

Parameter notation	Value or distribution
Number of customers	$N_q = 10$
Number of recharge stations	$N_g = 3$
Depot location (km)	$x^0, y^0 \sim \mathcal{U}(0, 1)$
Recharge station location (km)	$x^g, y^g \sim \mathcal{U}(0, 1)$
Customer pickup location (km)	$x_j^p, y_j^p \sim \mathcal{U}(0, 1)$
Customer delivery location (km)	$x_j^p, y_j^p \sim \mathcal{U}(0, 1)$
Customer load weight (kg)	$w_q \sim \mathcal{U}(0.0, 0.25, 0.5, 0.75, 1.0)$
Drone airspeed command (m/s)	$V = 10$
Drone battery capacity (kJ)	$B = 369.36$
Planning horizon (min)	$H = 120$
Customer due time (min)	$d_j \sim \mathcal{U}(t_{min}^j, H)$ where $t_{min}^j = \text{dist}(0, j)/V$

are pending for the drone to execute. Due to the battery limitation B , recharging facilities (x^g, y^g) are also deployed for drones to get recharged and maintain their working status. The drone is sent from the depot (x^0, y^0) to serve the customers and finally returns to the depot after completing all the tasks. Following the stated delivery scenario, delivery instances are generated using the parameter settings in Table 4.4.

Stochastic winds are also taken into account in the training environment. The wind speed is randomly generated via a Weibull distribution, which is widely used to describe wind variation [225]. The scale parameter $\bar{w}$ and shape parameter w_k in wind Weibull distributions respectively indicate the mean value of wind speed and distribution shape. The direction of the constant wind is assumed to follow a uniform distribution from $\theta_w = 0^\circ$ to 360° . Details of parameters related to the wind modelling are listed in Table 4.5.

Table 4.5: Parameter settings for wind modelling.

Parameter notation	Value or distribution
Wind speed (m/s)	$v_w = \mathcal{N}(\mu_{v_w}, \sigma_{v_w}^2)$
Wind speed mean (m/s)	$\mu_{v_w} \sim Weibull(\overline{v_w}, 2), \overline{v_w} = 5$
Wind speed std (m/s)	$\sigma_{v_w}^2 = 0.2\overline{v_w}$
Wind direction	$\theta_w = \mathcal{N}(\mu_{\theta_w}, \sigma_{\theta_w}^2)$
Wind direction mean	$\mu_{\theta_w} \sim \mathcal{U}(0^\circ, 360^\circ)$
Wind direction std	$\sigma_{\theta_w}^2 = 15^\circ$

DRL Planner Setup

The neural network structure and training hyper-parameters of the DRL route planner are set as follows:

- *Structure:* The AM-E network structure is as shown in Figure 3.8. The node embedding, edge embedding, and context embedding are all one-layer element-wise linear projections with 128 dimensions. The encoder is constituted by $N = 3$ sequential attention modules. The multi-head attention layer consists of $M = 8$ heads, in which key vectors and value vectors are of dimensions $d_k = d_v = 16$. In the feedforward layer, the node features are passed through node-wise projections of one hidden sublayer with the Rectified Linear Unit (ReLU) activation and 512 hidden units.
- *Training parameters:* We adopt the Adam optimizer to train the network with a constant learning rate of $\alpha = 10^{-4}$. We run training for 100 epochs. In one epoch, we process 1.28 million instances in 2500 iterations with a batch size of 512.

Energy Cost Calculation

Energy cost is a very important component in a drone’s energy-aware planning and validation. Three energy cost models are used in the simulations.

- *Baseline energy model:* The baseline energy model is adapted from [204] and employed to validate all the solutions in the comparison study, as it considers

the dynamics of drones and stochastic wind field, featuring a high fidelity.

- *Steady energy model:* The steady energy model is a simplified version of the baseline model. It assumes the drone is in a steady state most of the flight, and the external forces are in static equilibrium. This model is not as accurate as the baseline model but more efficient because it avoids the dynamic model and renders a static map from commanded airspeed and trust to energy. This model is used in the heuristic planning method.
- *MDN model:* The MDN energy cost model is proposed in this work and performs as the key component in EaPF. It interacts with the DRL planner generating route candidates and is used to build the risk-aware criterion.

4.8.2 Simulation Results

Four different energy-aware planning methods are implemented and compared for solving the above drone delivery planning problems.

1. *Energy-aware heuristics:* Google’s Or-tools is employed to solve the energy-aware drone delivery problem through an energy cost matrix of flight edges. The cost matrix is built up using an analytical energy cost model from [204] under the assumption of the accurate forecast of wind speed and direction. Google’s Or-tools is backed by a CP-SAT solver and is regarded as the most advanced commercial solver for combinatorial optimization.
2. *Stand-alone DRL planner:* The DRL planner is trained by interacting with the simulated environment via the REINFORCE algorithm, as presented in Section 4.6. The stand-alone DRL planner observes the energy cost from the validation environment each time the drone finishes a task, and makes decisions by choosing the next node with the highest probability.
3. *DRL planner (sample128):* The planner retains the same as the stand-alone DRL planner, but with the addition of a sampling decoding strategy to construct the next action. Specifically, it utilizes an environmental model to generate a set of trajectory samples of size 128 and choose the best one from it.
4. *DRL planner (EaPF):* In EaPF, the DRL planner constructs the route via a new proposed MBMS strategy, as presented in Section 4.4.1. The MBMS

strategy makes use of MDN energy models to construct and evaluate the route candidates.

We consider three series of validation scenarios by varying three variables: 1) the margin of safety c , 2) wind speed variance $\sigma_{v_w}^2$, and 3) wind direction variance $\sigma_{\theta_w}^2$. The margin of safety refers to a common strategy in practice, which is to reduce the available battery capacity used in planning to avoid battery depletion. We vary margins of safety among $c = 0\% \sim 30\%$. Large safety margins indicate conservative planning with respect to energy depletion risks. When c decreases, the planned results tend to be risky. Wind speed variances $\sigma_{v_w}^2$, and wind direction variances $\sigma_{\theta_w}^2$ indicate different level of uncertainties. These two dimensions are set to test the robustness of methods in the presence of varying wind conditions.

Results of all three simulation series are depicted in Figure 4.6. Four plots in the first row respectively depict four performance metrics with respect to varying battery safety margins. As seen from them, the margin value has a significant impact on the performance of *or-tools(e)* and *DRL (sample128)*. Planning with 30% margins, both methods produce much safer routes in terms of lower energy depletion risk, leading to an increase of overall performance with lower cost. Meanwhile, as the introduction of battery margin imposes a hard boundary to the amount of energy available, it sacrifices the capability of drones regarding task accomplishment for all methods.

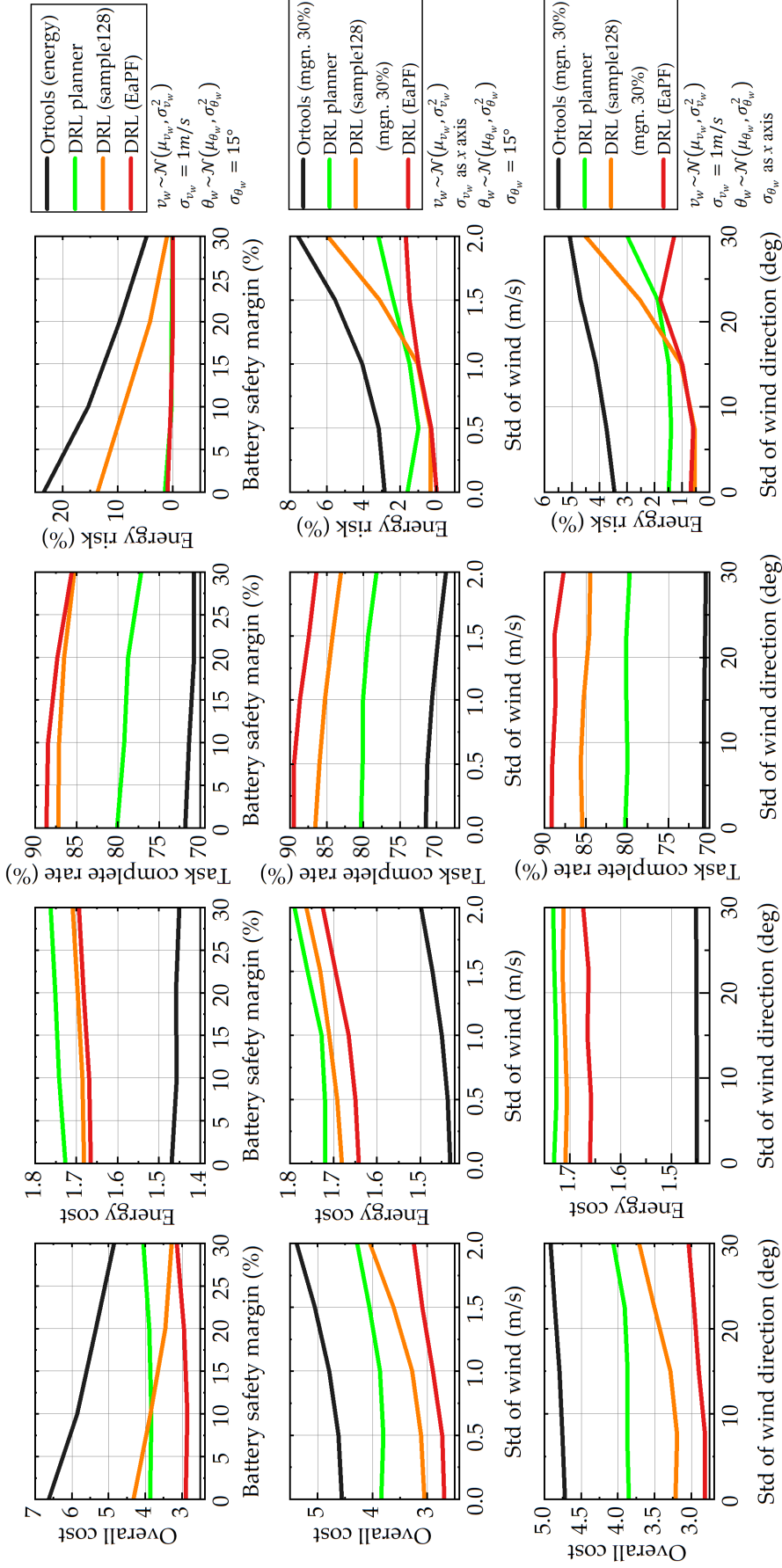


Figure 4.6: Simulation results in terms of overall cost, energy cost, task fail rate, and energy depletion rate, for solving drone delivery problems with different battery safety margins, and standard deviations of wind speed and wind direction. Results are compared among energy-aware Or-tools, stand-alone DRL planner, DRL planner (sample128), and DRL planner (EaPF).

