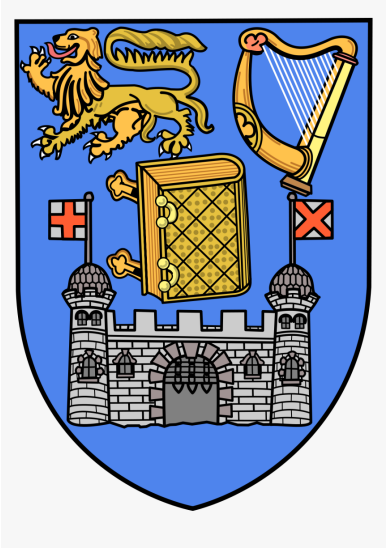
The University of Dublin, Trinity College



Optimising Energy Efficiency in UAV-Assisted Networks using Multi-Agent Reinforcement Learning

Author:

Babatunji Omoniwa

Supervisor:

Prof. Ivana Dusparic

Co-Supervisor:

Dr. Boris Galkin

A thesis submitted in fulfillment of the requirements for

the degree of Doctor of Philosophy (Computer Science) in the

School of Computer Science & Statistics

Trinity College Dublin, The University of Dublin

June 2023

Declaration

I declare that this thesis has not been submitted as an exercise for a degree at this or any other university and it is entirely my own work.

I agree to deposit this thesis in the University's open access institutional repository or allow the library to do so on my behalf, subject to Irish Copyright Legislation and Trinity College Library conditions of use and acknowledgement.

Babatunji Omoniwa

June 2023

Abstract

The demand for cellular connectivity continues to witness unprecedented growth over the years. Unmanned Aerial Vehicles (UAVs) equipped with small cells can provide ubiquitous connectivity to static and mobile ground users in situations of increased network demand or points of failure in existing terrestrial cellular infrastructure. We consider a system called a UAV-assisted network that uses UAVs to serve ground users. However, UAVs deplete energy while hovering in the sky and providing coverage for extended periods of time. Furthermore, multiple UAVs sharing the same frequency spectrum and deployed to provide wireless connectivity to users in a given area may experience a decrease in the system's energy efficiency (EE) due to interference from neighbouring UAV cells or other access points.

Recent approaches focus on optimising the system's EE by optimising the trajectory of UAVs serving only static ground users and neglecting mobile users. Several others neglect the impact of interference from nearby UAV cells, assuming an interference-free network environment. Furthermore, some works assume global spatial knowledge of ground users' location via a central controller (CC) that periodically scans the network perimeter and provides real-time updates to the UAVs for decision-making. However, this assumption may be unsuitable in disaster scenarios since it requires significant information exchange between the UAVs and CC. Moreover, it may not be possible to track users' locations in a disaster scenario. Despite growing research interest in decentralised control over centralised UAVs' control, collaboration among UAVs to improve the systems' EE has not been adequately explored. In dynamic environments with changing users' distribution, it is challenging to track users in real-time without apriori knowledge of the users' distribution or gaining such insight from a CC.

This thesis' main contribution, the Decentralised Multi-Agent Reinforcement Learning (DMARL), allows each UAV equipped with an autonomous agent to intelligently serve ground users while improving the overall system's EE. The DMARL attempts to improve the total system's energy efficiency while providing wireless connectivity to ground users in an interference-limited network environment. Thus, we address this by decomposing the DMARL into five variants. The first variant investigates how multiple UAVs, each with an independent learning agent learn a policy that improves the total system's energy efficiency while serving static and mobile ground users without the knowledge of the users' locations from a CC. An agent-controlled UAV can have a wider view of its environment by gaining more knowledge for better decisions when information is exchanged with closest neighbours. Therefore, we propose two modes of collaboration, an indirect and a direct variant (variants 2 and 3, respectively), to improve the system's EE in a shared, dynamic and interference-limited network environment. The direct collaboration allows UAVs to share their data via existing 3GPP guidelines, while the indirect variant has no such mechanism but implicitly reflects this knowledge in its reward formulation as an incentive towards collaborative behaviours. More importantly, the past coverage performance of UAVs may influence their decision to collaborate while serving users in dense and uneven users' distribution. Lastly, we propose direct and indirect collaborative variants that allow UAVs to be density-aware by collaborating to intelligently serve densely distributed users (variants 4 and 5, respectively).

We perform evaluations under different network configurations. Results show that our DMARL outperforms centralised baselines that assume prior global knowledge of ground users' location in terms of EE by as much as 80%. When compared to our closest decentralised MARL baseline which neglects the impact of interference when serving pedestrians, we discover that collaboration provides improved systems' EE by as much as 55% – 75%. In city traffic, motorways and national roads, the DMARL outperforms state-of-the-art MARL approaches which do not account for varying users' densities in terms of EE by as much as 65% – 98%. These findings demonstrate the effectiveness of our approach in providing UAVs deployed in an environment with the intelligence to provide coverage in an energy-efficient manner.

Acknowledgements

This thesis was made possible thanks to the support of alot of people. First, I would like to thank my supervisor, Professor Ivana Dusparic, for her guidance through her extensive expertise in this field and for all the valuable advice in the research process. She has been instrumental to this achievement, and without whom my studies in Ireland wouldn't have been possible. I would like to thank my co-supervisor, Dr Boris Galkin for his immense encouragement and the helpful insight gotten through his wealth of knowledge. I specially appreciate Professor Siobhan Clarke, the head of the ENABLE project and grant holder for her financial support during my Ph.D. studies. I would like to thank my examiners, Professor Dirk Pesch and Professor Melanie Bourouche, for their valuable feedback.

Many thanks to all SATORI members in CONNECT for their help and support. A big thanks to past and present members of ENABLE, CONNECT and the Distributed Systems Group who were supportive in diverse ways, they include in no particular order, Emma, Lassane, Maxime, Pat, Christian, Jose, Aqeel, Saqib, Paul, Jose, Mohit, Alberto, Andrea, Evelyn, Erika, Tochukwu, JB.

To my lovely wife, Janet, for her love and patience during my studies. Special thanks to my precious boys, Jason and Jamin, who had to endure my long absence. To my dad, mum, siblings and friends, I love you all.

Babatunji Omoniwa

University of Dublin, Trinity College

June 2023

Publications

- Babatunji Omoniwa, Boris Galkin and Ivana Dusparic, "Density-Aware Deep Reinforcement Learning to Optimise Energy Efficiency in UAV-Assisted Networks," *2023 19th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob),* 2023.

- Babatunji Omoniwa, Boris Galkin and Ivana Dusparic, "Communication-enabled deep reinforcement learning to optimise energy-efficiency in UAV-assisted networks," *Vehicular Communications,* Vol. 43, 2023. https://doi.org/10.1016/j.vehcom.2023.100640.

- Babatunji Omoniwa, Boris Galkin and Ivana Dusparic, "Optimising Energy Efficiency in UAV-Assisted Networks using Deep Reinforcement Learning," *IEEE Wireless Communications Letters,* vol. 11, no. 8, pp. 1590-1594, Aug. 2022. doi: 10.1109/LWC.2022.3167568.

- Boris Galkin, Babatunji Omoniwa and Ivana Dusparic, "Multi-Agent Deep Reinforcement Learning For Optimising Energy Efficiency of Fixed-Wing UAV Cellular Access Points," *ICC 2022 - IEEE International Conference on Communications,* 2022, pp. 1-6, doi: 10.1109/ICC45855.2022.9838809.

- Babatunji Omoniwa, Boris Galkin and Ivana Dusparic,"Energy-Aware Optimization of UAV Base Stations Placement via Decentralized Multi-Agent Q-Learning," *2022 IEEE 19th Annual Consumer Communications & Networking Conference (CCNC),* Jan. 2022, pp. 216-222.

- Babatunji Omoniwa, Maxime Guériau and Ivana Dusparic, "An RL-based Approach to Improve Communication Performance and Energy Utilization in Fog-based IoT," *2019 15th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob),* 2019, pp. 324-329.

Contents

List of Figures

List of Tables

List of Abbreviations

2D	Two-Dimensional
3D	Three-Dimensional
AI	Artificial Intelligence
ANN	Artificial Neural Network
AP	Access Point
BS	Base Station
CAVs	Connected and Autonomous Vehicles
CC	Central Controller
CMAD	Communication-enabled Multi-Agent Decentralised
CTDE	Centralised Training and Decentralised Execution
CDR	Connected users to Deployed users Ratio
D2D	Device to Device
DAMAD	Density-Aware Multi-Agent Decentralised
DACEMAD	Density-Aware Communication-Enabled Multi-Agent Decentralised
DDQN	Double Deep Q-Network
DDPG	Deep Deterministic Policy Gradient
DQLSI	Decentralised Q-learning with Local Sensory Information
DQN	Deep Q-Network
DTDE	Decentralised Training and Decentralised Execution
DRL	Deep Reinforcement Learning
EE	Energy Efficiency
FCD	Floating Car Data
GA	Genetic Algorithm
GPS	Global Positioning System
IL	Independent Learner

IoT	Internet-of-Things
LoS	Line-of-Sight
LSTM	Long Short-Term Memory
MDP	Markov Decision Process
MAD	Multi-Agent Decentralised
MAS	Multi-Agent System
MADDPG	Multi-Agent DDPG
MARL	Multi-Agent Reinforcement Learning
MIMO	Multiple-Input and Multiple-Output
MMDP	Multi-agent MDP
NN	Neural Network
NP-complete	Non-deterministic Polynomial-time complete
P-complete	Polynomial-time complete
POMDP	Partially Observable Markov Decision Process
PSO	Particle Swarm Optimisation
QL	Q-Learning
ReLu	Rectified Linear unit
RL	Reinforcement Learning
RMSprop	Root Mean Squared Propagation
SINR	Signal to Interference plus Noise Ratio
SNR	Signal to Noise Ratio
SUMO	Simulation of Urban MObility
UAV	Unmanned Aerial Vehicle
VANETs	Vehicular Ad-hoc Networks
WSN	Wireless Sensor Networks

List of Notations

U	Set of quad-rotor UAVs
ξ	Set of ground users
x_j, y_j, h_j	3D coordinates of UAV j
$x_{min}, y_{min}, h_{min}$	Minimum 3D coordinates
$x_{max}, y_{max}, h_{max}$	Maximum 3D coordinates
γ	Signal-to-interference-plus-noise-ratio
β	Attenuation factor
α	Path loss exponent
P	Transmit power of UAVs
d	distance
χ_{int}	Set of interfering UAVs
d_{col}	Minimal collision distance
T_{range}	Transmission range
σ^2	Power of the additive white Gaussian noise
B_w	Channel bandwidth
$\mathfrak{R}_{i,j}^t$	Achievable data rate
e_P	Propulsion energy
e_C	Communication energy
e_T	Total energy
δ_t	Duration of time step
κ_0	Coefficient of blade profile power
κ_1	Coefficient of induced power
κ_2	Coefficient of parasite power
U_{tip}	Rotor blade's tip speed
v_0	Mean hovering velocity

$\mathcal{R}_j^t$	Agent j's reward function at time t
C_j^t	Agent j's connectivity score at time t
N_d^t	Set of distances of neighbouring UAVs
f_j^t	Jain's fairness index
e_j^t	Agent j's energy level at time t
$\mho$	Agent's collaborative factor
D_s	Dimension of input state space
D_a	Dimension of action space
s, s'	State, Next state
a	Action
r	Reward signal
K	Number of hidden layers
W_k	Number nodes in k^{th} layer
N	Number of UAVs
N_e	Number of learning episodes
N_d	Set of neighbouring UAVs distances
T	Number of time steps
θ, θ^-	Parameters of neural network
$L(\theta)$	Loss function
π_j	Policy of Agent j
η	Energy efficiency
E	Bits representing each observation
$U_L(t)_j$	Number of neighbours of agent-controlled UAV j at time t

Chapter 1

Introduction

The demand for cellular connectivity continues to sky-rocket, with a projected 2.4-fold growth, from 6.1 billion in 2018 to 14.7 billion by 2023, in device connections [Cisco, 2018, as cited in [Camps-Mur et al., 2021]]. Unmanned Aerial Vehicles[1] (UAVs) equipped with small cells (also known as miniaturised radio access points (APs) or aerial base stations), can play an important role in supporting the Internet-of-Things (IoT) networks [Omoniwa et al., 2019] by providing seamless connectivity to ground users, who may be static or mobile. In particular, the adoption of UAVs to provide wireless connectivity to ground users in events of increased network load or points-of-failure in existing terrestrial cellular infrastructure has attracted the attention of the telecommunications sector, as well as the research community [Mozaffari et al., 2019]. In this thesis, we consider a system called a UAV-assisted network that uses UAVs to serve ground users. However, these UAVs have limited on-board battery capacity and deplete energy while they hover in the sky and provide coverage for extended periods of time [Galkin et al., 2019a, Mozaffari et al., 2017]. Furthermore, multiple UAVs deployed to provide wireless connectivity to users in a given area may experience a decrease in the system's energy efficiency (EE) due to interference from neighbouring UAV cells or other APs sharing the same frequency spectrum [Challita et al., 2019, Galkin et al., 2022a]. EE is an important metric used to measure how effectively energy is utilised to achieve a desired outcome, such as, improving the total throughput in the network.

[1] A UAV could be human-controlled with a ground pilot, or fully-autonomous [Galkin, 2021].

Despite recent research efforts of deploying UAVs as aerial base stations [Liu et al., 2019a, Liu et al., 2020, Liu et al., 2018, Wang et al., 2021], optimising the total system's EE of UAVs serving dynamic users in an interference-limited network environment has not been adequately explored. This research aims to investigate a Decentralised Multi-Agent Reinforcement Learning (DMARL) approach to optimising the total system's EE of UAVs serving ground users in a shared, dynamic and interference-limited network environment. This chapter provides an introduction to the study by first discussing the motivation and context, followed by the challenges and gaps, then the research aim and questions, thesis contributions, assumptions, and lastly, the structure of the thesis.