Table 4.6: Route planning results for drone deliveries in the presence of stochastic winds

Method	Overall cost	Task accomplishment rate (%)	Energy cost	Battery depletion risk (%)
Or-tools(e) (mgn. 0%)	6.63	71.86	1.47	23.5
Or-tools(e) (mgn. 30%)	4.85	70.77	1.45	4.80
DRL(g)	3.87	80.10	1.73	1.50
DRL(s) (mgn. 0%)	4.33	87.22	1.68	13.67
DRL(s) (mgn. 30%)	3.28	85.30	1.71	1.04
DRL(EaPF)	2.90	88.65	1.67	1.00

We further extract representative points into Table 4.6. The simulation setting follows those used in the first simulation series. If we compare the original $DRL(g)$ with $DRL(s)$ (mgn. 0%), where a branch of 128 trajectories is generated with a good intention to extend the solution exploration range, it is evident that rather than achieving the desired effect, $DRL(s)$ exhibit an increased cost and a higher rate of battery depletion. As previously mentioned, the evaluation of route candidates in $DRL(s)$ relies on one-time performance, making the planned result highly unreliable and risky. After adopting 30% safety margin, results are more robust leading to a reduced energy risk from 12.67% to 1.04%, and with the higher task accomplishment rate, the performance of $DRL(s)$ (mgn. 30%) surpasses the $DRL(g)$. Nonetheless, this unstable property of $DRL(s)$ cannot be eliminated and thus motivated the proposed $DRL(EaPF)$, where each trajectory sample is well evaluated and risks are also assessed with the incorporation of statistical models of uncertainties. As demonstrated in the table, the $DRL(EaPF)$ features an enhanced performance from all perspectives, showing a low battery depletion rate and high task accomplishment rate.

The following two rows in Figure 4.6 compare the performance of four algorithms in terms of varying wind speed variances and wind direction variances respectively. Note that the $Or-tools(e)$ and $DRL(sample)$ are implemented under a safety margin of 30%, with which they achieve the best planning performance. As seen from the figure, increasing variances of wind variables moderately degrade the performance of all the methods. Larger variances of wind conditions bring stronger disturbances to the system, in which circumstances both task accomplish rate and flight safety in consideration of battery depletion are adversely affected. A noticeable observation

is that when the standard deviation exceeds a certain value, which is $\sigma_{v_w} = 1m/s$, $\sigma_{\theta_w} = 15deg$ in the case, the robustness brought to $DRL(sample)$ by setting the safety margin is compromised, resulting in a battery depletion rate as high as 6%. It indicates that a larger proportion of battery margin is required, however, it is at the cost of weakening task-related performance. Overall, the integration of the DRL planner with the proposed EaPF framework yields the best performance, achieving a consistently high task completion rate while guaranteeing a low risk of battery depletion.

4.9 Conclusion

This chapter investigated the energy-aware route planning problem for unmanned aerial logistic systems. As UAV couriers are mostly constrained by limited battery capacities, it is imperative to monitor and predict their energy consumption in route planning to avoid potential losses and stabilize the delivery service. For that reason, we proposed the EaPF, which consists of an energy predictive model, a route optimizer, and a risk assessment module. The energy predictive model adopts statistical techniques. Rather than time-consuming sampling, an MDN model is trained to estimate the energy distribution with respect to winds and payload weights. It firstly facilitates a DRL route optimizer by employing a novel Model-based Multi-Sampling route construction strategy. Secondly, it enables the establishment of a risk-aware metric to consider the possible task lateness and battery depletion in the solution evaluation. Finally, we demonstrate the performance of the MDN energy predictive model and EaPF framework. The results have demonstrated that by incorporating EaPF, the DRL planner produces more reliable and robust solutions with an overall cost reduction of 25% on overall cost and a 9% increase in task accomplishment compared to the stand-alone DRL planner, and guarantees that the energy risks are within the 2% threshold.

Chapter 5

Overall Discussion

This thesis aims to establish a generic decentralized mission planning paradigm for multi-UAV systems to enable a more efficient and stable operation under stochastic and dynamic real-world environments. In accordance with the objectives presented in Chapter 1, planning algorithms are developed and validated in simulations. An overall discussion is presented in this section, including the development of simulation models, achievements, advantages and drawbacks of the proposed methods, and some insights from the research.

5.1 Simulation Environment Development

The developed simulation environment provides a generic platform for the future demonstration of mission planning algorithms on drone delivery cases. The environment is modular so that the drone operators and the energy model can be easily shifted as required. Other additional modules can also be potentially plugged in when necessary.

A simulation model for multi-drone delivery has been established for the purpose of training and validating the proposed approaches, which include recharging services and pickup-delivery precedence constraints. Parameters settings are shown in Table 4.4 and Table 4.5. Energy cost simulation models are also plugged in as a changeable module.

Two simulation models for drone energy cost have been established for the purpose of verification and validation. Both take account of wind conditions and payload

weights. One is based upon the analytic model proposed in [185]. It is computationally efficient and thus used in the training progress of the DRL planner. The other is built on the TCN deep drone power model[210], featuring higher precision. It emulates the energy changes observed in the real environment and is used to generate datasets to train and update the MDN model of energy consumption.

5.2 Accomplishments Toward Objectives

- **Enhancing the robustness of mission planning solutions against uncertainties**

The robustness of the multi-UAV mission algorithm has been enhanced from the perspective of task coordination and route planning.

In Chapter 2, the robustness of task coordination is enhanced with the proposed auction-based strategies via optimizing the priori expectation of the score function in consideration of uncertainties. A new score function is formulated based upon the value function of MDPs, which takes account of uncertainties and all possible permutations of task orders, eliminating the performance loss with respect to the discrepancies between the planning and stochastic real world. Simulation results support the theoretical analysis, where the proposed method experiences much less performance degradation after introducing the stochastic drone flight speed, compared to the CBBA algorithm, as shown in Table 2.7 and Table 2.9.

In Chapter 3, the robustness of the drone delivery planning algorithm is further enhanced against stochastic wind conditions. The wind effect is identified as one of the most influential factors and modelled into a stochastic drone energy consumption model. A DRL planner is then proposed to handle the drone delivery in the presence of winds via casting it into an MDP. The robustness of solutions is demonstrated in the presence of stochastic winds, where the DRL planner shows a more stable performance with windier scenarios, as demonstrated in Table 3.5.

- **Reducing the computational complexity of mission planning algorithm**

In Chapter 2, the computational complexity for solving decentralized multi-agent stochastic planning is reduced from NEXP-hard to P-complete via decomposing the multi-agent MDPs into a series of weakly-coupled task-constrained

MDPs. The task dependency due to an exclusive allocation constraint is despatched from MMDPs, forming the specific task-constrained MDPs. The multi-agent stochastic planning problem is then decomposed into two sub-problems with the aim of finding the task allocation and the optimal policy sequentially. It degrades the multi-agent MDP into a set of parallel single-agent MDPs, reducing the computational complexity from NEXP-hard to P-complete.

- **Developing an efficient decision-making algorithm for UAV operating with recharging stations**

The drone decision-making in consideration of recharging is handled in Chapter 3 by formulating a DDP-R problem and proposing a novel AM-E model to solve it. DDP-R is a specific variant of the VRP, describing the pickup and delivery problem in the presence of recharging facilities. AM-E enhances the description ability of the attention model to handle the increased heterogeneity caused by recharging nodes. Adjacent edge information is incorporated via a series of edge-enhanced dot-product attention layers. Validation results demonstrate the superior performance of AM-E compared to the origin model, as shown in Table 3.3 and Figure 3.9. The efficiency of the proposed method is also demonstrated via simulations, as shown in Table 3.4, where the solution generation time is always within 1 second.

- **Constructing the efficient energy-aware planning method for multi-UAV systems**

An energy-aware planning framework is established in Chapter 4, featuring functions of energy cost prediction, risk assessment, and performance enhancement for route planners. A MDN model is established for efficiently predicting a future energy cost statistical model, enabling efficient energy-aware planning without disregarding the stochastic factors from the real world. The effectiveness of the proposed planning framework is demonstrated in the context of drone deliveries, reducing the objective cost by 23% and 31% respectively for stochastic and deterministic cases in comparison with the stand-alone DRL planner, as shown in Table ?? and Table ??.

5.3 Benefits to the Real World

Society and the economy will benefit from a higher quality of services and a more reliable operation of multi-UAV systems when incorporated with the proposed robust mission planning algorithm. Specifically, when the drone flight speed has a variance of $3m/s$, using the newly formulated score function can increase the overall reward by 24%, compared to the original CBBA. With the average wind speed of $5m/s$, using the integrated energy-aware planning algorithm, the overall cost function is reduced by 23% compared to the stand-alone planner. These optimized metrics indicate a reduction in waiting time, an increase in task accomplishment rate, and a saving of energy consumption. Regarding risk management, the proposed energy-aware planning framework enables risk regulation and generates more reliable solutions subject to a predefined risk threshold. That helps the authorities to assess the safety of flights above urban areas, mitigating the safety concerns of civil residents.

Meanwhile, the electric-driven UAV systems provide an environmentally friendly solution for many civil applications by making the overall system cleaner and quieter. The drone delivery system enables zero-carbon-emission air transport and mitigates road congestion. The proposed mission planning system further minimizes the impact of the system on climate and the environment by optimizing the objective regarding total energy consumption. Moreover, the decentralized implementation of mission planning systems provides a resource-efficient solution, which makes full use of the onboard processing capability, saving the cost of deploying massive ground infrastructures.

5.4 Gaps from the Real-World Implementation

This study has included practical scenarios such as uncertain energy consumption in the presence of winds, recharging services during the mission execution, and asynchronized and decentralized intra-group task coordination. Nevertheless, there are still gaps in implementing the proposed algorithms in the real world

1. *Communication latency and package loss.*

The study herein is conducted under the assumption of ideal communication conditions, which, however, is not always true in the real world. High latency

and package loss may exist in drone communications as a consequence of network congestion, bandwidth limitation and weak signals. It can significantly affect timely decision-making and effective allocation of resources in mission planning by diverging the local situational awareness through a time gap.

Developing mission planning algorithms that consider partial observation is an alternative mitigation for this issue. Partial observation features a limited amount of communication with neighbours so that guarantees efficiency. However, it poses challenges to the convergence and performance of the planning, as partial observability will introduce non-stationarity and extra disturbances into multi-agent decision-making. Second, optimizing the communication topology to enable low latency is another subject that has been pursued in multi-UAV decision-making. Predictive models and algorithms are used to anticipate the future positions and intentions of other UAVs, enabling more proactive decision-making and navigation.

2. *Limited capability in handling highly dynamic events.*

The proposed task coordination strategy is still based upon the basic framework of the CBBA algorithm, which involves several iterations between the bundle construction phase and the conflict resolution phase before arriving at the final consensus throughout the network. This indicates that the coordination process is not efficient for dynamic events, e.g., a breakdown of the drone or the arrival of a new request, especially for large-scale scenarios. Thus, the method might not be applicable for rapid-changing environments, as the rapid-changing environment demands a quick re-planning ability of the task coordination method.

From this perspective, the decision-making node is expected to perform in a more independent way, where others' behaviours can be inferred beforehand, instead of solely depending on communications for coordination. This could benefit less iteration for consensus, facilitating a quicker re-planning loop. MARL algorithms, either with independent learners or using a centralized training and decentralized execution framework, are a promising solution for handling dynamic mission scenarios.

3. *Adaptability to real-world cases with complex geo-features.*

All demonstrations in this thesis are conducted using a simulation environment, where the locations of tasks and the deployment of recharging facilities are randomly generated inside a square of a mission area. While this random

setting aims for a certain level of adaptability on different mission profiles, it might lack the ability to handle more problem-specific issues, e.g., geo-fences or non-fly zones.

Future investigation is desirable to consider more problem-specific simulation settings, to move the research closer to the real-world implementation. We expect to employ higher-fidelity simulation models for network training and validation. The refined simulation model will utilize Unreal Engine, Cesium, and AirSim models to establish and integrate high-resolution environment models, UAV models, and UTM services to provide a trustworthy solution for multi-UAV mission planning.

5.5 New Insights from the Research

We obtained some additional experience and highlighted some comments when doing this research.

An evaluation of the UAV group size for a given mission profile is beneficial prior to implementing the mission planning algorithm. During simulations, we found that the capability of a group of fixed-number UAVs might be insufficient for a larger-size mission profile. It results in many tasks left unfinished, especially with time-window restrictions. On the other hand, sending too many drones than it is needed causes a waste of resources. It is beneficial to estimate the most suitable group of UAVs given a mission configuration. However, there is a lack of investigation in the existing knowledge body regarding the choices of the UAV group size, thus we identify this research as a great help as a preliminary study to the mission planning algorithm development.

Simulation-based optimization techniques have great potential in enhancing the solution quality and robustness of AI approaches. Robustness has always been one of the greatest concerns in applying AI techniques. To that end, there have been many attempts focusing on modifying the neural networks, e.g., using Long Short-Term Memory networks. However, Despite the considerable progress they have made, the neural network has an inherent limitation due to the approximate property, and the resultant benefit from a sophisticated design sometimes is even less significant than simply adding a sample decoding scheme, as shown in [226]. In this sense, taking advantage of the idea of sim-heuristic approaches, the simulation-based optimization approaches can be used to generate future actions with AI planners, enabling

an evaluation of the further performance of the solution candidate. However, this proposition is basically model-based, and thus its good performance requires a well-designed simulator.

Chapter 6

Conclusion and Future work

6.1 Conclusion

This thesis investigates the mission planning problem for the multiple-UAV system under a dynamic and stochastic environment. In the thesis, a robust task coordination strategy, an efficient DRL planner and an energy-aware planning framework are proposed, which has improved upon the current state-of-the-arts and achieved the following objectives: (1) Enhancing the robustness of mission planning solutions against uncertainties; (2) Reducing the computational complexity of mission planning algorithms; (3) Developing an efficient decision-making algorithm for UAV operating with recharging stations; (4) Constructing the efficient energy-aware planning method for multi-UAV systems. The proposed mission planning scheme follows a two-tier hierarchical architecture (i.e., task coordination and individual planning) and enables the incorporation of energy awareness in the decision-making. Conclusions regarding these three progressions in the development are highlighted as follows.

Firstly, in Chapter 2, we present two decentralized auction-based task allocation approaches for multi-agent systems while considering the possible operation uncertainties. Through casting the problem as task-constrained MDPs, the task dependency due to an exclusive constraint is detached from M-MDPs and then resolved by adopting the auction-based coordination method. For these multi-agent stochastic planning problems, the suggested technique resolves the trade-off concern between computational tractability and solution quality. Further, this algorithm is proven to be convergent and can reach a posterior optimality guarantee of at least 50% with

submodular reward functions. The second algorithm, namely Deep Auction, is an approximate variant of the first suggested auction-based coordination system that includes two neural network approximators to facilitate large-scale implementations. By theoretical analysis, these two proposed algorithms achieve more robustness and less computing complexity, compared to the original CBBA with respect to the score function calculation and to the centralized MARL in terms of the problem dimensions. Finally, a case study of drone delivery with time windows is studied. The simulation results demonstrate the theoretical benefits of the recommended methodologies.

In Chapter 3, a new variant of the vehicle routing problem has been investigated with extra considerations of recharging operations and the effect of varying winds. To address the delivery problem efficiently, a DRL-based method with an AM-E network model has been presented and demonstrated with high-quality solutions. AM-E model is composed of a novel edge-enhanced dot-product attention layer via merging edge features into the information propagation, offering the network a stronger ability to describe the complex relationship among heterogeneous nodes. To validate, we have built a simulated environment with a stochastic energy consumption model under varying winds to imitate real-world delivery scenarios, in which the energy-constrained UAV could be recharged in the middle of mission execution. With the designed masking policy and reward function, the AM-E model has been trained and evaluated via the simulated environment. Results showed the AM-E is fast, robust and has better scalability compared to three baseline heuristic methods and a state-of-the-art deep learning model.

In Chapter 4, an Energy-aware Planning Framework (EaPF) is proposed, which consists of an energy predictive model, a route optimizer and a risk assessment module. The energy predictive model adopts statistical techniques. Rather than doing tedious and time-consuming sampling, a Mixture Density Network (MDN) model is trained to estimate the energy distribution with respect to the winds and payload weight. It firstly facilitates a DRL route optimizer by employing a novel Model-based Multi-Sampling route construction strategy, secondly enables the establishment of a risk-aware metric to account for the possible task lateness and battery depletion into the solution evaluation. Finally, we demonstrate the performance of the EaPF via a drone delivery case study, where the MDN model and DRL route optimizer are also validated. The results have shown that by incorporating EaPF, the DRL planner produces more reliable and robust solutions, reducing the objective cost by 23% and 31% respectively for stochastic and deterministic cases in comparison with

the stand-alone DRL planner using the Greedy decoding strategy.

6.2 Future Work

For future investigation, the current research could be extended from the following three aspects.

First, though numerical simulations herein are delicately designed for training and validation, more empirical data still benefits the technology consolidation, making it even closer to real-world scenarios. It would be desirable to gather more experimental data or construct finer environments using micro-simulation or discrete-event simulation to train the networks and validate them in more realistic scenarios.

Second, while focusing on the robustness against stochastic operations, this paper did not fully explore the performance of proposed technologies in handling dynamic events, such as a breakdown of drones and arrival of new requests. Rapid-changing lists of pending tasks might pose challenges demanding a quick replanning ability of the coordination method. Recent work on multi-target tracking [60] predicts neighbours' bids before the coordination. This could be borrowed to reduce the iterations for consensus, facilitating a quicker replanning loop.

Third, the scope of this thesis is mostly around intra-group cooperation focusing on the middle layer of the hierarchical architecture of mission planning. However, it is worthwhile extending the current investigation to include the interplay analysis with the higher layer of cloud and the lower layer of executives, as it would provide insight into systematic behaviours and facilitate the development of decision-making algorithms with stronger robustness.