1.1 Motivation

An Unmanned Aerial Vehicle (UAV)[2], also known as a drone, is any flying machine that does not require an onboard pilot. UAVs have numerous real-world applications, ranging from assisted communication in disaster-affected areas to surveillance, deliveries and logistics, search and rescue operations [Mozaffari et al., 2019]. In particular, UAVs can be flexibly deployed to provide wireless connectivity to mobile users in out-of-coverage areas, complementing and lowering the cost of deploying terrestrial cellular infrastructures. Furthermore, UAVs may be deployed in situations of sudden service fluctuations in cellular users' demand, i.e., network load, or service outage due to disasters [Galkin, 2021]. For example, UAVs were deployed in Puerto Rico in 2017 to provide emergency cellular service to ground users after Hurricane Maria [Galkin, 2021]. However, it is challenging to provide ubiquitous network connectivity to users in dynamic network environments characterised by changing the density of users caused by the spatial and temporal variations due to the mobility and traffic situation in a geographical area [Marini et al., 2022].

The deployment of UAVs to provide wireless connectivity to ground users is gaining significant research attention [Galkin et al., 2022a]. We consider a use-case of UAVs providing wireless connectivity to ground users who do not have other wireless connectivity due to a

[2]In this thesis, we refer to multi-rotor types of UAVs and not fixed-winged UAVs unless otherwise stated. Multi-rotor UAVs have the ability for vertical take-off and their design has superior manoeuvrability over a fixed-wing (airplane) or an aerostatic (balloon) design [Galkin, 2019].

possible failure or outage in the existing mobile communication network. Our scenario considers a network of connected UAVs where the UAVs may be interconnected to each other via existing wireless technologies (e.g. WiFi, 4G/5G) while having a dedicated back-haul connection to the core network via satellite or cellular infrastructures [Cicek et al., 2020]. We understand that satellite or cellular infrastructures may be inaccessible to ground users. This inaccessibility of the ground users to such infrastructures may be due to long separation or obstacles that hinder effective service delivery to the ground users [Fotouhi et al., 2019]. As such, we propose that such infrastructures may serve as a back-haul to the UAVs (who are currently covering a hole in the cellular network). For example, an area having ground users may be out of coverage due to failures of close-by cellular infrastructures, thereby resulting in an outage. However, the failure may not have affected further cellular infrastructures that are too far away from the ground users that have a limited transmission range. In such scenarios, the far-away cellular infrastructures may back-haul the UAVs which are deployed to serve in these emergencies. Nevertheless, our research does not focus on optimising the existing back-haul connection link [Fotouhi et al., 2019] rather we focus on the interaction between the UAVs without having any dedicated central controller, and the UAV to ground users communication.

To derive the full benefit of UAV deployments, recent researchers have focused on addressing some main challenges, such as, the 3D trajectory optimisation [Liu et al., 2019a, Lyu et al., 2017], energy efficiency (EE) optimisation [Liu et al., 2020], energy consumption minimization [Zeng et al., 2019], and coverage optimisation [Wang et al., 2021, Liu et al., 2020]. As energy-constrained UAVs fly in the sky, they may encounter interference from nearby UAV cells or other access points sharing the same frequency band[3], thereby affecting the system's EE [Galkin et al., 2022a]. Several research contributions have been made to optimise the EE of UAVs deployed to serve ground users, however, many of such works neglect the impact of interference on the system's performance.

EE is an important metric in UAV-assisted networks for several reasons. As such, we present some conceptual reasons to use EE as a metric throughout this thesis. UAVs have limited

[3]This is a term used in telecommunications for a range of frequencies defined and dedicated to a specific type of service or radio technology.

energy resources, making it difficult for UAVs to keep flying in the sky. Several researchers have proposed energy optimisation techniques to improve the flight duration [cite]. While energy optimisation seeks to address only how the UAVs minimise their energy consumption, EE optimisation seeks to minimise the energy consumption of UAVs while delivering on the outcome of task assigned. Energy-efficient UAV-assisted networks can enable longer endurance while providing extended coverage capabilities [Mozaffari et al., 2019]. Furthermore, energy-efficient operations can lead to significant cost savings in the network, that is, in minimising the battery costs, maintenance expenses, and overall operational costs. Several other works focus on only maximising the throughput in UAV-assisted networks [Hayat et al., 2016]. Optimising the total system throughput without considering the need of optimising the energy consumption may not be desirable in energy-constrained wireless networks as this. Hence, EE is a crucial metric in UAV-assisted networks for reducing the energy consumed by UAVs deployed to provide wireless connectivity to users on the ground.

Compared with a traditional terrestrial cellular communication network, channel modelling for an airborne, UAV-assisted wireless system is more challenging due to the mobility and direct line-of-sight (LoS) communication link from nearby UAVs [Zhang et al., 2022]. Crucially, UAVs require robust strategies to provide ubiquitous wireless coverage to ground users in a dynamic network environment. Unlike previous work that assumes global spatial knowledge of ground users' location through a central controller that periodically scans the network perimeter and provides real-time updates to the UAVs for decision-making, we focus on a decentralised approach suitable in emergency scenarios where there may be service downtime due to failure in the controller, or loss of UAVs' control packets due to an unreliable wireless channel or traffic congestion in the network [Challita et al., 2019]. In particular, it may be unfeasible for UAVs to periodically wait for control packets from the central server before executing an action in a disaster scenario. For instance, a UAV needs to react spontaneously to an observed change in its environment. Furthermore, the growth in the number of deployed UAVs may pose a different kind of challenge, making centralised management difficult. As such, it becomes imperative to de-emphasize methods that focus on human control or centralised control of UAVs. Moreover, in such scenarios it is difficult to keep track of the

location of all ground users in real time.

On this note, there has been a shift towards the decentralised control of UAVs, with recent research adopting disruptive machine learning techniques to solve complex optimisation problems in UAV-assisted networks [Liu et al., 2020, Hu et al., 2020, Wang et al., 2021]. Machine learning techniques adopt the concept of an "*agent*", which is an independent entity or software program installed on a host to allow for interaction with its immediate environment, by perceiving its surroundings through sensors, then acting via actuators [Dorri et al., 2018]. Over the years, there has been growing research interest towards agent-based control in UAV-assisted networks [Liu et al., 2019a, Galkin et al., 2022a, Liu et al., 2020], with each agent-based design serving some specific functions. A centrally-controlled actor-critic algorithm was proposed in [Samir et al., 2021] to optimise the trajectories of UAVs while maximising the coverage of vehicles in an interference-free environment. However, as the number of UAVs in the network increases, it may become impractical for effective decision-making and control in disaster scenarios. Multi-agent learning is challenging in itself, requiring agents to learn their policies while taking into account the consequences of the actions of others. The decentralized Multi-Agent Deep Deterministic Policy Gradient (MADDPG) approach proposed in [Liu et al., 2020, Wang et al., 2021] was an improvement to the centralized learning approach in [Liu et al., 2018], where all agents are controlled by a single actor-critic network. Although these approaches [Liu et al., 2020, Liu et al., 2018, Wang et al., 2021] focus on optimising the systems' EE while serving static pedestrian users, they did not account for the interference from neighbouring UAV cells. UAVs may require robust strategies to optimise their flight trajectory while providing coverage to ground users in a dynamic environment. Multi-Agent Reinforcement Learning (MARL) has been shown to perform well in decision-making tasks in such a dynamic environment [Liu et al., 2020, Wang et al., 2021, Liu et al., 2019a]. To improve the performance of the decentralised control, several methods have been studied [Busoniu et al., 2006, Tan, 1993, Kim et al., 2019b].

The decentralised control of UAVs comes with its challenges, of which one remains the collaboration challenge. The problem of collaboration, where agents jointly work towards improving global performance, has received considerable research attention [Dafoe et al., 2020]. The

terms collaboration, cooperation, and coordination are related but distinct terms used in existing MARL literature. We provide a breakdown of the differences between these concepts:

Collaboration: This refers to the act of agents working together towards a common goal or task. In MARL, collaboration involves agents sharing information, coordinating their actions, and learning from each other to achieve optimal outcomes [Lesser, 1999, Panait and Luke, 2005]. Collaborative behavior in MARL can lead to emergent strategies and behaviors that were not explicitly programmed [Stone et al., 2010].

Cooperation: This refers to the willingness of agents to work together and contribute towards a common goal. It involves agents making decisions that benefit both themselves and the collective group [Jiang et al., 2018]. Cooperation is about agents aligning their actions to maximise the collective reward.

Coordination: This involves agents synchronizing their actions and behaviors to achieve a desired outcome. It focuses on ensuring that agents' actions are complementary and do not conflict with each other [Boutilier, 1999]. Coordination can be achieved through communication, where agents exchange information about their observations, intentions, and plans [Pesce and Montana, 2019].

In this thesis, we choose the term *'collaboration'* since it involves agent-controlled UAVs actively working together towards a common goal. Collaboration among artificial intelligence (AI)-powered UAVs is crucial and has not received adequate research attention. We note that robust strategies are required to allow for seamless collaboration among UAVs while jointly executing their tasks.

In this thesis, we adopt a DMARL approach and propose five variants of this approach to maximise the total system's EE by optimising the trajectory of each UAV, the energy consumed and the number of connected static and mobile ground users over a series of time steps, while taking into account the impact of interference from nearby UAV cells.

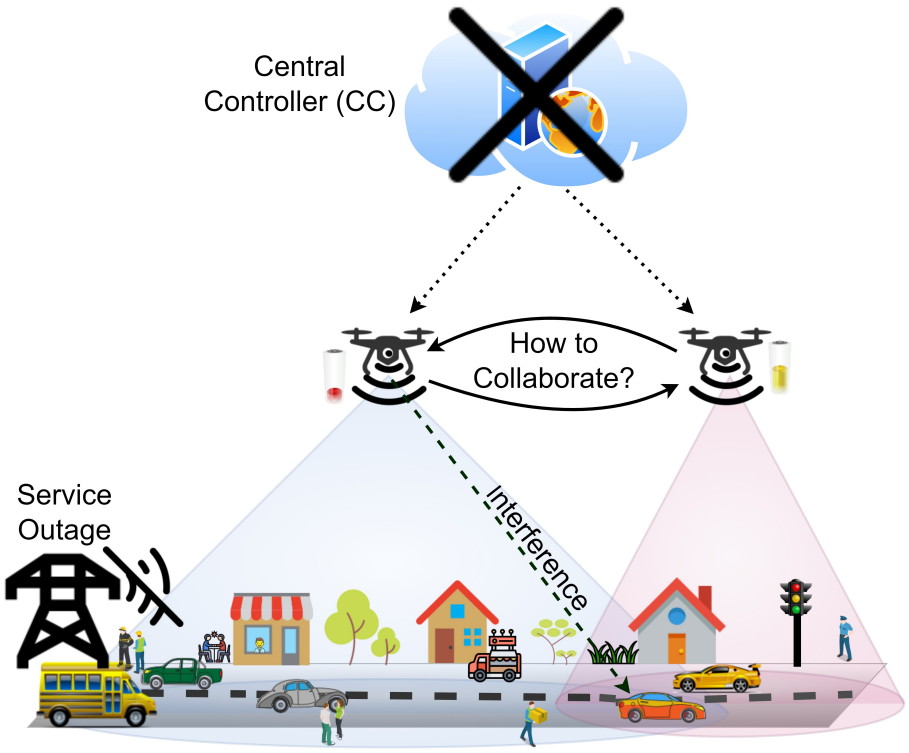


Figure 1.1: UAVs providing coverage to ground users in a shared, dynamic and interference-limited environment.

1.2 Challenges

There are several challenges with deploying UAVs as aerial base stations to provide connectivity to ground users. Figure 1.1 shows the network architecture considered in this thesis. It shows the deployment of multiple UAVs to serve ground users during a service downtime and in the absence of a central controller (CC). The energy-constrained UAVs are expected to perform the coverage task (providing wireless connectivity to users) without having prior knowledge of the spatial locations of the users via a CC in this interference-limited environment. We outline the challenges faced in our scenario as follows:

Challenge 1: Lack of Apriori Knowledge of Ground Users' Location

The assumption of a centralised entity with global spatial knowledge of the ground location of users may be unrealistic in typical disaster scenarios. The work [Galkin et al., 2016] and [Liu et al., 2019a] rely on a CC that partitions the ground users into

different clusters using a K-means algorithm and assigns a single UAV to serve each cluster. In [Islam et al., 2022], the CC is used to predict the future distribution of vehicles using a long short-term memory (LSTM) neural network. However, a failure in the CC may result in downtime of network service. Furthermore, the reliance on a CC may result in increased periodic updates between the UAVs and the CC. Nevertheless, a decentralised UAV control may suffice in emergencies by eliminating a potential single point of failure in the network. However, it is challenging to provide coverage to users without having knowledge of their locations.

Challenge 2: Mobility of Ground Users

Several works limit their investigation to UAVs serving static ground users [Mozaffari et al., 2017, Liu et al., 2020, Liu et al., 2018, Wang et al., 2021, Galkin et al., 2022b]. Although, it is easier for UAVs to serve static users than mobile users such as, pedestrians and vehicles, the multi-UAV deployment problem[4] in itself maps to an NP-hard problem [Sanchez-Aguero et al., 2020, Liu et al., 2019a]. The authors in [Mozaffari et al., 2017] proposed an iterative algorithm to minimise the energy consumption of UAVs serving as aerial base stations to uniformly distributed static sensors. In [Ruan et al., 2018], a game-theoretic approach was proposed to maximise the system's EE while maximising the ground area covered by the UAVs irrespective of the presence of ground users. In [Liu et al., 2020], a deep reinforcement learning (DRL) approach was presented to jointly optimise the system's EE and wireless coverage of static ground users. Mobility may bring about uncertainty in UAV networks, and makes it challenging for the UAVs to serve mobile ground users in real time. Hence, approaches to optimise the UAVs' trajectories while serving ground users must be adaptive to the mobility of the users.