References

- [1] NASA. Utm 101.
- [2] Jutta Kiener and Oskar Von Stryk. Towards cooperation of heterogeneous, autonomous robots: A case study of humanoid and wheeled robots. *Robotics and Autonomous Systems*, 58(7):921–929, 2010.
- [3] Shushman Choudhury, Kiril Solovey, Mykel J Kochenderfer, and Marco Pavone. Efficient large-scale multi-drone delivery using transit networks. *Journal of Artificial Intelligence Research*, 70:757–788, 2021.
- [4] Rajaa Ayadi and Youssef Benadada. Memetic algorithm for a multi-objective vehicle routing problem with multiple trips. *Int. J. Comput. Sci. Appl.*, 10(2):72–91, 2013.
- [5] Ruifan Liu, Hyo-Sang Shin, and Antonios Tsourdos. Edge-enhanced attentions for drone delivery in presence of winds and recharging stations. *Journal of Aerospace Information Systems*, pages 1–13, 2023.
- [6] Michal Tanas, Witold Holubowicz, Andrzej Adamczyk, and Grzegorz Taberski. The talos project. eu wide robotic border guard system. In *2011 16th International Conference on Methods & Models in Automation & Robotics*, pages 336–340. IEEE, 2011.
- [7] Navid Ali Khan, NZ Jhanjhi, Sarfraz Nawaz Brohi, Raja Sher Afgun Usmani, and Anand Nayyar. Smart traffic monitoring system using unmanned aerial vehicles (uavs). *Computer Communications*, 157:434–443, 2020.
- [8] Ilker Bozcan and Erdal Kayacan. Au-air: A multi-modal unmanned aerial vehicle dataset for low altitude traffic surveillance. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 8504–8510. IEEE, 2020.
- [9] David Erdos, Abraham Erdos, and Steve E Watkins. An experimental uav system for search and rescue challenge. *IEEE Aerospace and Electronic Systems Magazine*, 28(5):32–37, 2013.
- [10] Yulun Tian, Katherine Liu, Kyel Ok, Loc Tran, Danette Allen, Nicholas Roy, and Jonathan P How. Search and rescue under the forest canopy using multiple

- uavs. *The International Journal of Robotics Research*, 39(10-11):1201–1221, 2020.
- [11] Ignacio Martinez-Alpiste, Gelayol Golcarenarenji, Qi Wang, and Jose Maria Alcaraz-Calero. Search and rescue operation using uavs: A case study. *Expert Systems with Applications*, 178:114937, 2021.
- [12] Seyedamirali Mostafavizadeh Ardestani, Peter J Jin, Oliver Volkmann, Jie Gong, Zixiang Zhou, and Cecilia Feeley. 3d accident site reconstruction using unmanned aerial vehicles (uav). Technical report, 2016.
- [13] Farrokh Mohamadi. Vertical takeoff and landing (vtol) small unmanned aerial system for monitoring oil and gas pipelines, November 4 2014. US Patent 8,880,241.
- [14] Chuang Deng, Shengwei Wang, Zhi Huang, Zhongfu Tan, and Junyong Liu. Unmanned aerial vehicles for power line inspection: A cooperative way in platforms and communications. *J. Commun.*, 9(9):687–692, 2014.
- [15] Mark Jarman, John Vesey, and Paul Febvre. Unmanned aerial vehicles (uavs) for uk agriculture: creating an invisible precision farming technology. *White Paper*, July, 2016.
- [16] Dionisio Doering, Arthur Benenmann, Rafael Lerm, Edison Pignaton de Freitas, Ivan Muller, Jean M Winter, and Carlos Eduardo Pereira. Design and optimization of a heterogeneous platform for multiple uav use in precision agriculture applications. *IFAC Proceedings Volumes*, 47(3):12272–12277, 2014.
- [17] Anibal Ollero and L Merino. Unmanned aerial vehicles as tools for forest-fire fighting. *Forest Ecology and Management*, 234(1):S263, 2006.
- [18] S Sudhakar, Varadarajan Vijayakumar, C Sathiya Kumar, V Priya, Logesh Ravi, and V Subramaniaswamy. Unmanned aerial vehicle (uav) based forest fire detection and monitoring for reducing false alarms in forest-fires. *Computer Communications*, 149:1–16, 2020.
- [19] Shiva Ram Reddy Singireddy and Tugrul U Daim. Technology roadmap: Drone delivery–amazon prime air. *Infrastructure and Technology Management: Contributions from the Energy, Healthcare and Transportation Sectors*, pages 387–412, 2018.
- [20] Charles T Howell III, Frank Jones, Taylor Thorson, Richard Grube, Cecil Mellanson, Lee Joyce, John Coggin, and Jack Kennedy. The first government sanctioned delivery of medical supplies by remotely controlled unmanned aerial system (uas). Technical report, 2016.
- [21] Unmanned Cargo. Drones going postal a summary of postal service delivery drone trials, 2017.

- [22] Guillaume Hoareau, Johannes J Liebenberg, John G Musial, and Todd R Whitman. Package transport by unmanned aerial vehicles, August 15 2017. US Patent 9,731,821.
- [23] Georgios Fevgas, Thomas Lagkas, Vasileios Argyriou, and Panagiotis Sariannidis. Coverage path planning methods focusing on energy efficient and cooperative strategies for unmanned aerial vehicles. *Sensors*, 22(3):1235, 2022.
- [24] Allison Ryan, Marco Zennaro, Adam Howell, Raja Sengupta, and J Karl Hedrick. An overview of emerging results in cooperative uav control. In *2004 43rd IEEE Conference on Decision and Control (CDC)(IEEE Cat. No. 04CH37601)*, volume 1, pages 602–607. IEEE, 2004.
- [25] Edwin Ordoukhanian and Azad M Madni. Resilient multi-uav operation: key concepts and challenges. In *54th AIAA Aerospace Sciences Meeting*, page 0475, 2016.
- [26] Gustavo SC Avellar, Guilherme AS Pereira, Luciano CA Pimenta, and Paulo Iscold. Multi-uav routing for area coverage and remote sensing with minimum time. *Sensors*, 15(11):27783–27803, 2015.
- [27] Syed Agha Hassnain Mohsan, Muhammad Asghar Khan, Fazal Noor, Insaf Ullah, and Mohammed H Alsharif. Towards the unmanned aerial vehicles (uavs): A comprehensive review. *Drones*, 6(6):147, 2022.
- [28] Yang Gao and Dongsheng Li. Unmanned aerial vehicle swarm distributed co-operation method based on situation awareness consensus and its information processing mechanism. *Knowledge-Based Systems*, 188:105034, 2020.
- [29] Johannes Autenrieb, Natalia Strawa, Hyo-Sang Shin, and Ju-Hyeon Hong. A mission planning and task allocation framework for multi-uav swarm co-ordination. In *2019 Workshop on Research, Education and Development of Unmanned Aerial Systems (RED UAS)*, pages 297–304. IEEE, 2019.
- [30] Keum-Seong Kim, Ho-Yeon Kim, and Han-Lim Choi. A bid-based grouping method for communication-efficient decentralized multi-uav task allocation. *International Journal of Aeronautical and Space Sciences*, 21:290–302, 2020.
- [31] Jiawei Cao, Rodney Swee Huat Teo, Sunan Huang, and Qinyuan Ren. A fast, robust and decentralized approach for altitude de-confliction of multiple uavs. In *2019 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 79–85. IEEE, 2019.
- [32] Boris Galkin, Jacek Kibilda, and Luiz A DaSilva. Uavs as mobile infrastructure: Addressing battery lifetime. *IEEE Communications Magazine*, 57(6):132–137, 2019.
- [33] Ty Nguyen and Tsz-Chiu Au. Extending the range of delivery drones by exploratory learning of energy models. In *AAMAS*, pages 1658–1660, 2017.

- [34] Mohammad Sadra Rajabi, Pedram Beigi, and Sina Aghakhani. Drone delivery systems and energy management: A review and future trends. *arXiv preprint arXiv:2206.10765*, 2022.
- [35] Haque Nawaz, Husnain Mansoor Ali, and Asif Ali Laghari. Uav communication networks issues: a review. *Archives of Computational Methods in Engineering*, 28:1349–1369, 2021.
- [36] Bo Hang Wang, Dao Bo Wang, Zain Anwar Ali, Bai Ting Ting, and Hao Wang. An overview of various kinds of wind effects on unmanned aerial vehicle. *Measurement and Control*, 52(7-8):731–739, 2019.
- [37] Anandrao Shesherao Biradar. *Wind estimation and effects of wind on waypoint navigation of UAVs*. Arizona State University, 2014.
- [38] Xueyuan Wang, M Cenk Gursoy, Tugba Erpek, and Yalin E Sagduyu. Learning-based uav path planning for data collection with integrated collision avoidance. *IEEE Internet of Things Journal*, 9(17):16663–16676, 2022.
- [39] Mathijs De Weerd and Brad Clement. Introduction to planning in multiagent systems. *Multiagent and Grid Systems*, 5(4):345–355, 2009.
- [40] Fei Yan, Xiaoping Zhu, Zhou Zhou, and Jing Chu. A hierarchical mission planning method for simultaneous arrival of multi-uav coalition. *Applied sciences*, 9(10):1986, 2019.
- [41] Simon Lacroix, Rachid Alami, Thomas Lemaire, Gautier Hattenberger, and Jérémie Gancet. Decision making in multi-uavs systems: Architecture and algorithms. *Multiple heterogeneous unmanned aerial vehicles*, 37:15–48, 2007.
- [42] Jia Song, Kai Zhao, and Yang Liu. Survey on mission planning of multiple unmanned aerial vehicles. *Aerospace*, 10(3):208, 2023.
- [43] Yara Rizk, Mariette Awad, and Edward W Tunstel. Cooperative heterogeneous multi-robot systems: A survey. *ACM Computing Surveys (CSUR)*, 52(2):1–31, 2019.
- [44] Ruifan Liu, Hyo-Sang Shin, Bonbon Yan, and Antonios Tsourdos. An auction-based coordination strategy for task-constrained multi-agent stochastic planning with submodular rewards. *arXiv preprint arXiv:2212.14624*, 2022.
- [45] Yu Zhang and Lynne E Parker. Iq-asymtre: Forming executable coalitions for tightly coupled multirobot tasks. *IEEE Transactions on Robotics*, 29(2):400–416, 2012.
- [46] Chase C Murray and Amanda G Chu. The flying sidekick traveling salesman problem: Optimization of drone-assisted parcel delivery. *Transportation Research Part C: Emerging Technologies*, 54:86–109, 2015.

- [47] Xiaoshan Bai, Ming Cao, Weisheng Yan, and Shuzhi Sam Ge. Efficient routing for precedence-constrained package delivery for heterogeneous vehicles. *IEEE Transactions on Automation Science and Engineering*, 17(1):248–260, 2019.
- [48] Kevin Dorling, Jordan Heinrichs, Geoffrey G Messier, and Sebastian Magierowski. Vehicle routing problems for drone delivery. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 47(1):70–85, 2016.
- [49] Francesco Betti Sorbelli, Federico Corò, Sajal K Das, and Cristina M Pinotti. Energy-constrained delivery of goods with drones under varying wind conditions. *IEEE Transactions on Intelligent Transportation Systems*, 22(9):6048–6060, 2020.
- [50] Amila Thibbotuwawa, Grzegorz Bocewicz, Grzegorz Radzki, Peter Nielsen, and Zbigniew Banaszak. Uav mission planning resistant to weather uncertainty. *Sensors*, 20(2):515, 2020.
- [51] Sara Imran Khan, Zakria Qadir, Hafiz Suliman Munawar, Soumya Ranjan Nayak, Anil Kumar Budati, Keshav D Verma, and Deo Prakash. Uavs path planning architecture for effective medical emergency response in future networks. *Physical Communication*, 47:101337, 2021.
- [52] Jian Tang, Kejun Zhu, Haixiang Guo, Chengzhu Gong, Can Liao, and Shuwen Zhang. Using auction-based task allocation scheme for simulation optimization of search and rescue in disaster relief. *Simulation Modelling Practice and Theory*, 82:132–146, 2018.
- [53] Ebtehal Turki Alotaibi, Shahad Saleh Alqefari, and Anis Koubaa. Lsar: Multi-uav collaboration for search and rescue missions. *IEEE Access*, 7:55817–55832, 2019.
- [54] Samira Hayat, Evşen Yanmaz, Christian Bettstetter, and Timothy X Brown. Multi-objective drone path planning for search and rescue with quality-of-service requirements. *Autonomous Robots*, 44(7):1183–1198, 2020.
- [55] Jürgen Scherer, Saeed Yahyanejad, Samira Hayat, Evsen Yanmaz, Torsten Andre, Asif Khan, Vladimir Vukadinovic, Christian Bettstetter, Hermann Hellwagner, and Bernhard Rinner. An autonomous multi-uav system for search and rescue. In *Proceedings of the First Workshop on Micro Aerial Vehicle Networks, Systems, and Applications for Civilian Use*, pages 33–38, 2015.
- [56] Teng Li, Hyo-Sang Shin, and Antonios Tsourdos. Efficient decentralized task allocation for uav swarms in multi-target surveillance missions. In *2019 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 61–68. IEEE, 2019.
- [57] Nathan Patrizi, Georgios Fragkos, Kendric Ortiz, Meeko Oishi, and Eirini Eleni Tsiropoulou. A uav-enabled dynamic multi-target tracking and sensing framework. In *GLOBECOM 2020-2020 IEEE Global Communications Conference*, pages 1–6. IEEE, 2020.

- [58] Jesus Capitan, Matthijs TJ Spaan, Luis Merino, and Anibal Ollero. Decentralized multi-robot cooperation with auctioned pomdps. *The International Journal of Robotics Research*, 32(6):650–671, 2013.
- [59] Jesús Capitan, Luis Merino, and Aníbal Ollero. Cooperative decision-making under uncertainties for multi-target surveillance with multiples uavs. *Journal of Intelligent & Robotic Systems*, 84:371–386, 2016.
- [60] Yunyun Zhao, Xiangke Wang, Chang Wang, Yirui Cong, and Lincheng Shen. Systemic design of distributed multi-uav cooperative decision-making for multi-target tracking. *Autonomous Agents and Multi-Agent Systems*, 33:132–158, 2019.
- [61] ZHOU Wenhong, LI Jie, LIU Zhihong, and SHEN Lincheng. Improving multi-target cooperative tracking guidance for uav swarms using multi-agent reinforcement learning. *Chinese Journal of Aeronautics*, 35(7):100–112, 2022.
- [62] Marija Popović, Teresa Vidal-Calleja, Gregory Hitz, Jen Jen Chung, Inkyu Sa, Roland Siegwart, and Juan Nieto. An informative path planning framework for uav-based terrain monitoring. *Autonomous Robots*, 44:889–911, 2020.
- [63] Nicolas Bono Rossello, Renzo Fabrizio Carpio, Andrea Gasparri, and Emanuele Garone. Information-driven path planning for uav with limited autonomy in large-scale field monitoring. *IEEE Transactions on Automation Science and Engineering*, 19(3):2450–2460, 2021.
- [64] Zhongjie Lin. *Multiple UAV cooperation for wildfire monitoring*. University of Toronto (Canada), 2017.
- [65] Sanjay Dhir and Sanjay Dhir. Dhl. *Cases in Strategic Management: A Flexibility Perspective*, pages 55–72, 2019.
- [66] J Vanian. Drones may one day buzz inside a walmart near you. *Fortune*, 2017.
- [67] Hossein Eskandaripour and Enkhsaikhan Boldsaikhan. Last-mile drone delivery: Past, present, and future. *Drones*, 7(2):77, 2023.
- [68] Judy Scott and Carlton Scott. Drone delivery models for healthcare. 2017.
- [69] Ali-akbar Agha-mohammadi, N Kemal Ure, Jonathan P How, and John Vian. Health aware stochastic planning for persistent package delivery missions using quadrotors. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3389–3396. IEEE, 2014.
- [70] Claire Morley, Maria Unwin, Gregory M Peterson, Jim Stankovich, and Leigh Kinsman. Emergency department crowding: a systematic review of causes, consequences and solutions. *PloS one*, 13(8):e0203316, 2018.
- [71] Nitin Kaushik, Victor S Khangulov, Matthew O’Hara, and Ramy Arnaout. Reduction in laboratory turnaround time decreases emergency room length of stay. *Open access emergency medicine: OAEM*, 10:37, 2018.

- [72] Maryam Torabbeigi, Gino J Lim, and Seon Jin Kim. Drone delivery scheduling optimization considering payload-induced battery consumption rates. *Journal of Intelligent & Robotic Systems*, 97:471–487, 2020.
- [73] Ruifan Liu, Hyo-sang Shin, Minguk Seo, and Antonios Tsourdos. Energy-aware route planning for drone delivery systems. In *Proceedings of the 2022 CEAS EuroGNC conference*. Berlin, Germany, May 2022. CEAS-GNC-2022-045.
- [74] Comets project. <http://www.comets-uavs.org/>, 2002-2005.
- [75] Close-search. <https://cordis.europa.eu/project/id/248137>, 2009.
- [76] Rescuecell. <https://cordis.europa.eu/project/id/315007/reporting>.
- [77] J Eyerman, G Crispino, A Zamarro, and R Durscher. Drone efficacy study (des): Evaluating the impact of drones for locating lost persons in search and rescue events brussels. Technical report, 2018.
- [78] Jorge Pena Queralta, Jussi Taipalmaa, Bilge Can Pullinen, Victor Kathan Sarker, Tuan Nguyen Gia, Hannu Tenhunen, Moncef Gabbouj, Jenni Raitoharju, and Tomi Westerlund. Collaborative multi-robot systems for search and rescue: Coordination and perception. *arXiv preprint arXiv:2008.12610*, 2020.
- [79] Joel G Manathara, PB Sujit, and Randal W Beard. Multiple uav coalitions for a search and prosecute mission. *Journal of Intelligent & Robotic Systems*, 62:125–158, 2011.
- [80] Hyondong Oh. *Towards autonomous surveillance and tracking by multiple UAVs*. PhD thesis, Cranfield University, 2013.
- [81] Mário Gabriel Santos De Campos, Caroline PC Chanel, Corentin Chauffaut, and Jérôme Lacan. Towards a blockchain-based multi-uav surveillance system. *Frontiers in Robotics and AI*, 8:557692, 2021.
- [82] Rema Ibrahim Hamad Abushahma, Musab AM Ali, Nur Adilah Abd Rahman, and Omar Ismael Al-Sanjary. Comparative features of unmanned aerial vehicle (uav) for border protection of libya: a review. In *2019 IEEE 15th International Colloquium on Signal Processing & Its Applications (CSPA)*, pages 114–119. IEEE, 2019.
- [83] Narina Thakur, Preeti Nagrath, Rachna Jain, Dharmender Saini, Nitika Sharma, and D Jude Hemanth. Artificial intelligence techniques in smart cities surveillance using uavs: a survey. *Machine Intelligence and Data Analytics for Sustainable Future Smart Cities*, pages 329–353, 2021.
- [84] Ryan D Friese, James A Crowder, Howard Jay Siegel, and John N Carbone. Surveillance mission planning: Model, performance measure, bi-objective analysis, partial surveils. In *21st International Conference on Artificial Intelligence (ICAI ‘19)*, 2019.