Challenge 3: Interference from nearby UAV cells

As UAVs fly in the sky, they may encounter interference[5] from nearby UAV cells or

[4]This involves deploying multiple UAVs to perform a coverage task.

[5]Interference is a phenomenon that occurs when the wireless communication signals are disrupted or weakened by the presence of other wireless signals.

other APs sharing the same frequency band. In a typical shared wireless environment like this, managing interference can be challenging [Warrier et al., 2022, Challita et al., 2019]. To reduce the complexity in their system design, previous work [Liu et al., 2020, Liu et al., 2018, Wang et al., 2021, Liu et al., 2019a] did not consider the impact of interference. Interference may bring constraints to the environment, making it difficult for UAVs to discover the best set of actions to execute in this shared environment. More importantly, if interference is not effectively managed, it may hinder a UAV from providing coverage in an energy-efficient manner since the interference experienced from neighbouring UAV cells may lead to a decrease in the total throughput in the network which adversely impacts on the system's EE [Challita et al., 2019].

Challenge 4: Conservation of UAVs' Energy during Flight

It is challenging to conserve the energy of UAVs during prolonged coverage tasks, considering their limited onboard battery capacity. UAVs may deplete energy during propulsion for flying and hovering and during communication [Mozaffari et al., 2017]. To avoid UAVs dropping from the sky when they run out of battery power, it is important to optimise each UAV's flight trajectory while minimising energy consumption during coverage tasks. Some research focuses on the placement of UAVs without considering the UAVs' energy usage while manoeuvring in the sky [Islam et al., 2022, Hanna et al., 2019].

Challenge 5: UAVs' collaboration to accomplish coverage tasks in a dynamic environment

The deployment of multiple UAVs in our shared and interference-limited environment can make the environment exhibit non-stationarity [Panait and Luke, 2005] since the state of the environment is not a function of the singular action of a UAV. Rather the state of the environment is a consequence of the actions of other UAVs in the network. From the perspective of a single UAV, the presence of other UAVs in this shared environment makes it non-stationary and dynamic. Notwithstanding the non-stationarity from the competing actions of neighbouring UAVs, the issue is further worsened by the

presence of mobile ground users requiring coverage in the environment. As such, it is challenging for UAVs to collaborate among themselves to provide ubiquitous coverage to ground users. For instance, the collaborative coverage game presented in [Ruan et al., 2018] where the UAVs' individual decision mutually influences that of other UAVs is known to be NP-hard. Several approaches consider only uniformly distributed ground users in geographically-confined areas [Mozaffari et al., 2017, Liu et al., 2020, Liu et al., 2018, Liu et al., 2019a]. These approaches performed well in reasonably even densities of users but might not perform as well with an uneven distribution where some areas are denser than others, i.e., in an event scenario with a concentration of users, or mostly in vehicular scenarios where users are congregated in the road space, in particular congested road space. As such, there is a need to investigate collaborative approaches that allow UAVs to be density-aware, capable of collaborating and providing coverage intelligently in such a scenario.

1.3 Research Questions

Motivated by the challenges and research gaps identified in Section 1.2, this thesis aims to provide answers to the research questions. With the overarching research question, "Can UAVs deployed to provide wireless connectivity to mobile ground users improve the total system's energy efficiency in a shared, dynamic and interference-limited network environment?", we provide more specific questions as follows:

RQ1: Can UAVs serving mobile ground users improve the total system's energy efficiency in a shared, dynamic and interference-limited network environment without relying on a central controller for decision-making?

RQ2: Can collaboration with closest neighbours improve the total system's energy efficiency while minimising the total energy consumed by UAVs in a shared, dynamic and interference-limited network environment?

RQ3: Can UAVs collaborate intelligently to improve the total system's energy efficiency in highly mobile, dense and unevenly distributed users in an urban environment?

With the outlined research questions, we present the contribution of this thesis.

1.4 Thesis Contribution

The contribution of this thesis is a Decentralised Multi-Agent Reinforcement Learning (DMARL) approach to optimise the total EE of multiple UAVs serving ground users by jointly optimising the flight trajectory of each UAV, the energy consumed by the UAVs and the number of connected ground users in a shared, dynamic and interference-limited network environment and under a strict energy budget. Table 1.1 provides a summary of our contribution. This study attempts to address the research gaps by proffering answers to the research questions highlighted in Section 1.3 and in doing so provides the following contributions.

- **C1:** Current MARL approaches rely on a CC to pre-partition the coverage region and provide real-time periodic updates to the UAVs for decision-making. These approaches may not be suitable in disasters due to significant communication overhead between the CC and UAVs. Moreover, damages and failure in certain parts of the network infrastructure may make it difficult for the CC to keep track of the ground locations of mobile users in emergencies. Critically, a failure in the CC or its control packets may impact the service operation of the UAVs. Our study is one of the first that considers a fully-decentralised MARL with UAVs deployed to serve mobile ground users without having to rely on a central entity to gain knowledge of the users' location. This thesis proposes a variant of DMARL, called the Decentralized Q-learning with Local Sensory Information[6] (DQLSI), which equips each UAV with an *autonomous and independent learning agent* that interacts with its environment to improve the total energy efficiency of UAVs in the network by jointly maximising the number of connected ground users and energy utilisation of UAVs without any feedback from a CC. Our proposed DQLSI algorithm assumes that each agent-controlled UAV has local observability[7] for decision making while serving ground users in disaster scenarios,

[6]Local sensory information refers to the sensory input that an agent perceives from its immediate environment through its sensors. The rest of our proposed algorithms also have this property.

[7]Local observability refers to the ability of an RL agent to execute decisions based on only the information it perceives from its environment.

and is particularly suitable when there is limited or service outage existing cellular infrastructures. Nevertheless, we assume the existence of a dedicated back-haul to provide connectivity for the deployed UAVs to the core network. We evaluated the DQLSI variant of DMARL against centralised approaches.

- **C2:** Interference from neighbouring UAV cells impacts negatively the total EE of multiple UAVs serving ground users. Interference may bring about *non-stationarity* to the environment, making it difficult for UAVs to collaborate while serving dynamic ground users. The term *non-stationarity* refers to the issue of policy changes in agent-controlled UAVs during the learning process. The policies of a UAV in the shared network may adversely affect the performance of other UAVs operating in that same environment. In an attempt for agent-controlled UAVs to improve their individual performance during the learning process, the UAVs may exhibit selfish behaviours that impact the performance of others via interference. Hence, it is desirable for UAVs to collaborate to improve the overall network performance in a shared and dynamic network environment. Current MARL approaches neglect the impact of interference and do not have a mechanism to enhance collaboration among UAVs. This thesis proposes DMARL variants that allow agents to collaborate indirectly using the Multi-Agent Decentralised Double Deep Q–Network (MAD–DDQN) and directly using the Communication-enabled MAD–DDQN (CMAD–DDQN) variant. The CMAD–DDQN extends the MAD–DDQN with a communication mechanism, to improve the total EE of multiple UAVs serving ground users in a shared, dynamic and interference-limited network environment. However, the performance improvement of the CMAD–DDQN over MAD–DDQN comes at some communication overhead cost. In both variants, we design each agent's reward to reflect the coverage performance locally. Our formulation uses a neighbour collaborative factor that gives agent-controlled UAVs an incentive to collaborate. We evaluated the MAD–DDQN and CMAD–DDQN variants of DMARL against a state-of-the-art decentralised multi-agent deep deterministic policy gradient algorithm [Liu et al., 2020].

- **C3:** Current MARL approaches applied to optimise the users' coverage worked well in reasonably even densities but might not perform as well with an uneven distribution where some areas are denser than others, i.e., in an event scenario with a concentration of users, or mostly in vehicular scenarios where users are congregated in the road space, in particular congested road space. The DMARL addresses the issue of how UAVs can collaborate to provide coverage to dense areas of the network, i.e., we look into what information should be exchanged among UAVs to make them density-aware while improving the total EE of multiple UAVs serving ground users. This thesis proposes variants that optimise the UAVs' trajectory towards the dense areas in the network. The density-aware MAD–DDQN and density-aware CMAD-DDQN variants were extended from the MAD-DDQN and CMAD-DDQN variants, respectively. The density-aware CMAD-DDQN allows each agent directly share its best neighbour connectivity score, best-experienced connectivity score and the position where it experienced the best number of connected users to keep track of dense users' areas in the network. On the other hand, the density-aware MAD-DDQN has no direct communication mechanism but provides agents with a motivation to collaborate which is reflected in its weighted reward formulation. Details of our collaborative variants can be found in Chapter 4 of this thesis. Our proposed density-aware variants maximise the total EE of multiple UAVs serving ground users while jointly optimising each UAV's trajectory, the number of connected users, and the energy consumption by UAVs in an interference-limited network environment. We evaluated the density-aware variants of DMARL against a state-of-the-art decentralised MARL approach [Liu et al., 2020].

1.5 Assumptions of Study

In this thesis, we make certain assumptions.

1. Each UAV serves as a small cell AP and is equipped with radio equipment such as antennas and a radio transceiver[8] for the transmission and reception of wireless signals.

[8]The transceiver units are used by the UAVs to communicate with each other, ground users or with a ground control station (GCS).

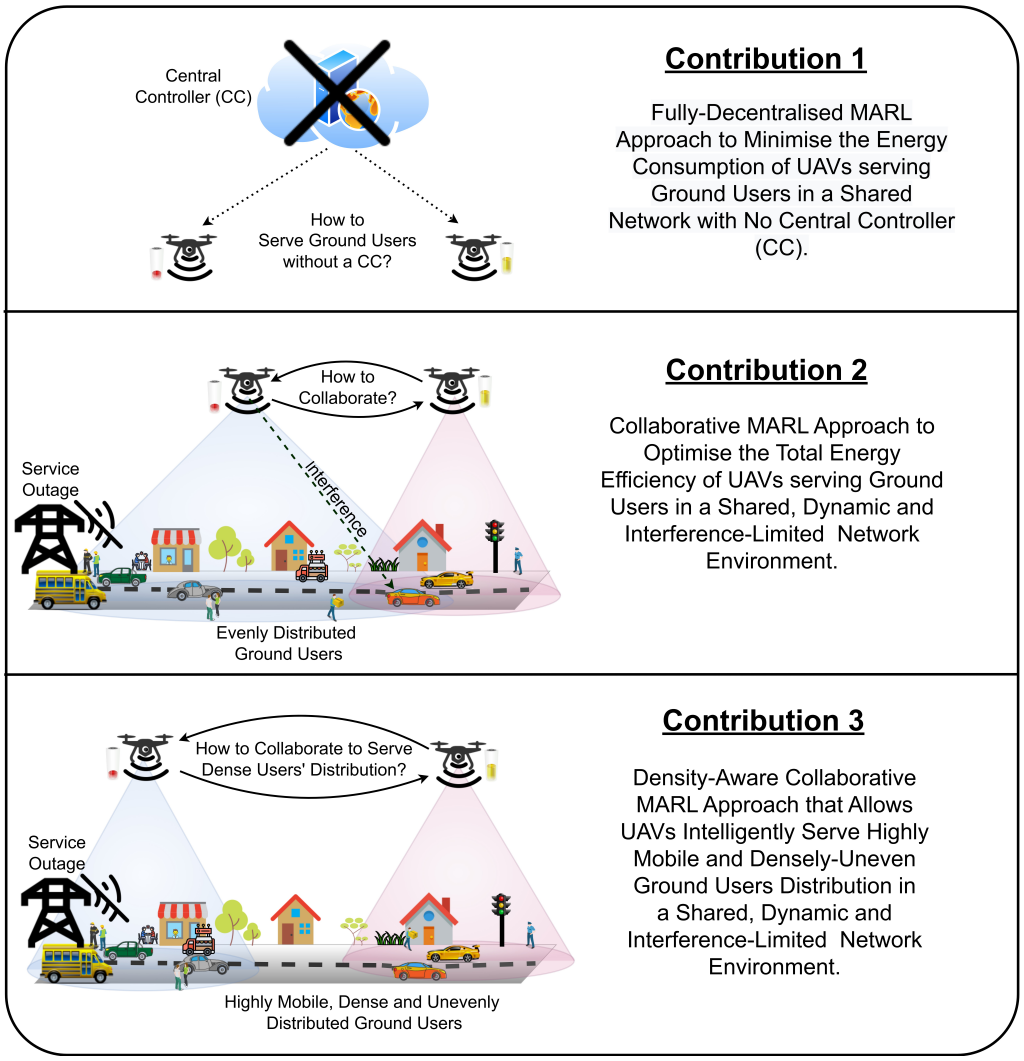


Figure 1.2: Thesis contributions

A Global Positioning System (GPS) for navigation and infrared obstacle sensors for flight safety are also mounted on each UAV [Galkin, 2019]. The UAV is equipped with two sets of antennas, one for communicating with ground users, and another for communicating with other UAVs and the wireless back-haul[9] to the network. We also assume perfect communication channel with the back-haul.

2. We assume a decentralised architecture of multiple UAVs deployed to serve ground users, where each UAV is equipped with an RL agent that drives the decision-making procedure. Furthermore, we assume that no central controller exist in the network due to damages caused by a disaster. The network of UAVs serving ground users is assumed to be decentralised, meaning that the agent-controlled UAVs execute actions

[9]Backhaul refers to the link between the core network and sub-networks existing.