- [85] Alberto Elfes. Using occupancy grids for mobile robot perception and navigation. *Computer*, 22(6):46–57, 1989.
- [86] Teresa Vidal-Calleja, Daobilige Su, Freek De Bruijn, and Jaime Valls Miro. Learning spatial correlations for bayesian fusion in pipe thickness mapping. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 683–690. IEEE, 2014.
- [87] Geoffrey A Hollinger and Gaurav S Sukhatme. Sampling-based robotic information gathering algorithms. *The International Journal of Robotics Research*, 33(9):1271–1287, 2014.
- [88] Tauã M Cabreira, Lisane B Brisolara, and Ferreira Jr Paulo R. Survey on coverage path planning with unmanned aerial vehicles. *Drones*, 3(1):4, 2019.
- [89] Doo-Hyun Cho, Jung-Su Ha, Sujin Lee, Sunghyun Moon, and Han-Lim Choi. Informative path planning and mapping with multiple uavs in wind fields. In *Distributed Autonomous Robotic Systems: The 13th International Symposium*, pages 269–283. Springer, 2018.
- [90] G Ayorkor Korsah, Anthony Stentz, and M Bernardine Dias. A comprehensive taxonomy for multi-robot task allocation. *The International Journal of Robotics Research*, 32(12):1495–1512, 2013.
- [91] Brian P Gerkey and Maja J Matarić. A formal analysis and taxonomy of task allocation in multi-robot systems. *The International journal of robotics research*, 23(9):939–954, 2004.
- [92] Rainer E. Burkard, Mauro Dell’Amico, and Silvano Martello. Assignment problems. In *IFIP Congress: Fundamentals - Foundations of Computer Science*, 1998.
- [93] Alaa Khamis, Ahmed Hussein, and Ahmed Elmogy. Multi-robot task allocation: A review of the state-of-the-art. *Cooperative robots and sensor networks 2015*, pages 31–51, 2015.
- [94] Han-Lim Choi, Luc Brunet, and Jonathan P How. Consensus-based decentralized auctions for robust task allocation. *IEEE transactions on robotics*, 25(4):912–926, 2009.
- [95] Hyo-Sang Shin, Teng Li, Hae-In Lee, and Antonios Tsourdos. Sample greedy based task allocation for multiple robot systems. *Swarm Intelligence*, 16(3):233–260, 2022.
- [96] Sameera S Ponda. *Robust distributed planning strategies for autonomous multi-agent teams*. PhD thesis, Massachusetts Institute of Technology, 2012.
- [97] Luke Johnson, Han-Lim Choi, Sameera Ponda, and Jonathan P How. Allowing non-submodular score functions in distributed task allocation. In *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*, pages 4702–4708. IEEE, 2012.

- [98] Ruifan Liu, Minguk Seo, Binbin Yan, and Antonios Tsourdos. Decentralized task allocation for multiple uavs with task execution uncertainties. In *2020 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 271–278. IEEE, 2020.
- [99] Xiaowei Fu, Hui Wang, Bin Li, and Xiaoguang Gao. An efficient sampling-based algorithms using active learning and manifold learning for multiple unmanned aerial vehicle task allocation under uncertainty. *Sensors*, 18(8):2645, 2018.
- [100] Christian Blum and Andrea Roli. Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM computing surveys (CSUR)*, 35(3):268–308, 2003.
- [101] Helena Ramalhinho Lourenço, Olivier C Martin, and Thomas Stützle. Iterated local search: Framework and applications. In *Handbook of metaheuristics*, pages 129–168. Springer, 2019.
- [102] Christos Voudouris, Edward PK Tsang, and Abdullah Alsheddy. Guided local search. In *Handbook of metaheuristics*, pages 321–361. Springer, 2010.
- [103] Pierre Hansen, Nenad Mladenović, and José A Moreno Pérez. Variable neighbourhood search: methods and applications. *Annals of Operations Research*, 175(1):367–407, 2010.
- [104] Gamal Attiya and Yskandar Hamam. Task allocation for maximizing reliability of distributed systems: A simulated annealing approach. *Journal of parallel and Distributed Computing*, 66(10):1259–1266, 2006.
- [105] Jameer G Kotwal and Tanuja S Dhope. Solving task allocation to the worker using genetic algorithm. *International Journal of Computer Science and Information Technologies*, 6(4):3736–3741, 2015.
- [106] Xiaoshan Bai, Weisheng Yan, Shuzhi Sam Ge, and Ming Cao. An integrated multi-population genetic algorithm for multi-vehicle task assignment in a drift field. *Information Sciences*, 453:227–238, 2018.
- [107] Parag C Pendharkar. An ant colony optimization heuristic for constrained task allocation problem. *Journal of Computational Science*, 7:37–47, 2015.
- [108] Hamid Reza Boveiri. A novel aco-based static task scheduling approach for multiprocessor environments. *International Journal of Computational Intelligence Systems*, 9(5):800–811, 2016.
- [109] Riccardo Poli. Analysis of the publications on the applications of particle swarm optimisation. *Journal of Artificial Evolution and Applications*, 2008, 2008.
- [110] Changyun Wei, Ze Ji, and Boliang Cai. Particle swarm optimization for cooperative multi-robot task allocation: a multi-objective approach. *IEEE Robotics and Automation Letters*, 5(2):2530–2537, 2020.

- [111] Adil Baykasoglu, Lale Ozbakir, and Pinar Tapkan. Artificial bee colony algorithm and its application to generalized assignment problem. *Swarm Intelligence: Focus on Ant and particle swarm optimization*, 1, 2007.
- [112] Kai Zhang, Emmanuel G Collins Jr, and Dongqing Shi. Centralized and distributed task allocation in multi-robot teams via a stochastic clustering auction. *ACM Transactions on Autonomous and Adaptive Systems (TAAS)*, 7(2):1–22, 2012.
- [113] Kai Zhang, Emmanuel G Collins, and Adrian Barbu. An efficient stochastic clustering auction for heterogeneous robotic collaborative teams. *Journal of Intelligent & Robotic Systems*, 72:541–558, 2013.
- [114] Mohamed Badreldin, Ahmed Hussein, and Alaa Khamis. A comparative study between optimization and market-based approaches to multi-robot task allocation. *Advances in Artificial Intelligence*, 3, 2013.
- [115] George Marios Skaltsis, Hyo-Sang Shin, and Antonios Tsourdos. A survey of task allocation techniques in mas. In *2021 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 488–497. IEEE, 2021.
- [116] Fatma Faruq, David Parker, Bruno Lacerda, and Nick Hawes. Simultaneous task allocation and planning under uncertainty. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3559–3564. IEEE, 2018.
- [117] Yara Rizk, Mariette Awad, and Edward W Tunstel. Decision making in multi-agent systems: A survey. *IEEE Transactions on Cognitive and Developmental Systems*, 10(3):514–529, 2018.
- [118] Yoav Shoham and Kevin Leyton-Brown. *Multiagent systems: Algorithmic, game-theoretic, and logical foundations*. Cambridge University Press, 2008.
- [119] Craig Boutilier. Sequential optimality and coordination in multiagent systems. In *IJCAI*, volume 99, pages 478–485, 1999.
- [120] Daniel S Bernstein, Robert Givan, Neil Immerman, and Shlomo Zilberstein. The complexity of decentralized control of markov decision processes. *Mathematics of operations research*, 27(4):819–840, 2002.
- [121] Raphen Becker, Shlomo Zilberstein, Victor Lesser, and Claudia V Goldman. Solving transition independent decentralized markov decision processes. *Journal of Artificial Intelligence Research*, 22:423–455, 2004.
- [122] Rajiv Kumar, Pradeep Varakantham, and Akshat Kumar. Decentralized planning in stochastic environments with submodular rewards. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31, 2017.
- [123] Francisco S Melo and Manuela Veloso. Decentralized mdps with sparse interactions. *Artificial Intelligence*, 175(11):1757–1789, 2011.

- [124] Joshua David Redding. *Approximate multi-agent planning in dynamic and uncertain environments*. PhD thesis, Massachusetts Institute of Technology, 2011.
- [125] Lloyd S Shapley. Stochastic games. *Proceedings of the national academy of sciences*, 39(10):1095–1100, 1953.
- [126] AnYuan Guo and Victor R Lesser. Planning for weakly-coupled partially observable stochastic games. In *IJCAI*, pages 1715–1716, 2005.
- [127] Eric A Hansen, Daniel S Bernstein, and Shlomo Zilberstein. Dynamic programming for partially observable stochastic games. In *AAAI*, volume 4, pages 709–715, 2004.
- [128] Rosemary Emery-Montemerlo, Geoff Gordon, Jeff Schneider, and Sebastian Thrun. Game theoretic control for robot teams. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pages 1163–1169. IEEE, 2005.
- [129] Rosemary Emery-Montemerlo, Geoff Gordon, Jeff Schneider, and Sebastian Thrun. Approximate solutions for partially observable stochastic games with common payoffs. In *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems, 2004. AAMAS 2004.*, pages 136–143. IEEE, 2004.
- [130] Gerardo Beni and Jing Wang. Swarm intelligence in cellular robotic systems. In *Robots and biological systems: towards a new bionics?*, pages 703–712. Springer, 1993.
- [131] Katia P Sycara. Multiagent systems. *AI magazine*, 19(2):79–79, 1998.
- [132] James Kennedy and Russell Eberhart. Particle swarm optimization. In *Proceedings of ICNN’95-international conference on neural networks*, volume 4, pages 1942–1948. IEEE, 1995.
- [133] Haibin Duan and Peixin Qiao. Pigeon-inspired optimization: a new swarm intelligence optimizer for air robot path planning. *International journal of intelligent computing and cybernetics*, 7(1):24–37, 2014.
- [134] Dušan Teodorovic and Mauro Dell’Orco. Bee colony optimization—a cooperative learning approach to complex transportation problems. *Advanced OR and AI methods in transportation*, 51:60, 2005.
- [135] Manuele Brambilla, Eliseo Ferrante, Mauro Birattari, and Marco Dorigo. Swarm robotics: a review from the swarm engineering perspective. *Swarm Intelligence*, 7:1–41, 2013.
- [136] Levent Bayındır. A review of swarm robotics tasks. *Neurocomputing*, 172:292–321, 2016.