Table 1.1: Summary of Contributions

Contribution	**Decentralised Multi-Agent Reinforcement Learning**				
	Independent Learning Agent	Indirect Collaborative Agent	Direct Collaborative Agent	Density-Aware Indirect Collaborative Agent	Density-Aware Direct Collaborative Agent
Fully Autonomous/No CC	✓	✓	✓	✓	✓
Interference-Limited Channel	✓	✓	✓	✓	✓
User Mobility	✓	✓	✓	✓	✓
Agent's Architecture	Tabular	Deep NN	Deep NN	Deep NN	Deep NN
Dense-Aware (Section 6.6) (Urban Traffic Density)	NA	✓ (Low Density)	✓ (Low Density)	✓ (High Density)	✓ (High Density)
Collaborative (Approach)	✓ (Broadcast)	✓ (Indirect)	✓ (Direct)	✓ (Indirect)	✓ (Direct)
Communication-Enabled	✗	✗	✓	✗	✓

independently of any centralised control.

3. We assume that each RL agent that controls a UAV is a "*local agent*". A *local agent* receives a local observation and executes an action that yields some reward based on its local performance. An agent-controlled UAV is a UAV that is controlled by an agent architecture. A "*neighbour*" of an agent-controlled UAV is defined as a node that is within the communication range of the agent-controlled UAV. Specifically, the neighbours of each agent-controlled UAV can differ at each time step owing to the mobility of nodes. We explicitly assume that each agent-controlled UAV is capable of interacting with other UAVs within its communication range.

4. UAVs are energy-constrained, with a limited energy budget to perform their coverage task. We assume that each UAV is equipped with a Lithium Polymer battery. This is a rechargeable type of battery that uses a polymer electrolyte rather than a liquid electrolyte. They are known to be more efficient and safe for use, however, they run down during prolonged use. UAVs deplete energy when they are in operation, making the total energy consumption equal to the sum of propulsion energy e_P to enable its flight and communication energy e_C consumption for wireless signal processing and data transmission. The communication energy is practically much smaller than the propulsion energy, i.e., $e_C \ll e_P$ [Eom et al., 2020, Zeng and Zhang, 2017]. Hence in this work we consider the energy consumed due to propulsion, and ignore energy consumption from the circuits for signal processing such as channel decoders and analogue-to-digital

converters.

5. Ground users can be static or mobile (pedestrians or vehicles). We assume that the initial deployment locations of the UAVs are determined beforehand based on mission objectives. In our evaluation, we assume that UAVs are deployed to provide coverage to ground users such as static and mobile pedestrians in a 1 km^2 area of some selected areas of Dublin, Ireland. We consider the deployment of UAVs to serve vehicles in a 3 km^2 Dublin city centre urban area, a 7 km motorway and 6.5 km national road in Dublin, Ireland using SUMO that mimics the traffic conditions in the environment.

6. The aerial positioning of UAVs in the sky creates strong LoS channels with neighbouring UAV cells which in some circumstances deteriorates the overall network performance [Galkin, 2019]. We assume that the UAV that is closest to a user will be the UAV with the strongest received signal power; hence, the user will always be serviced by the closest UAV, and other UAVs which are beyond the serving UAV distance will act as interferers [Galkin et al., 2019b]. The performance of each UAV is impaired by interference from nearby UAV cells. Unlike recent work that does not consider the impact of interference from nearby UAV cells by considering the networks to be strictly noise-limited, we take into account the impact of interference on the system's performance.

7. In circumstances where UAVs communicate with neighbours, we do not consider delayed or lossy communication, which we understand may be a source of additional complexity. We hope to account for this in our future work.

1.6 Thesis Structure

The remainder of this thesis is organised as follows:

Chapter 2 – Related Work

This chapter presents some RL concepts to better understand the proposed DMARL approach. We provide an overview of UAVs' control strategies as used in recent works and present a review of state-of-the-art works on the deployment of multiple autonomous UAVs

which are related to our use-case scenario.

Chapter 3 – Multi-UAV Model Design

In this chapter we describe the system model used throughout this thesis, and formulate the multi-UAV deployment problem to maximise the total system's EE by jointly optimising its 3D trajectory, number of connected users, and the UAVs' energy consumed while deployed to serve ground users under a strict energy budget.

Chapter 4 – Decentralised Multi-Agent Reinforcement Learning (DMARL) for UAV-Assisted Networks

This chapter introduces the main contribution of this thesis. First, we present a set of design requirements for the DMARL for UAV-assisted networks. We then proceed to the design of the DMARL for UAV-assisted networks, mapping our thesis contributions to meet the design requirements while addressing the research questions.

Chapter 5 – Implementation

In this chapter, we implement the DMARL for UAV-assisted networks.

Chapter 6 – Evaluation

In this chapter, we present the evaluation objectives, and the metrics used and outline the baseline approaches. We then present the results and discuss our findings.

Chapter 7 – Conclusion & Open Questions

This chapter concludes this thesis. It discusses contributions and findings, and future work.

Chapter 2

Related Work

2.1 Introduction

In the previous chapter, we provide motivation for Reinforcement Learning (RL)-based control for UAVs to provide wireless connectivity to mobile users in a shared, dynamic and interference-limited network environment. In this chapter, we discuss in detail RL, in order to provide the necessary background for understanding the Decentralised Multi-Agent Reinforcement Learning (DMARL). We review related work that deploys autonomous UAVs to serve ground users and the approaches adopted. We focus on decentralised RL-based multi-UAV control with a particular focus on UAVs serving as aerial base station applications. The taxonomy of this chapter is given in Figure 2.1.

2.2 Reinforcement Learning

Reinforcement Learning (RL) is a machine learning technique that allows an agent[1] to learn by trial and error via interaction with its environment [Sutton and Barto, 2018, Busoniu et al., 2008]. Figure 2.2 shows the interaction between an RL agent and its environment. The interaction of an agent with its environment is formalised using a Markov decision process (MDP) [Watkins and Dayan, 1992, Sutton and Barto, 2018]. The MDP can be defined as

[1] An agent is a software program or entity whose obligation is to learn and make decisions [Sutton and Barto, 2018].

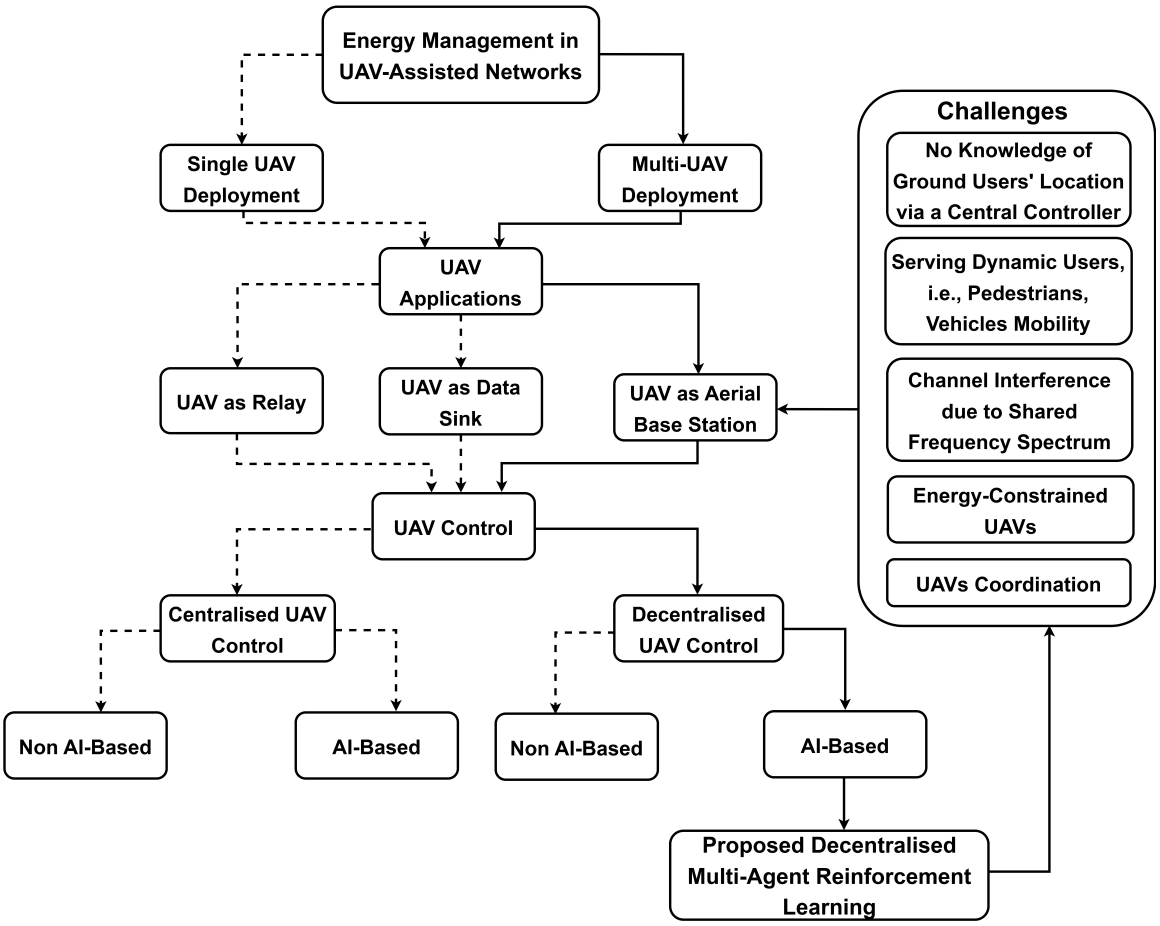
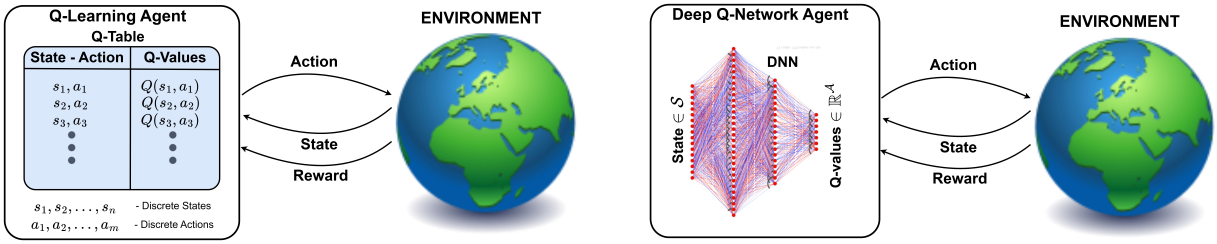


Figure 2.1: Taxonomy of Study.

a tuple, $\langle\, \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma \,\rangle$, where $\mathcal{S}$ represents a finite set of states. $\mathcal{A}$ represents a finite set of actions. The transition function $\mathcal{P} : \mathcal{S} \times \mathcal{A} \to \delta(\mathcal{S})$ represents the transition probability from state $s \in \mathcal{S}$ to $s' \in \mathcal{S}$ given the action $a \in \mathcal{A}$. The reward function $\mathcal{R} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$ defines the reward the agent receives for transiting from state $s \in \mathcal{S}$ to $s' \in \mathcal{S}$ after executing an action $a \in \mathcal{A}$. The discount factor $\gamma \in [0,\ 1]$ balances the trade-off between immediate and future rewards. MDPs are widely used models to obtain optimal decisions in single-agent, fully-observable environments [Sutton and Barto, 2018]. Solving an MDP[2] will yield a policy $\pi : \mathcal{S} \to \mathcal{A}$, which maps states to actions. An optimal policy π^* is one that maximises the expected discounted sum of rewards. At each time step, an RL agent observes the state of the environment and takes an action that changes the state of the environment. For each such action, the agent receives a reward signal. The goal of an RL

[2]The complexity of the MDP is in the worst-case P–Complete [Amato et al., 2013].



(a) Q-learning. (b) Deep Q-network.

Figure 2.2: Reinforcement learning with the *(a)* Q-learning agent and *(b)* deep Q-network agent interacting with the environment.

agent is to learn good policies for sequential decision problems, by optimizing a cumulative future reward signal [Sutton and Barto, 2018]. Figure 2.2 shows RL agents' interaction with the environment. Conventional single-agent RL algorithms like Q-learning [Watkins and Dayan, 1992], deep Q-networks [Mnih et al., 2015], double deep Q-networks [Hasselt et al., 2016] can be directly applied to the multi-agent settings. However, when more than one agent is deployed in a shared environment, the system dynamics change and will no longer be a function of the action of a single agent [Papoudakis et al., 2020]. The presence of other agents acting in it may make the environment non-stationary. Nevertheless, this non-stationarity may be addressed if agents have mechanisms for collaboration [Panait and Luke, 2005, Papoudakis et al., 2019, Papoudakis et al., 2020]. Next, we look at notable RL approaches used in literature.

2.2.1 Q-Learning

Q-learning was introduced by Chris Watkins in 1989 [Watkins and Dayan, 1992]. It was designed for stationary, single-agent, fully-observable environments with discrete states and actions. Q-learning is a model-free, off-policy algorithm [Sutton and Barto, 2018] that learns the value of executing an action a in a state s as seen in Figure 2.2a. Like other off-policy RL algorithms, it can learn from data collected by any behavioural policy without requiring a model of the environment. The Q-learning agent uses any policy to estimate Q that maximizes the future reward. The Q-learning update for agent j is given as,

$$Q_j(s_j, a_j) \leftarrow (1-\alpha)Q_j(s_j, a_j) + \alpha\Big[r_j + \gamma \max_{a'_j} Q_i(s'_j, a'_j)\Big], \tag{2.1}$$

where s_j is the present local state observed by agent j, s'_j is the new local state observed by agent j, a_j is the action taken by agent j, r_j is the reward received by agent j in that time step, α is the learning rate and $\gamma \in [0, 1]$ is the discount factor. As stated in [Watkins and Dayan, 1992], Q-learning is a primitive form of learning but serves as a basis for more sophisticated designs. Next, we look into how Artificial Neural Networks (ANNs) can be integrated with RL as function approximators to estimate the Q-value while exploring the environment using the ϵ–greedy policy.