- [137] Jan Carlo Barca and Y Ahmet Sekercioglu. Swarm robotics reviewed. *Robotica*, 31(3):345–359, 2013.
- [138] Hazim Shakhatreh, Ahmad H Sawalmeh, Ala Al-Fuqaha, Zuochao Dou, Eyad Almaita, Issa Khalil, Noor Shamsiah Othman, Abdallah Khreishah, and Mohsen Guizani. Unmanned aerial vehicles (uavs): A survey on civil applications and key research challenges. *Ieee Access*, 7:48572–48634, 2019.
- [139] Wanqing Zhao, Qinggang Meng, and Paul WH Chung. A heuristic distributed task allocation method for multivehicle multitask problems and its application to search and rescue scenario. *IEEE transactions on cybernetics*, 46(4):902–915, 2015.
- [140] Khajonpong Akkarajitsakul, Ekram Hossain, and Dusit Niyato. Coalition-based cooperative packet delivery under uncertainty: A dynamic bayesian coalitional game. *IEEE Transactions on Mobile Computing*, 12(2):371–385, 2011.
- [141] Pritee Agrawal, Pradeep Varakantham, and William Yeoh. Scalable greedy algorithms for task/resource constrained multi-agent stochastic planning. 2016.
- [142] Pau Segui-Gasco, Hyo-Sang Shin, Antonios Tsourdos, and VJ Segui. Decentralised submodular multi-robot task allocation. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2829–2834. IEEE, 2015.
- [143] Zhengjia Xu, Ivan Petrunin, Teng Li, and Antonios Tsourdos. Efficient allocation for downlink multi-channel noma systems considering complex constraints. *Sensors*, 21(5):1833, 2021.
- [144] Frits De Nijs, Erwin Walraven, Mathijs De Weerd, and Matthijs Spaan. Constrained multiagent markov decision processes: A taxonomy of problems and algorithms. *Journal of Artificial Intelligence Research*, 70:955–1001, 2021.
- [145] Andrey Kolobov. Planning with markov decision processes: An ai perspective. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 6(1):1–210, 2012.
- [146] Pritee Agrawal, Pradeep Varakantham, and William Yeoh. Decentralized planning for non-dedicated agent teams with submodular rewards in uncertain environments. 2018.
- [147] Trevor Campbell, Luke Johnson, and Jonathan P How. Multiagent allocation of markov decision process tasks. In *2013 American Control Conference*, pages 2356–2361. IEEE, 2013.
- [148] Carlos E Guestrin and Geoffrey Gordon. Distributed planning in hierarchical factored mdps. *arXiv preprint arXiv:1301.0571*, 2012.

-
- [149] Nicolas Meuleau, Milos Hauskrecht, Kee-Eung Kim, Leonid Peshkin, Leslie Pack Kaelbling, Thomas L Dean, and Craig Boutilier. Solving very large weakly coupled markov decision processes. In *AAAI/IAAI*, pages 165–172, 1998.
 - [150] Craig Boutilier and Tyler Lu. Budget allocation using weakly coupled, constrained markov decision processes. 2016.
 - [151] George L Nemhauser, Laurence A Wolsey, and Marshall L Fisher. An analysis of approximations for maximizing submodular set functions—i. *Mathematical programming*, 14:265–294, 1978.
 - [152] Andreas Krause and Daniel Golovin. Submodular function maximization. *Tractability*, 3:71–104, 2014.
 - [153] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.
 - [154] Wouter Kool, Herke Van Hoof, and Max Welling. Attention, learn to solve routing problems! *arXiv preprint arXiv:1803.08475*, 2018.
 - [155] Guillaume Bono, Jilles S Dibangoye, Olivier Simonin, Laëtitia Matignon, and Florian Pereyron. Solving multi-agent routing problems using deep attention mechanisms. *IEEE Transactions on Intelligent Transportation Systems*, 22(12):7804–7813, 2020.
 - [156] Paolo Toth and Daniele Vigo. *The vehicle routing problem*. SIAM, 2002.
 - [157] Insu Hong, Michael Kuby, and Alan T Murray. A range-restricted recharging station coverage model for drone delivery service planning. *Transportation Research Part C: Emerging Technologies*, 90:198–212, 2018.
 - [158] Jean Louis Fendji Kedieng Ebongue, Israel Bayaola, Christopher Thron, and Anna Förster. Charging stations placement in drone path planning for large space surveillance. In *CARI 2020-Colloque Africain sur la Recherche en Informatique et en Mathématiques Appliquées*, 2020.
 - [159] Jane Lin, Wei Zhou, and Ouri Wolfson. Electric vehicle routing problem. *Transportation research procedia*, 12:508–521, 2016.
 - [160] Tomislav Erdelić and Tonči Carić. A survey on the electric vehicle routing problem: variants and solution approaches. *Journal of Advanced Transportation*, 2019, 2019.
 - [161] JQ James, Wen Yu, and Jiatao Gu. Online vehicle routing with neural combinatorial optimization and deep reinforcement learning. *IEEE Transactions on Intelligent Transportation Systems*, 20(10):3806–3817, 2019.

- [162] Hesamoddin Tahami, Ghaith Rabadi, and Mohamed Haouari. Exact approaches for routing capacitated electric vehicles. *Transportation Research Part E: Logistics and Transportation Review*, 144:102126, 2020.
- [163] Guy Desaulniers, Fausto Errico, Stefan Irnich, and Michael Schneider. Exact algorithms for electric vehicle-routing problems with time windows. *Operations Research*, 64(6):1388–1405, 2016.
- [164] Ilker Kucukoglu, Reginald Dewil, and Dirk Cattrysse. The electric vehicle routing problem and its variations: A literature review. *Computers & Industrial Engineering*, 161:107650, 2021.
- [165] KS Ruland and EY Rodin. The pickup and delivery problem: Faces and branch-and-cut algorithm. *Computers & mathematics with applications*, 33(12):1–13, 1997.
- [166] Stefan Ropke and Jean-François Cordeau. Branch and cut and price for the pickup and delivery problem with time windows. *Transportation Science*, 43(3):267–286, 2009.
- [167] Wen-Chyuan Chiang and Robert A Russell. Simulated annealing metaheuristics for the vehicle routing problem with time windows. *Annals of Operations Research*, 63:3–27, 1996.
- [168] Jörg Homberger and Hermann Gehring. Two evolutionary metaheuristics for the vehicle routing problem with time windows. *INFOR: Information Systems and Operational Research*, 37(3):297–318, 1999.
- [169] Paulo Roberto de Oliveira da Costa, Stefano Mauceri, Paula Carroll, and Fabiano Pallonetto. A genetic algorithm for a green vehicle routing problem. *Electronic notes in discrete mathematics*, 64:65–74, 2018.
- [170] Sina Rastani and Bülent Çatay. A large neighborhood search-based matheuristic for the load-dependent electric vehicle routing problem with time windows. *Annals of Operations Research*, pages 1–33, 2021.
- [171] Veaceslav Ghilas, Emrah Demir, and Tom Van Woensel. An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows and scheduled lines. *Computers & Operations Research*, 72:12–30, 2016.
- [172] Dominik Goeke. Granular tabu search for the pickup and delivery problem with time windows and electric vehicles. *European Journal of Operational Research*, 278(3):821–836, 2019.
- [173] Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. Pointer networks. *Advances in neural information processing systems*, 28, 2015.
- [174] Irwan Bello, Hieu Pham, Quoc V Le, Mohammad Norouzi, and Samy Bengio. Neural combinatorial optimization with reinforcement learning. *arXiv preprint arXiv:1611.09940*, 2016.

- [175] Mohammadreza Nazari, Afshin Oroojlooy, Lawrence Snyder, and Martin Takác. Reinforcement learning for solving the vehicle routing problem. *Advances in neural information processing systems*, 31, 2018.
- [176] Bo Lin, Bissan Ghaddar, and Jatin Nathwani. Deep reinforcement learning for the electric vehicle routing problem with time windows. *IEEE Transactions on Intelligent Transportation Systems*, 23(8):11528–11538, 2021.
- [177] Jingwen Li, Liang Xin, Zhiguang Cao, Andrew Lim, Wen Song, and Jie Zhang. Heterogeneous attentions for solving pickup and delivery problem via deep reinforcement learning. *IEEE Transactions on Intelligent Transportation Systems*, 23(3):2306–2315, 2021.
- [178] Liyu Gong and Qiang Cheng. Exploiting edge features for graph neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9211–9219, 2019.
- [179] Ziming Wang, Jun Chen, and Haopeng Chen. Egat: Edge-featured graph attention network. In *Artificial Neural Networks and Machine Learning–ICANN 2021: 30th International Conference on Artificial Neural Networks, Bratislava, Slovakia, September 14–17, 2021, Proceedings, Part I 30*, pages 253–264. Springer, 2021.
- [180] Guillaume Bono. *Deep multi-agent reinforcement learning for dynamic and stochastic vehicle routing problems*. PhD thesis, Université de Lyon, 2020.
- [181] Rafael Basso, Balázs Kulcsár, and Ivan Sanchez-Diaz. Electric vehicle routing problem with machine learning for energy prediction. *Transportation Research Part B: Methodological*, 145:24–55, 2021.
- [182] Martin L Puterman. Markov decision processes. *Handbooks in operations research and management science*, 2:331–434, 1990.
- [183] Juan Zhang, James F Campbell, Donald C Sweeney II, and Andrea C Hupman. Energy consumption models for delivery drones: A comparison and assessment. *Transportation Research Part D: Transport and Environment*, 90:102668, 2021.
- [184] Thomas Kirschstein. Comparison of energy demands of drone-based and ground-based parcel delivery services. *Transportation Research Part D: Transport and Environment*, 78:102209, 2020.
- [185] Jack W Langelaan, Sven Schmitz, Jose Palacios, and Ralph D Lorenz. Energetics of rotary-wing exploration of titan. In *2017 IEEE Aerospace Conference*, pages 1–11. IEEE, 2017.
- [186] Kimon P Valavanis and George J Vachtsevanos. *Handbook of unmanned aerial vehicles*, volume 1. Springer, 2015.
- [187] Gary L Johnson. *Wind energy systems*. Citeseer, 1985.