2.2.2 Deep Q-Network (DQN)

A novel variant of Q-learning called deep Q-network (DQN), was introduced by Mnih et al. (2015). The DQN as shown in Figure 2.2b is a combination of reinforcement learning with a class of artificial neural networks known as deep neural networks. A DQN is a multi-layered neural network (NN) that for a given input state $s \in \mathcal{S}$ yields an output vector of Q-values $Q(s, a; \theta)$ corresponding to each executable action, where θ are the parameters of the network. For an $\mathcal{S}$-dimensional state space and an action space containing $\mathcal{A}$ set of actions, the NN is a function from $\mathbb{R}^{\mathcal{S}}$ to $\mathbb{R}^{\mathcal{A}}$ [Hasselt et al., 2016]. DQN attempts to address some instabilities caused by the correlations present in the sequence of observations and the fact that small updates to Q may significantly change the policy and therefore change the data distribution, and the correlations between the action-values (Q) and the target values $r + \gamma \max_{a'} Q(s', a')$. First, a biologically inspired mechanism called experience replay was introduced to randomise the data, hence removing correlations in the observation sequence and smoothing over changes in the data distribution. Second, an iterative update was used to adjust the action-values towards target values which are updated periodically, thereby reducing correlations with the target. The target used by DQN is given as,

$$y_j = r_j + \gamma \max_{a'_j} Q(s'_j, a'_j; \theta^{(t)}) \tag{2.2}$$

During the learning process, DQN minimises the error estimated by the loss function by optimising the weights θ. The loss is measured as the difference between the predicted and

target Q-value. Given as,

$$L(\theta) = \big(y_j - Q(s_j, a_j; \theta)\big)^2, \tag{2.3}$$

where y_j is the target Q-value, $Q(s_j, a_j; \theta)$ is the predicted Q-values. The parameter $\theta^{(t)}$ is updated once every T_{target} time-steps by letting $\theta^{(t)} = \theta$.

2.2.3 Double Deep Q-Network (DDQN)

Despite breakthroughs in traditional Q-learning and DQN, Hasselt et al. (2016) in their work noted the challenge of overestimation caused by noise in the environment which degrades the performance of the algorithm when tested on the *Asterix* and *Wizard of Wor* games. The max operator given in Equation (2.1) and Equation (2.2), which uses the same values both to select an action and also to evaluate the action does not address the overestimation problem [François-Lavet et al., 2018]. Hence, [Hasselt et al., 2016] introduced the Double DQN (DDQN) to address this challenge by decoupling the selection from the evaluation. Like the DQN, the DDQN algorithm is model-free since it solves the RL task directly using generated samples, without explicitly estimating the reward and transition dynamics $\mathcal{P}(r, s'|s, a)$. The algorithm is also off-policy since it learns about the greedy policy $a = \arg\max_{a'} Q(s, a'; \theta)$ while following a behaviour distribution that ensures adequate exploration of the state space [Mnih et al., 2015]. DDQN evaluates the greedy policy according to the online network but uses the target network to estimate its value. This is achieved by learning two value functions by assigning random experiences to update one of the two value functions, hereby leading in two sets of weights, θ and θ^-. For each update, one set of weights is used to determine the greedy policy and the other for evaluation. For clarity and comparison, we rewrite the DQN target in Equation (2.2) as,

$$y_j = r_j + \gamma Q\Big(s', \arg\max_{a'_j} Q(s'_j, a'_j; \theta), \theta\Big), \tag{2.4}$$

while the target used by DDQN is given as,

$$y_j = r_j + \gamma Q_{(2)}\Big(s', \arg\max_{a'_j} Q_{(1)}(s'_j, a'_j; \theta), \theta^-\Big). \tag{2.5}$$

It can be observed in the case of the DQN that the action selection and evaluation is done using the weights θ, while the DDQN selects the greedy policy according to the current values, using weights θ and fairly evaluates the value of the policy using weights θ^{-}. In this thesis, we adopt the DDQN agent architecture and apply this in our multi-agent setting. Nevertheless, researchers have proposed the deep deterministic policy gradient agent algorithm [Lillicrap et al., 2015] to solve challenging problems.

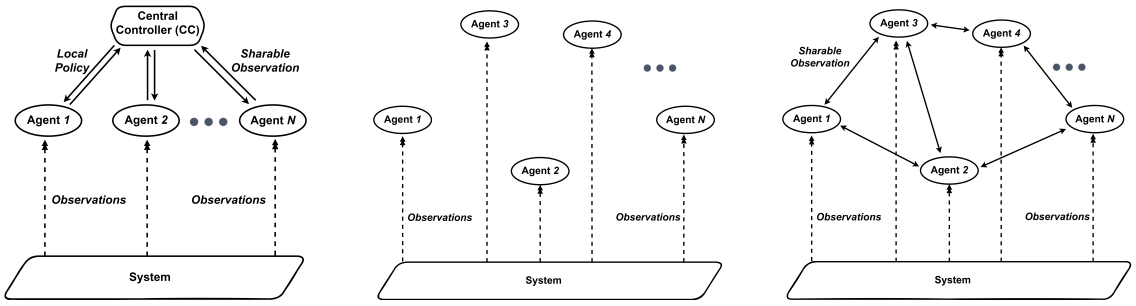
2.2.4 Deep Deterministic Policy Gradient (DDPG)

The deep deterministic policy gradient (DDPG) algorithm is a model-free, online, off-policy reinforcement learning method. A DDPG agent is an actor-critic RL agent that searches for an optimal policy that maximizes the expected cumulative long-term reward. The DDPG was first introduced in [Lillicrap et al., 2015] to solve problems with high-dimensional continuous observation and action spaces. In particular, it is most notably used in physical control tasks. However, a major drawback of learning in continuous action spaces is exploration [Lillicrap et al., 2015]. The DDPG is implemented using two sets of actor-critic networks, making a total of four NNs: a Q network, a deterministic policy network, a target Q network, and a target policy network. The target networks are time-delayed copies of their original networks that slowly track the learned networks, thus they help to improve training stability. The algorithm uses noisy perturbations for exploration by the actor network, specifically an *Ornstein-Uhlenbeck* process for generating noise, sampling the noise from a correlated normal distribution [Lillicrap et al., 2015]. A random mini-batch of experiences is sampled from the set of experiences (s, a, r, s') stored in the replay buffer. This mini-batch of experiences is then used to update the networks [Lillicrap et al., 2015, Algorithm 1]. Next, we introduce a system where multiple agents co-exist in the same environment.

2.2.5 Multi-Agent System

A Multi-Agent System (MAS) is a group of autonomous, interacting entities called agents that share a common environment, which they perceive with sensors and upon which they act with actuators [Busoniu et al., 2008]. MASs has gained grounds in the area of robotics and drone

networks [Liu et al., 2020], intelligent transport systems [Guériau and Dusparic, 2020], energy distribution [Dusparic et al., 2015], and the analysis of social dilemmas [Canese et al., 2021]. With these advances, research strides have been made towards extending existing single-agent RL algorithms to multi-agent approaches. However, direct implementation of single-agent RL to multiple agents may not guarantee convergence due to non-stationarity. Non-stationarity occurs when multiple agents are learning concurrently in the same environment, thereby making the true values of the agent's actions change over time [Sutton and Barto, 2018]. Figure 2.3 shows three representative information architectures commonly used in MAS research [Zhang et al., 2021a]. Since different architectures may suit specific problems, it is crucial to investigate Multi-agent reinforcement learning (MARL) algorithms in accordance with measures that provide the agents with sufficient information for better decision-making. In the next section, we discuss MARL as it relates to current approaches.



(a) Centralised setting. (b) Fully decentralised setting. (c) Decentralised setting with networked agents.

Figure 2.3: Three representative information architecture in MARL. Specifically, in *(a)*, there exists a central controller (CC) that may be responsible for both disseminating local policies to each agent and aggregating information from the agents, for example, joint actions, joint rewards, and joint observations. In both *(b)* and *(c)*, we have decentralised architecture with no CC. In *(b)*, the agents are fully decentralised, with no explicit information exchange with each other. Rather, each *independent learning agent* makes decisions based on its local observations, without any collaboration and/or aggregation of data from neighbours or CC. In *(c)*, agents are connected via a possibly time-varying communication network, so that the local information can spread across the network, by information exchange with only each agent's neighbours. *(c)* is more common in collaborative MARL settings.

2.2.6 Multi-Agent Reinforcement Learning

The work [Busoniu et al., 2006] classified MARL algorithms based on the type of task they addressed, namely: fully collaborative, fully competitive, and mixed (neither collaborative

nor competitive). In this thesis, we consider a fully-collaborative setting in which distributed agents have an aligned goal of improving the overall system performance [Panait and Luke, 2005]. Nevertheless, the presence of a central controller[3] (CC) as seen in Figure 2.3a reduces the problem to a MDP with a joint action space [Busoniu et al., 2006]. The architecture shown in Figure 2.3b shows a set of independent learning (IL) agents that assume other agents as part of the environment. Several real-world problems can be solved more effectively using a collaborative approach where different agents collaborate to achieve common goals. Specifically, collaboration among multiple agents has been extensively studied [Papoudakis et al., 2020]. According to [Amato et al., 2013, Busoniu et al., 2006], these collaborative approaches provide robustness to individual agent failures and are known to be more scalable to complex, long-duration missions. For example, it could be of particular benefit to surveillance, disaster mitigation and extra-terrestrial operations. The decentralised architecture as seen in Figures 2.3b and 2.3c via parallel computation can help speed up the learning in MARL [Busoniu et al., 2006]. Furthermore, it is often unrealistic to assume the existence of an all-knowing central agent for computing optimal policies. At any given time in such an environment, an agent may not have a full observability [Panait and Luke, 2005]. A ubiquitous, instantaneous, and lossless communication available to all agents can allow them to have access to all observations at each time step thereby ensuring speedy learning by agents [Oliehoek and Spaan, 2012]. However, communication often comes at a cost, thus, requiring agents to strategise on what and when to communicate [Amato et al., 2013]. On this note, it becomes crucial to consider a choice of a MARL algorithm to use based on suitability and purpose. However, MARL comes with several challenges, such as *computational complexity, non-stationarity, partial observability,* and *credit assignment*, which require robust, less-complex and adaptive learning algorithm designs to cope with real-world problems.

2.2.6.1 Computational Complexity

A vast majority of RL problems have low sample efficiency, which requires an agent to interact with its environment for a longer duration in order to learn a useful policy [Wong et al., 2022].

[3] A central entity used to monitor and control a set of machines.

For instance, it can take an RL agent tens of thousands of trial samples to learn certain games [Mnih et al., 2015], which humans can master after dozens of trials. The sample complexity is how large a training set is required in order to learn a good approximation to the target, while the computational complexity is how much computation is required to manipulate a training set and output an approximation to the target [Kakade, 2003]. The sample complexity may worsen with the presence of multiple interacting agents since an agent will require more training samples to learn a good policy. Multi-agent problems are known to deal with high computational demands, and the higher the number of agents, the more demanding it is on computing power [Wong et al., 2022]. In particular, continuous-space MARL problems are noted for slow learning of new tasks and, in the worst case, may fail to master such tasks [van Hasselt, 2012]. The manner in which a MARL problem is modelled may to a large extent determine its complexity [Amato et al., 2013]. Nevertheless, research effort has been made towards developing or modifying existing MARL algorithms to ensure that they are sample and computationally efficient [Wong et al., 2022], which could help in speeding up the learning of new tasks. However, non-stationarity may be a culprit to computational complexity, where agents may face difficulty adapting to the changing policies of other agents.

2.2.6.2 Non-Stationarity

In a typical multi-agent setting, agents learn and interact with the environment simultaneously. Due to the presence of other agents with changing policies acting in the same environment, the state transitions and reward may no longer be stationary [Wong et al., 2022]. A non-stationary environment is one that changes during learning which may prevent the agents from converging to stationary policies [Terry et al., 2020]. Recent research [Phan et al., 2021, Wong et al., 2022] has attributed such non-stationarity caused by simultaneously learning agents to a violation of the Markov assumption. According to [Marinescu, 2016], non-stationarity in MARL can arise for two reasons:

(a) *Agent-induced factor*, where the impact of multiple agents simultaneously acting within the same environment will result in their collective actions being non-deterministic since

the environment responds differently for each combination of actions.

(b) *Environment-induced factor*, where the environment itself evolves continually over time, hence resulting in an agent's action being affected by the evolving environment.

Several researchers have proposed different methods to address non-stationarity. They include:

(i) Enhancing collaboration among agents to hasten the learning process [Ranjan Kumar and Varakantham, 2020, Phan et al., 2021, Lesser, 1999].

(ii) Re-modelling the problem to classify other agents are part of the environment [Wong et al., 2022, Zhang et al., 2021a, Zhang et al., 2018].

(iii) Communication may be used [Terry et al., 2020], and different training agents can exchange information about their observations, actions and intentions to stabilize their training [Papoudakis et al., 2019]. From Figure 2.3c, we see a set of networked agents able to share some observations. Nevertheless, we understand that communication is a function of the communication range and distance between the communicating entities.

(iv) Centralised Training and Decentralised Execution (CTDE) [Lowe et al., 2017, Zhang et al., 2018] where each agent is provided with the other agents' information during the centralised training phase while allowed to act independently based on its individual policies during the decentralised execution phase [Li et al., 2022].

(v) Decentralised Training and Decentralised Execution (DTDE) allows each agent to learn policies that can generalise the policies played out in its environment [Kim et al., 2019b, Papoudakis et al., 2019, Foerster et al., 2017]. In particular, collaborative DTDE-based approaches may help in addressing non-stationarity in fully decentralised environments [Gronauer and Diepold, 2022].

Some MARL problems may exhibit non-stationarity due to a combination of both agent-induced and environment-induced factors, thereby requiring robust MARL strategies [Marinescu, 2016]. Moreover, a decentralised MARL architecture may often face the partial observability challenge, where agents have access only to local observations from the environment.

This partial information aggravates the issues caused by non-stationarity [Zhang et al., 2021a], hereby requiring a combination of the above methods.

2.2.6.3 Partial Observability

Partial observability refers to when an agent has no access to the actual state of the environment but has access only to its local observations, thereby significantly impacting the training performance [Papoudakis et al., 2019]. An agent may be oblivious to the actions and rewards of other agents in the environment, making it difficult to attribute a change in the environment to its individual action. In decentralised setups, communication among agents may address this challenge by reducing the complexity of finding good policies [Amato et al., 2013]. For instance, an agent exploring different parts of the environment may share its observations to mitigate partial observability [Wong et al., 2022]. Nevertheless, many real-world problems (e.g., autonomous driving, UAV-assisted networks, game playing) involve multiple agents with partial observability and limited communication, thereby making it difficult to generate accurate models for these domains due to complex interactions between agents and the environment [Omidshafiei et al., 2017]. Despite several research efforts to address partial observability in MARL, the question of "what observations should be shared among agents to improve the learning performance?" demands further investigations.

2.2.6.4 Credit Assignment

Credit assignment refers to the problem of measuring an action's influence on future rewards, which often arises when individual agents cannot view their contribution to the global reward [Wong et al., 2022]. One crucial challenge of credit assignment is the process of mapping immediate actions to the rewards that they influence in the future [Taylor, 2015]. For example, in a football team, it can be difficult to determine whether or not winning the game was affected by the practice session the team had the previous day, the lucky boots worn by some team members, or the kind of food some players ate that morning. Reward shaping is one of the most intuitive and effective solutions to credit assignment [Zou et al., 2019], with a goal to shape the originally delayed rewards to effectively reward or penalize intermediate actions.

Although reward shaping may enhance collaborative behaviours in agents, it requires careful formulation, which is hard and often manually done [Wu et al., 2021]. On the other hand, it was stated in [Mannion et al., 2018] that carelessly applying reward shaping has previously been shown to alter an agent's original goals.

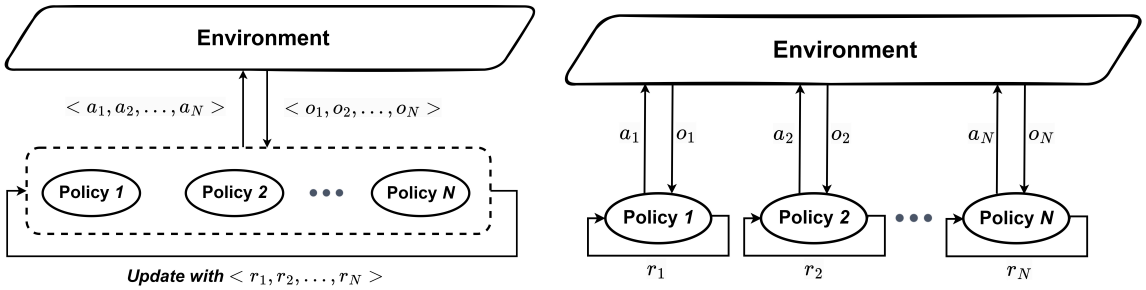
There exist two typical reward functions for MARL [Mannion et al., 2018]:

(a) *Local reward*, which is unique to each agent. This type of reward answers the question "how do an agent's actions contribute to a system that involves the actions of many agents?". The local reward is based on the utility of the part of a system that agent j can observe directly. However, individual agents may become self-interested, selfishly seeking to maximise their local reward signal, often at the expense of global system performance [Mannion et al., 2018, Devlin et al., 2014].

(b) *Global reward*, which reflects the overall performance. This type of reward answers the question of "how do all agent's actions contribute to a global system performance?". The global reward may encourage all agents to act in the system's interest. However, since all agents receive the same reward signal, regardless of whether their actions improved the system performance, it may encourage undesirable behaviour among some agent [Mannion et al., 2018, Devlin et al., 2014].

The work [Wu et al., 2021] argued that the shared global reward may lead to the *lazy agent problem* in MARL, where it is difficult to attribute an agent's contribution to the global performance. The *lazy agent problem* can be addressed by assigning an individual reward to each agent [Wu et al., 2021] which supports the work from [Papoudakis et al., 2019] who argued the necessity to find alternative learning approaches that can decompose rewards locally to agents. Several other approaches, such as the potential-based difference reward shaping [Devlin et al., 2014], and the dynamic reward shaping approach, where rewards vary with time [Tenorio-Gonzalez et al., 2010], have been proposed to improve MARL solutions. However, some MARL problems may have to deal with providing a balance between both local rewards and global rewards. This may require a well-crafted and informative reward formulation to incentivise the agents to collaborate in the MARL environment while addressing

the *lazy agent problem.*



(a) Centralised Learning. (b) Decentralised Learning.

Figure 2.4: Centralised and Decentralised Learning.

2.3 Collaboration in Multi-Agent Reinforcement Learning

The collaboration of agents in MARL remains a challenging task [Jaques et al., 2018]. Recent research effort that focused on agents' collaboration [Rashid et al., 2018, Lowe et al., 2017, Foerster et al., 2017] often resorted to CTDE-based approaches to promote collaboration among agents [Gronauer and Diepold, 2022]. Figure 2.4 shows the centralised and decentralised training. During centralised training, each agent is provided with the other agents' information and possibly the global state as seen in Figure 2.4a, while during decentralised execution, each agent makes decisions (independently of any additional information) based on its individual policies as seen in Figure 2.4b [Li et al., 2022, Gronauer and Diepold, 2022]. However, using a centralised controller for training is computationally expensive in large environments and increases the possibility of a single point of failure [Wong et al., 2022]. In the absence of a CC that chooses a single joint policy to be provided to each agent, ensuring harmonised action choice among independent decision-makers requires some mechanism for collaboration [Boutilier, 1999].

Specifically, in most real-life disaster applications, agents may be required to collaborate among themselves without a central entity having prior global knowledge. For example, rescue agent-controlled UAVs may be deployed to an earthquake site to provide monitoring services. The UAVs may need to collaborate with each other without any prior harmonisation [Stone et al., 2010]. More importantly, the fact that humans are normally expected to collaborate in

an ad hoc[4] fashion strongly inspires the challenge of designing autonomous agents of similar capabilities. Collaboration can be achieved when agents are motivated through a derived utility to collaborate or when agents share a common understanding via a communication mechanism to speed up the learning process.

2.3.1 Collaboration Via Reward Assignment

Researchers have investigated emergent behaviours in multiple agents and how collaboration among agents can be achieved. Wong et al. (2022) raised an important question on how to formulate a reward function such that the agents adapt to the actions of each other while achieving collaboration and minimising conflicting behaviour. A collaborative Q-learning approach was proposed in [Zhang et al., 2020b] that models MARL as a dynamic reward assignment problem in a fully collaborative setting. The authors argue that collaboration among agents can be achieved naturally if each agent j acts independently following its own value function, by executing an action that leads to a state that is perceived to be more rewarding to itself than other agents. However, the question of whether collaboration should be implemented on a local scale or a global scale may arise. A collaborative navigation problem was presented in where all agents must collaborate through physical actions to arrive at a set of landmarks [Lowe et al., 2017]. Agents observe the relative positions of other agents and landmarks and are collectively rewarded based on the proximity of any agent to each landmark. However, the authors stated scalability as a downside to this approach and suggested *modularity* where rewards can be assigned to an agent based on neighbour information.

In [Freed et al., 2022], a partial reward decoupling technique was proposed to decompose large collaborative MARL problems into decoupled sub-problems involving subsets of agents, thereby simplifying credit assignment problems. The decomposition is performed in such a way that if agents learn to optimally collaborate with other agents within their subgroup, then the agents will also achieve optimal group-level collaboration. The work [Freed et al., 2022]

[4] An ad hoc network may be established on the fly, without relying on a pre-existing infrastructure. Such networks can be deployed quickly, making them suitable for emergencies such as natural disasters [Goldsmith and Wicker, 2002].

is based on the intuition that if a specific agent j does not impact the expected future reward of another agent z, then agent j can be decoupled from agent z. Several collaborative MARL research focus on equally-shared rewards among agents to motivate them to collaborate and try to avoid selfish behaviours that impact the overall performance [Gronauer and Diepold, 2022]. More generally, we talk about collaborative settings when agents are encouraged to collaborate but do not own an equally-shared reward.

According to [Kim et al., 2019a], this line of research is referred to as a collaborative behaviour without direct communication and it is often achieved via reward assignment. Despite the argument that shaping the reward function may yield improved collaboration [Jaques et al., 2018], collaboration may be also achieved by fostering direct communication among agents themselves [Kim et al., 2019a].

2.3.2 Collaboration via Communication

Communication is widely known to play a crucial role in enhancing collaboration among humans [Számadó, 2010]. Communication among agents is important in addressing the partial observability challenge in MARL since it provides agents with the ability to better collaborate by inferring the underlying state of the environment [Canese et al., 2021]. For instance, agents exploring different parts of the environment can share observations to mitigate partial observability and share their intents to anticipate each others' actions to deal with non-stationarity. Communication protocols are often hand-designed and optimised for the execution of particular tasks [Canese et al., 2021]. Hence, the fundamental question of: What or how information is to be shared to ensure agents collaborate may arise.

Several MARL pieces of work have been proposed to answer the question. Some assume a central communication structure which uses a central controller, a dedicated super agent or a proxy to control and harmonise how messages between agents are exchanged [Zhu et al., 2022, Pesce and Montana, 2019]. Kin et al. (2019a) proposed a multi-agent deep reinforcement learning framework called SchedNet. SchedNet decides which agents should be allowed to broadcast their messages to minimise communication costs. However, it relies on the global

information of each agent's partially observed information for decision-making. Moreover, this approach may be unsuitable in emergencies since it incurs significant control packet overhead and a loss of control packet may lead to system-wide failure. To address this, some work considers a fully connected communication setup where each pair of agents will be connected and packets can be transmitted in an ad hoc broadcast manner [Zhu et al., 2022]. Reinforced Inter-Agent Learning (RIAL) [Foerster et al., 2016], Differentiable Inter-Agent Learning (DIAL) [Foerster et al., 2016], and CommNet[Sukhbaatar et al., 2016] were proposed where agents can share parameters with others via a communication protocol. However, in most real-life applications, system-wide communication may be impractical due to several communication constraints, such as the shared wireless channel [Pesce and Montana, 2019, Kim et al., 2019a], noisy channel [Foerster et al., 2016] and limited bandwidth [Foerster et al., 2016, Kim et al., 2019a]. Moreover, communication among all agents may make it difficult to extract useful information for collaboration, while communication with only nearby agents may restrain the range of collaboration [Jiang et al., 2018]. Nevertheless, neighbouring agents are more likely to interact with and affect each other. Furthermore, it can be costly and counterproductive to consider all other agents when communicating, therefore, Jiang et al. (2018) proposed a partially connected communication setup since it is efficient and effective to consider communication with only neighbouring agents. Zhao et al. (2022) proposed a fully distributed approach that allows agents to share information with their neighbours through a communication network and executes decisions based on its local reward and information received from their neighbours. In the next section, we look into the applications of UAVs in wireless environments and provide insight into recent research on application of UAVs in wireless networks.

2.4 Application of UAVs in Wireless Networks

Although UAVs have found relevance in several military operations such as being used to carry out mission-critical inspection and monitoring services, they are projected to deliver services for civilian applications such as agriculture, transportation, communication, surveillance,

and disaster management [Hayat et al., 2016, Shakhatreh et al., 2019, Mozaffari et al., 2019]. UAVs can also be classified based on type, into fixed-wing and rotary-wing UAVs [Mozaffari et al., 2019]. In contrast to rotary-wing UAVs, fixed-wing UAVs such as small aircraft have more weight, and higher speed, and they need to move forward in order to remain aloft in the sky. However, rotary-wing UAVs such as quadrotors, can hover and remain stationary over a given area [Hayat et al., 2016, Mozaffari et al., 2019]. Depending on their use case scenario in wireless networks, each of these types of UAVs may be deployed. Existing literature on UAV-assisted networks may be categorised into some use case categories as seen in Figure 2.5: UAVs as relays, UAVs as data sinks/disseminators, UAVs as aerial base stations [Hayat et al., 2016]. Nevertheless, recent research on UAV-assisted networks attempts to address some peculiar challenges [Mozaffari et al., 2019] such as optimal 3D placement [Hanna et al., 2019, Shakhatreh et al., 2017], channel modeling [Yan et al., 2019], energy limitation [Galkin et al., 2019a, Zeng and Zhang, 2017, Zeng et al., 2019], flight trajectory planning [Lee et al., 2021b, Liu et al., 2019b], interference management [Warrier et al., 2022, Challita et al., 2019], and connectivity [Liu et al., 2018, Liu et al., 2020]. Table 2.1 shows a summary of related work that applied UAVs in wireless networks.

(a) UAV as a relay. (b) UAV as data sink/disseminator. (c) UAV as an aerial base station.

Figure 2.5: UAVs use case in wireless networks.