- [188] Charles R Chalk, TP Neal, TM Harris, Francis E Pritchard, and Robert J Woodcock. Background information and user guide for mil-f-8785b (asg), 'military specification-flying qualities of piloted airplanes'. Technical report, Cornell Aeronautical Lab Inc Buffalo Ny, 1969.
- [189] Md Shamim Hussain, Mohammed J Zaki, and Dharmashankar Subramanian. Edge-augmented graph transformers: Global self-attention is enough for graphs. *arXiv preprint arXiv:2108.03348*, 2021.
- [190] Elias Khalil, Hanjun Dai, Yuyu Zhang, Bistra Dilkina, and Le Song. Learning combinatorial optimization algorithms over graphs. *Advances in neural information processing systems*, 30, 2017.
- [191] Alex Nowak, Soledad Villar, Afonso S Bandeira, and Joan Bruna. A note on learning algorithms for quadratic assignment with graph neural networks. *stat*, 1050:22, 2017.
- [192] Tugrul U Singireddy, Shiva Ram Reddy and Daim. Technology roadmap: Drone delivery–amazon prime air. In *Infrastructure and Technology Management*, pages 387—412. Springer, 2018.
- [193] Jarotwan Koiwanit. Analysis of environmental impacts of drone delivery on an online shopping system. *Advances in Climate Change Research*, 9(3):201–207, 2018.
- [194] Fabiano Kümmel Heller. *Technological innovation applied to walmart and tesco's supply chain*. PhD thesis, 2017.
- [195] Marie Paul Nisingizwe, Pacifique Ndishimye, Katare Swaibu, Ladislav Nshimiyimana, Prosper Karame, Valentine Dushimiyimana, Jean Pierre Musabyimana, Clarisse Musanabaganwa, Sabin Nsanzimana, and Michael R Law. Effect of unmanned aerial vehicle (drone) delivery on blood product delivery time and wastage in rwanda: a retrospective, cross-sectional study and time series analysis. *The Lancet Global Health*, 10(4):e564–e569, 2022.
- [196] Zipline. Zipline keynote 2023: The best delivery experience not on earth.
- [197] Kevin Dorling, Jordan Heinrichs, Geoffrey G. Messier, and Sebastian Magierowski. Vehicle routing problems for drone delivery. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 47:70–85, 2017.
- [198] Insu Hong, Michael Kuby, and Alan T. Murray. A range-restricted recharging station coverage model for drone delivery service planning. *Transportation Research Part C: Emerging Technologies*, 90:198–212, 2018.
- [199] Jean Louis Fendji Kedieng Ebongue, Israel Bayaola, Christopher Thron, and Anna Förster. Charging stations placement in drone path planning for large space surveillance. In *CARI 2020 - Colloque Africain sur la Recherche en Informatique et en Mathématiques Appliquées*, Thies, Senegal, October 2020.

- [200] Parikshit Maini, Kaarthik Sundar, Mandeep Singh, Sivakumar Rathinam, and PB Sujit. Cooperative aerial-ground vehicle route planning with fuel constraints for coverage applications. *IEEE Transactions on Aerospace and Electronic Systems*, 55(6):3016–3028, 2019.
- [201] Amila Thibbotuwawa, Peter Nielsen, Banaszak Zbigniew, and Grzegorz Bocewicz. Factors affecting energy consumption of unmanned aerial vehicles: An analysis of how energy consumption changes in relation to uav routing. In Jerzy Świątek, Leszek Borzemski, and Zofia Wilimowska, editors, *Information Systems Architecture and Technology: Proceedings of 39th International Conference on Information Systems Architecture and Technology – ISAT 2018*, pages 228–238, Cham, 2019. Springer International Publishing.
- [202] Nicolas Bono Rossello, Renzo Fabrizio Carpio, Andrea Gasparri, and Emanuele Garone. Information-driven path planning for uav with limited autonomy in large-scale field monitoring. *IEEE Transactions on Automation Science and Engineering*, pages 1–11, 2021.
- [203] Yongyong Wei and Rong Zheng. Informative path planning for mobile sensing with reinforcement learning. *Proceedings - IEEE INFOCOM*, 2020-July:864–873, 2020.
- [204] Zhilong Liu, Raja Sengupta, and Alex Kurzhanskiy. A power consumption model for multi-rotor small unmanned aircraft systems. *2017 International Conference on Unmanned Aircraft Systems, ICUAS 2017*, pages 310–315, 2017.
- [205] Joshua K. Stolaroff, Constantine Samaras, Emma R. O’Neill, Alia Lubers, Alexandra S. Mitchell, and Daniel Ceperley. Energy use and life cycle greenhouse gas emissions of drones for commercial package delivery. *Nature Communications*, 9(1):1–13, 2018.
- [206] Jack W. Langelaan, Sven Schmitz, Jose Palacios, and Ralph D. Lorenz. Energetics of rotary-wing exploration of Titan. *IEEE Aerospace Conference Proceedings*, 2017.
- [207] Thomas Kirschstein. Comparison of energy demands of drone-based and ground-based parcel delivery services. *Transportation Research Part D: Transport and Environment*, 78(December 2019):102209, 2020.
- [208] Alex S. Praselia, Rong Jong Wai, Yi Lun Wen, and Yu Kai Wang. Mission-based energy consumption prediction of multirotor uav. *IEEE Access*, 7:33055–33063, 2019.
- [209] Ty Nguyen and Tsz Chiu Au. Extending the range of delivery drones by exploratory learning of energy models. *Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS*, 3:1658–1660, 2017.

- [210] Arnav Choudhry, Brady Moon, Jay Patrikar, Constantine Samaras, and Sebastian Scherer. CVaR-based Flight Energy Risk Assessment for Multirotor UAVs using a Deep Energy Model. 2021.
- [211] Thiago A. Rodrigues, Jay Patrikar, Arnav Choudhry, Jacob Feldgoise, Vaibhav Arcot, Aradhana Gahlaut, Sophia Lau, Brady Moon, Bastian Wagner, H. Scott Matthews, Sebastian Scherer, and Constantine Samaras. In-flight positional and energy use data set of a DJI Matrice 100 quadcopter for small package delivery. *Scientific Data*, 8(1):6–13, 2021.
- [212] Jianfa Wu, Honglun Wang, Yu Huang, Zikang Su, and Menghua Zhang. Energy management strategy for solar-powered uav long-endurance target tracking. *IEEE Transactions on Aerospace and Electronic Systems*, 55(4):1878–1891, 2018.
- [213] Mason Thammawichai, Sujit P Baliyarasimhuni, Eric C Kerrigan, and Joao B Sousa. Optimizing communication and computation for multi-uav information gathering applications. *IEEE Transactions on Aerospace and Electronic Systems*, 54(2):601–615, 2017.
- [214] Uihwan Choi, Chaehwan Moon, and Jaemyung Ahn. Energy minimization of dynamic multi-uav communication network for cooperative multi-hop data gathering. *IEEE Transactions on Aerospace and Electronic Systems*, 2022.
- [215] Amila Thibbotuwawa, Grzegorz Bocewicz, Peter Nielsen, and Banaszak Zbigniew. Planning deliveries with uav routing under weather forecast and energy consumption constraints. *IFAC-PapersOnLine*, 52(13):820–825, 2019.
- [216] R Tyrrell Rockafellar, Stanislav Uryasev, et al. Optimization of conditional value-at-risk. *Journal of risk*, 2:21–42, 2000.
- [217] Hoong Chuin Lau, William Yeoh, Pradeep Varakantham, Duc Thien Nguyen, and Huaxing Chen. Dynamic stochastic orienteering problems for risk-aware applications. *Uncertainty in Artificial Intelligence - Proceedings of the 28th Conference, UAI 2012*, pages 448–458, 2012.
- [218] Angel A. Juan, Javier Faulin, Scott E. Grasman, Markus Rabe, and Gonalo Figueira. A review of simheuristics: Extending metaheuristics to deal with stochastic combinatorial optimization problems. *Operations Research Perspectives*, 2:62–72, 2015.
- [219] D. E. Mazzuco, A. M. Carreiro Danielli, D. L. Oliveira, P. P.P. Santos, M. M. Pereira, L. C. Coelho, and E. M. Frazzon. A concept for simulation-based optimization in Vehicle Routing Problems. *IFAC-PapersOnLine*, 51(11):1720–1725, 2018.
- [220] Thomas Mayer, Tobias Uhlig, and Oliver Rose. An open-source discrete event simulator for rich vehicle routing problems. *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, pages 1305–1310, 2016.

-
- [221] Edward Farhan, Mohammed and Gohre, Brett and Junprung. Reinforcement learning in anylogic simulation models: a guiding example using pathmind. *2020 Winter Simulation Conference (WSC)*, 4(5):3212—3223, 2020.
 - [222] Markus Rabe, Felix Dross, and Alexander Wuttke. Combining a Discrete-event Simulation Model of a Logistics Network with Deep Reinforcement Learning. pages 765–774, 2017.
 - [223] James J.Q. Yu and Albert Y.S. Lam. Autonomous Vehicle Logistic System: Joint Routing and Charging Strategy. *IEEE Transactions on Intelligent Transportation Systems*, 19(7):2175–2187, 2018.
 - [224] Andrew R. Runnalls. Kullback-Leibler approach to Gaussian mixture reduction. *IEEE Transactions on Aerospace and Electronic Systems*, 43(3):989–999, 2007.
 - [225] Gary L Johnson. *Wind energy systems*. Citeseer, 1985.
 - [226] Ruifan Liu, Hyo-Sang Shin, Minguk Seo, and Antonios Tsourdos. Delivery route planning for unmanned aerial system in presence of recharging stations. In *AIAA AVIATION 2022 Forum*, page 4028, 2022.