2.4.1 UAVs as Relays

UAVs can be used as relays in wireless networks to extend the communication range of a pair of devices or provide an alternative communication path between two communicating entities whose direct link is unavailable due to obstacles or poor wireless channel conditions. For example, the distance from a source to the destination may be too long or blocked, hereby requiring a UAV to act as a relay. Figure 2.5a shows the use of a UAV relay to

complement the existing terrestrial infrastructure that has a limited coverage range. The aerial positioning and flexible deployment of UAVs make them particular candidates for use as relays [Mozaffari et al., 2019]. In particular, UAVs as relays are excellent choices, especially for delay-tolerant applications [Bouk et al., 2015]. The works [Chen et al., 2018a, Chen et al., 2018b, Gao et al., 2021, Demir et al., 2020] used numerical methods to minimise the outage in the network via deploying UAVs as relays. The PSO algorithm was used in [Hadiwardoyo et al., 2020] to improve vehicular communication by optimising the position of the UAV relay. In emergencies with limited functioning communication infrastructures, UAVs serving as relays may help in connecting hard-to-reach devices. Recently, RL techniques have been used to optimise network performance. In [Huang and Xu, 2021], a DQN-based deployment algorithm was proposed to obtain optimal placement of the UAV relay while optimising the energy consumption. In [Lee et al., 2021a], a deep reinforcement learning (DRL) technique was proposed to improve UAV positioning in such a way as to maximise the number of connections while maintaining a strongly connected UAV network. The approach shows the potential of using RL techniques to aid communication during disaster relief operations.

2.4.2 UAVs as Data Sinks/Disseminators

On one hand, UAVs can be used as data disseminators to transmit data from one point to another within the network. For example, over-the-air updates could be carried out using UAVs where UAVs travel across different locations with the objective of deploying certain software updates on ground devices. On the other hand, UAVs can be used as data sinks to aggregate sensed data from ground devices and possibly forward the collected data to a remote location for processing. A classical method was used in [Xue et al., 2019] to maximise the amount of UAV data disseminated among spatially dispersed Internet-of-Things (IoT) devices by jointly optimising the resource assignment strategy and UAV's mobility in the 3D space. The work is applicable to infrastructure-less IoT where UAVs are deployed as data sinks/disseminators as seen in Figure 2.5b. A DRL technique was adopted in [Zhang et al., 2020a] to jointly minimise the age of information and the energy consumption of a UAV acting as a data sink in a wireless sensor network (WSN).

Table 2.1: UAV-Assisted Networks

References	Approach	Multi-UAV	Use-Case	Objective	2D/3D	SINR/SNR	Static/Mobile	Control
[Mozaffari et al., 2017]	Iterative	Yes	Base station	EE	3D	SINR	Static	Centralised
[Liu et al., 2020]	MADDPG	Yes	Base station	EE + Coverage	2D	–	Static	Decentralised
[Wang et al., 2021]	MADDPG	Yes	Base station	EE + Coverage	2D	–	Static	Decentralised
[Ruan et al., 2018]	Game Theory	Yes	Base station	EE + Coverage	2D	–	Static	Centralised
[Liu et al., 2018]	Actor-Critic	Yes	Base station	EE + Coverage	2D	–	Static	Centralised
[Liu et al., 2019a]	QL	Yes	Base station	Energy + Coverage	3D	SNR	Both (RW)	Centralised
[Lee et al., 2021b]	DQN	Yes	Base station	Energy + Coverage	3D	SNR	Mobile	Centralised
[Galkin et al., 2022a]	Dueling DQN	No	Dissemination	Association	3D	SINR	–	Centralised
[Cicek et al., 2019]	Survey	Yes	Base station	Users QoS	3D	SNR	Both	–
[Kalantari et al., 2017]	Branch & bound	No	Base station	Coverage	3D	SNR	Both	Centralised
[Bayerlein et al., 2021]	DRL	No	Data harvesting	Throughput	2D	SNR	Mobile	Centralised
[Liu et al., 2019b]	QL	Yes	Base station	Throughput	3D	SINR	Mobile	Centralised
[Chen et al., 2018a]	Numerical	Yes	Relay	Outage + BER	2D	SNR	–	–
[Chen et al., 2018b]	Numerical	No	Relay	Outage + BER	2D	SNR	–	–
[Gao et al., 2021]	Numerical	No	Relay	Outage + BER	2D	SNR	Mobile	–
[Demir et al., 2020]	Numerical	No	Relay	Latency	2D	SNR	Mobile (V2X)	Centralised
[Islam et al., 2022]	LSTM+PSO	Yes	Relay	Coverage	2D	–	Mob. (SUMO)	Centralised
[Saxena et al., 2019]	DRL	Yes	Base station	Coverage	3D	SINR	Mob. (data)	Centralised
[Oubbati et al., 2019]	Heuristic	Yes	Relay	PDR + Delay	3D	–	Mob. (SUMO)	Centralised
[Raza et al., 2021]	Heuristic	Yes	Relay, BS	Energy + Delay	2D	–	BonnMotion (NS-2)	Centralised
[Samir et al., 2021]	Convex opt.	Yes	Relay	Dep. UAVs	2D	SNR	Mob. (Poisson)	Centralised
[Lin et al., 2020]	Swarm opt.	Yes	Relay, BS	Delay,Thput	2D	–	Mob. (Data)	Centralised
[Hadiwardoyo et al., 2020]	PSO	No	Relay	RSSI	3D	–	Mob. (SUMO)	Centralised
[Betalo et al., 2022]	GA + MSTP	Yes	Sink	WSN Lifetime	2D	SNR	Static	Centralised
[Bayerlein et al., 2021]	Double DQN	Yes	Sink	Mission Time	2D	SNR	Static	Centralised
[Lee et al., 2021a]	DRL (PPO)	Yes	Relay	Connectivity	3D	–	Mob.(RPGM)	Centralised
[Huang and Xu, 2021]	DQN	Yes	Relay	Outage + Energy	2D	SINR	Static	Centralised
[Hadiwardoyo et al., 2019]	Testbed	No	Relay	RSSI	3D	–	Mob. (SUMO)	Centralised
[Zhang et al., 2020a]	Actor-Critic	No	Relay	AoI, Energy	3D	–	–	Centralised
[Peng and Shen, 2020]	DDPG	Yes	Base station	Resource Offload	2D	SINR	Mobile (V2X)	Centralised
[Yuan et al., 2021]	Actor-Critic	Yes	Base station	Throughput	2D	SNR	Mobile (V2X)	Centralised
Our Proposed	DMARL	Yes	Base station	EE, Energy, Coverage	3D	SINR	Both (RW, RWP, GMM, SUMO)	Decentralised

The work [Betalo et al., 2022] deployed multiple UAVs to serve as data sinks in a WSN while minimising the total energy consumed by the UAVs. The approach applied the Genetic Algorithm (GA) to maximise the WSNs' lifetime and then used Multiple Traveling Salesman Problem (MTSP) based path planning algorithm to solve the flight trajectory of the UAVs. In [Bayerlein et al., 2021], an autonomous UAV is tasked with gathering data from distributed sensor nodes subject to limited flying time and obstacle avoidance. To avoid the challenge of expensive recomputations or to relearn a behaviour when a change in the scenario parameters occurs, the authors proposed a double deep Q-network (DDQN) with combined experience replay to learn a UAV control policy that generalises over changing conditions. Throughout this thesis, our focus will be on the deployment of UAVs providing wireless connectivity to a set of ground users. Next, we will discuss in detail the application of UAVs as aerial base stations.

2.5 UAV Base Station Deployment

Research into UAV deployment in wireless cellular networks has gained pace in recent years [Mozaffari et al., 2019]. UAVs can readily serve as aerial base stations in wireless networks by providing ubiquitous coverage within a serving geographical area [Shakhatreh et al., 2019]. For instance, a UAV may be deployed for rapid service recovery after disaster scenarios, or when the cellular network service is not available or play a crucial role in complementing existing cellular infrastructures when service demand is at its peak [Hayat et al., 2016, Shakhatreh et al., 2019]. In Figure 2.5c, a rotary-wing UAV is deployed to provide wireless connectivity to ground users in the absence of cellular service. Since rotary-wing UAVs have the unique ability to hover and remain stationary over a given area [Galkin, 2021, Hayat et al., 2016, Mozaffari et al., 2019], throughout this thesis, we consider the deployment of rotary-wing UAVs for providing coverage to the ground used. Nevertheless, it is argued in [Galkin, 2019] that the energy consumption of a rotary-wing UAV exceeds that of fixed-wing aircraft since the forward thrust needed to make a fixed-wing aircraft airborne is significantly smaller than the force needed in rotary-wing UAV, and this directly translates to a longer flight

duration. On this note, it is important to improve energy utilisation by optimising the flight trajectory of these rotary-wing UAVs if they are to be deployed to serve ground users for an extended period of time.

A large number of work focus on single UAV deployment to serve ground users [Zeng and Zhang, 2017, Zeng et al., 2019], while others [Mozaffari et al., 2017, Liu et al., 2020] focus on the deployment of multiple UAVs to provide wireless coverage to ground users in geographically large areas.

2.5.1 Single UAV Deployment

In certain circumstances, a single UAV may be deployed to serve ground users. A classical method was proposed in [Zeng and Zhang, 2017] to maximise the UAV's energy efficiency while optimising the flight trajectory of a fixed-wing UAV. A travelling salesman problem is formulated in [Zeng et al., 2019] to optimise the flight trajectory of a rotary-wing UAV. The authors in [Azari et al., 2018] proposed an analytical approach to minimise the outage probability by optimising the UAVs' height. In [Xu et al., 2011], a generic optimal terrain coverage algorithm was proposed to automate terrain coverage using a single UAV. The work [Shakhatreh et al., 2017] proposes a particle swarm optimization (PSO) algorithm to find an efficient 3D placement of a single UAV that minimises the total transmit power needed to serve some indoor users. Since these works focus on single UAV deployments, they may be impractical in geographically-large areas [Liu et al., 2018, Liu et al., 2019a] where more than one UAV may be required to serve. On this note, throughout this thesis, our focus will be on the deployment of multiple UAVs.

2.5.2 Multi-UAV Deployment

Multiple UAVs may be deployed in a shared environment to perform a coverage task. The multi-UAV deployment problem maps to an NP-hard problem [Sanchez-Aguero et al., 2020, Galkin et al., 2016, Liu et al., 2019a]. Several works assume an oracle having the global knowledge of ground users' locations that partitions the entire coverage region into clusters and assigns UAVs to serve in each cluster as shown in Figure 2.6. The work [Galkin et al.,

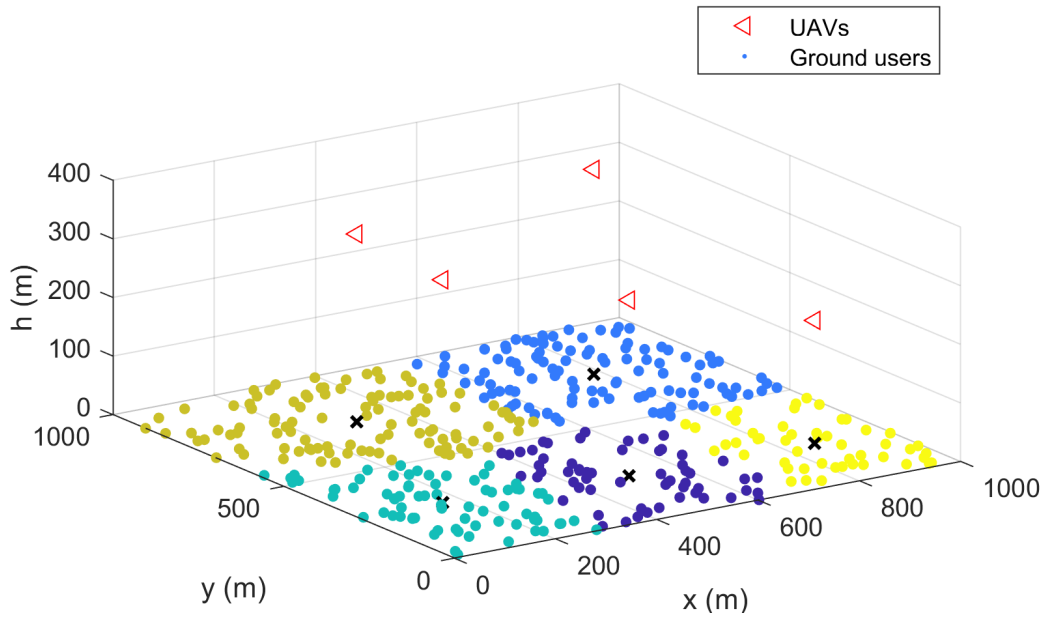


Figure 2.6: K–Means clustering-based algorithm [Galkin et al., 2016, Liu et al., 2019a] with 5 UAVs deployed to serve 400 ground users in five clusters partitioned by a CC.

2016] proposed a K-Means clustering algorithm to partition the coverage area and the enclosed users into K subsets which represent candidate coverage areas for the UAVs. Each UAV assigned to serve a subset will position itself in the subset's centroid. In [Liu et al., 2019a], a K-Means clustering algorithm was first applied before using the tabular Q-learning to optimise the flight trajectory of the UAVs around the centroids. The work [Kalantari et al., 2016] applied a PSO algorithm to find the 3D placement of multiple UAVs serving ground users.

In [Mozaffari et al., 2017], multiple UAVs were deployed to serve a set of ground users with the focus of minimising the total energy consumed by the UAVs using an iterative algorithm. However, the work was limited to the coverage of static ground users. In [Ruan et al., 2018], a game-theoretic approach was proposed to optimise the system's EE of deployed UAVs while maximising the ground area covered irrespective of the presence of ground users. The works [Liu et al., 2018, Liu et al., 2020] consider the deployment of multiple UAVs flying at a fixed altitude to serve ground users in a target region. The target region is divided into K cells with each cell containing a point of interest (possibly the position of the ground user) as shown in Figure 2.7. However, ground users' mobility may pose a huge challenge in these approaches since the central entity that pre-partitions the coverage region may need to send

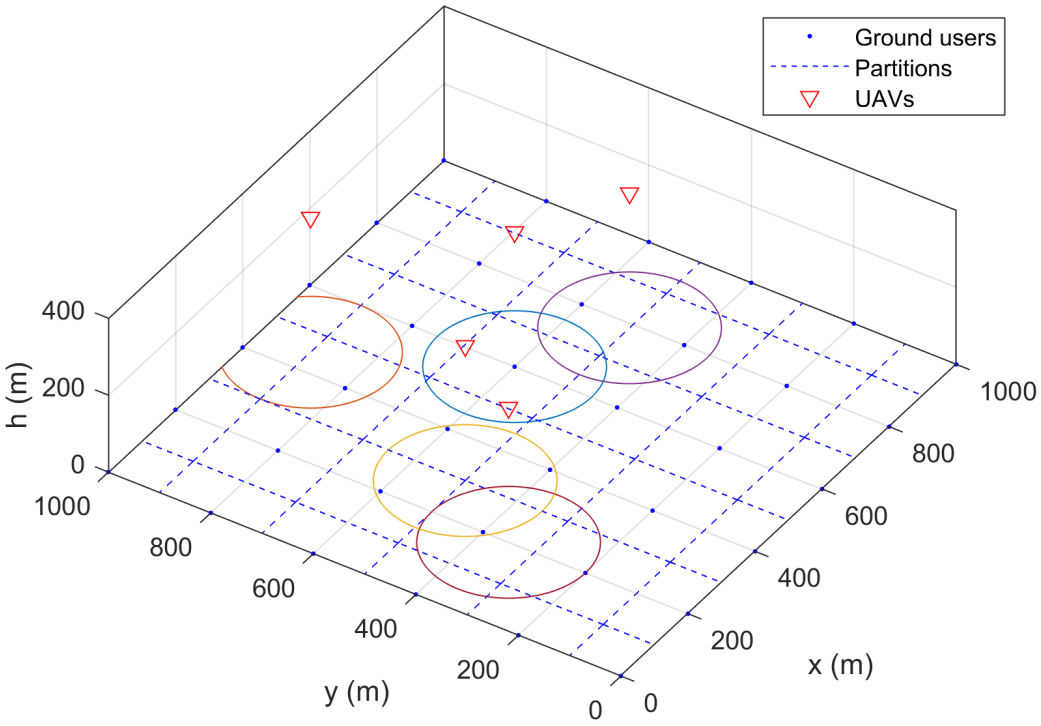


Figure 2.7: K–Cells partitioning-based algorithm [Liu et al., 2020] with 5 UAVs deployed to serve a set of ground users on the pre-partitioned geographical space.

periodic updates to the UAVs for decision-making. Specifically, these approaches may be impractical in disaster scenarios as it requires significant communication overhead of control packets.

2.6 AI-based Control in UAV-Assisted Networks

Artificial Intelligence (AI)-based control can be achieved in a centralised or decentralised manner [Dusparic, 2010]. AI-based control in a UAV-assisted network entails the control of flight and navigation of UAVs in the sky following pre-defined or adaptive waypoints[5] [Hayat et al., 2016]. These waypoints can be decided by a CC, residing in a base station or ground control station, and then sent over a dedicated communication link to the UAV. A UAV may also decide its trajectory on-the-fly by using the information collected from its environment (terrain, obstacles, as well as the presence of other UAVs) via onboard sensors [Hayat et al., 2016]. Throughout this thesis, we focus on the latter, where UAVs act autonomously and without reliance on a CC for decision-making. Over the years, there has been growing

[5]A waypoint is an intermediate point or line of travel.

research interest towards agent-based control in UAV-assisted networks [Liu et al., 2019a, Galkin et al., 2022a, Liu et al., 2020], with each agent-based design serving some specific functions. A centrally-controlled actor-critic algorithm was proposed in [Samir et al., 2021] to optimise the trajectories of UAVs while maximising the coverage of vehicles in an interference-free environment. However, as the number of UAVs in the network are increased, it may become impractical for effective decision-making and control in disaster scenarios. Multi-agent learning is challenging in itself, requiring agents to learn their policies while taking into account the consequences of the actions of others. Table 2.2 shows a summary of RL-related work on multiple UAVs deployed as aerial base stations.

2.6.1 Centralised Control

A central controller (CC) is an entity in the network that is designated as the controller and is responsible for managing the execution of other entities. The work in [Sherman et al., 2021] considered a Q-learning agent called a CC which controls the UAVs in the network, by using the ϵ–greedy based action selection to increase the life-span of the network. The authors assume that a GCS is responsible for collecting and monitoring the UAV locations, their energy levels, and the energy levels of the charging stations, as well as controlling the actions of the UAVs. In [Galkin et al., 2016, Liu et al., 2019a], users in the same cluster j are served a single UAV j that is assigned to serve that cluster as seen in Figure 2.6. The partitioning of the coverage region can only be achieved by an oracle or CC with prior spatial knowledge of the locations of users within the area of interest. Moreover, there will be a need for the CC to always recalculate the centroid (i.e., the geometric centre) of the cluster when ground users change their position. In [Liu et al., 2018, Liu et al., 2020], the coverage region is pre-partitioned into k–cells by a CC. It is argued in [Chen et al., 2022] that interference management is crucial for providing satisfactory service, especially in emergency scenarios, where it is unrealistic to control all UAVs in a centralised manner by gathering global user information. In [Samir et al., 2021], UAVs are deployed to a road segment to serve mobile vehicles on a highway. The UAVs periodically send observations from the vehicular environment to the central control agent, where the actor and critic networks

decide the best control policy that jointly maximises the energy utilisation and the number of connected vehicles. The work [Peng and Shen, 2020] proposed the use of a CC to perform resource optimisation in a UAV-assisted vehicular network. The work considered the CC to be controlled by a DDPG agent to support as many offloaded computing tasks with satisfied delay and QoS requirements. In [Yuan et al., 2021], a deep RL approach was proposed to improve the total throughput in a UAV-assisted vehicular network under some energy constraints. The work assumes the presence of a CC that aggregates vehicles into cells based on their location. However, the complexity of the algorithm is further increased as the number of cells in the network is increased. Moreover, the authors reiterated the need to reduce the computational workload on the central agent by considering a multi-agent system where agents are deployed in UAVs. In this thesis, we aim to investigate the performance of our proposed solution in different vehicular network settings and traffic conditions. The authors [Chen et al., 2022] resorted to a CTDE to overcome the non-stationarity issue of MARL and to endow the UAVs with distributed decision-making capability. However, these algorithms may not work as well when changes in the environment occur during execution. Intuitively, it may be challenging to have such global knowledge in a disaster scenario or a quickly evolving network topology. Furthermore, since the UAVs will be completely reliant on periodic updates from the CC for decision-making, a failure in the CC may lead to undesirable service downtime. In general, the choice of a decentralised approach is motivated by the fact that a centralised approach will require additional control signals to be transmitted to the UAVs continuously.

2.6.2 Decentralised Control

Decentralised control entails having the logic, and input/output functions located at the individual entity, which is completely independent of other entities. Normally, these kinds of systems might require some means of collaboration when working on a joint task. A distributed placement approach was proposed in [Hanna et al., 2019] using iterative gradient descent and an iterative brute-force method to find the optimise the positions of UAVs in

order to improve the LoS multiple-input and multiple-output (MIMO)[6] channel capacity. It was reiterated in [Hanna et al., 2019] that solving optimization centrally would be too complex, while offloading tasks to the base station would incur a large communication overhead. Like [Chen et al., 2022], the works [Liu et al., 2020] used a CTDE approach to handle the non-stationarity. The works [Liu et al., 2020, Wang et al., 2021] adopt a Multi-Agent Deep Deterministic Policy Gradient (MADDPG) that trains a centralised critic for each agent and each critic takes both the full actions as well as the observations of all agents as its input. Nevertheless, during execution, each agent is allowed to execute its policy in a decentralised manner. The distributed MADDPG approach proposed in [Liu et al., 2020, Wang et al., 2021] was an improvement to the fully-centralised learning approach in [Liu et al., 2018], where all agents are controlled by a single actor-critic network, both during training and execution. Although these approaches [Liu et al., 2020, Liu et al., 2018, Wang et al., 2021] focus on optimising the systems' EE while serving static pedestrian users, they did not account for the interference from neighbouring UAV cells. Collaboration among UAVs was not considered in [Liu et al., 2019a, Liu et al., 2018, Liu et al., 2020, Wang et al., 2021, Chen et al., 2022], however, learning convergence was achieved via a central controller that has global knowledge of the environment. In a fully-decentralised, interference-limited environment, UAVs require robust collaborative strategies to optimise their flight trajectory while providing coverage to ground users. Since the UAVs have the goal improving the total EE of the UAVs while jointly optimising the number of connected ground users and energy utilisation of the UAVs, the UAVs must be capable of changing their location while serving in a shared, dynamic and interference-limited network environment.

2.7 Summary

This chapter analysed current research on RL-based multi-UAV systems. Several approaches have been proposed in the literature, for a variety of different UAV scenarios and different applications. In this thesis, we limit our scope to the deployment of multiple rotary-wing

[6] MIMO is a term is used in wireless communication to denote the use of multiple antennas at the transmitter and the receiver.

Table 2.2: Related RL Work on Multiple UAVs Deployed as Aerial Base Stations.

Paper	*Approach*	*Training*	*Execution*	*Collaborative*	*Flight Trajectory*
[Liu et al., 2018]	DDPG	Centralised	Centralised	✗	2D
[Liu et al., 2020]	MADDPG	Centralised	Decentralised	✗	2D
[Chen et al., 2022]	MADDPG	Centralised	Decentralised	✗	2D
[Wang et al., 2021]	MADDPG	Centralised	Decentralised	✗	2D
[Samir et al., 2021]	DDPG	Centralised	Centralised	✗	2D
[Peng and Shen, 2020]	DDPG	Centralised	Centralised	✗	2D
[Yuan et al., 2021]	DDPG	Centralised	Centralised	✗	2D
[Liu et al., 2019a]	Cluster-based QL	Centralised	Centralised	✗	3D

Paper	*Ground Users*	*CC Partitioning*	*Interference*	*EE*	*Objective*
[Liu et al., 2018]	Static	K-Cells	✗	✓	EE, Coverage
[Liu et al., 2020]	Static	K-Cells	✗	✓	EE, Coverage
[Chen et al., 2022]	Static	–	✓	✗	Energy
[Wang et al., 2021]	Static	–	✗	✗	Energy, Fairness
[Samir et al., 2021]	Mobile (Vehicles)	–	✗	✗	Energy, Coverage
[Peng and Shen, 2020]	Mobile (Vehicles)	–	✓	✗	Resource Offloading
[Yuan et al., 2021]	Mobile (Vehicles)	K-Cells	✗	✗	Throughput
[Liu et al., 2019a]	Mobile (RW)	K-Clusters	✗	✗	Trajectory

UAVs serving as aerial base stations to serve ground users in emergencies, where there is a service outage due to failure in existing cellular infrastructure or increased service demand on limited available infrastructure. However, UAVs are energy-constrained and deplete energy while providing wireless coverage to ground users for an extended period. Researchers have proposed to improve the energy utilisation of UAVs by optimising their flight trajectory with the assistance of a CC that has global knowledge of the ground users' locations, partitions the coverage area into clusters and then assigns UAVs to serve in each cluster. An iterative algorithm was proposed in [Mozaffari et al., 2017] to minimise the energy consumption of UAV base stations providing coverage to static ground users. However, we understand that ground users may be mobile, thus, further research may be required to investigate the impact of mobility on the total energy efficiency in the network. The work [Liu et al., 2019a] applied a centralised cluster-based Q-learning where the UAVs are controlled by a CC and deployed to serve pedestrians following a random walk mobility model. The work also neglected the impact of interference from neighbouring UAVs, which may require further investigations. Furthermore, its applicability in disasters may be an issue of concern in terms of sending control packets back and forth the network. This assumption may be impractical in disaster scenarios where there is a loss of control packets due to possible failure in the CC, or difficulty in tracking users' location for periodic updates to be sent to UAVs. For these reasons,

we consider these works [Mozaffari et al., 2017, Liu et al., 2019a] one of the closest and further use them as baselines. Our work sets out to eliminate the need for a CC due to our specific emergency scenario that assumes a service outage in the existing terrestrial cellular network. Thus, we attempt to answer the research question **RQ1**, "*Can UAVs serving mobile ground users improve the total system's energy efficiency in a shared, dynamic and interference-limited network environment without relying on a central controller for decision-making?*".

Unfortunately, a majority of existing work consider scenarios where UAVs operate in interference-free environments, either because they operate in isolation from other devices or because they allocate different spectrum to each cluster. The work [Liu et al., 2020] considered a multi-agent DDPG approach with centralised training and decentralised execution to optimise the total EE of the UAVs serving a set of static ground users. The work also ignored the impact of interference from near-by UAV cells, which may not be practical when UAVs sharing the same frequency spectrum are deployed in a shared environment. This assumption makes the problem of optimising the EE of multiple UAVs deployed as aerial base stations in a shared wireless environment tractable, however, it limits the applicability of the work. For these reasons, we consider the work [Liu et al., 2020] one of the closest and further use it as a baseline. Our work considers multiple UAVs operating as aerial base stations in a shared environment where the frequency spectrum is a scarce resource and the UAVs may have to reuse this frequency resource. Our assumption may be practical and useful in spectrum resource management, however, it introduces non-stationarity in the environment through interference from nearby UAVs or APs sharing the same frequency spectrum. Moreover, interference makes it difficult for UAVs to discover the best set of actions to execute in this shared environment without insights from a CC. More importantly, if interference is not effectively managed, it may hinder a UAV from providing coverage in an energy-efficient manner since the interference experienced from neighbouring UAV cells may lead to a decrease in the total system's EE. In this thesis, we build on collaborative MARL works that improve collaboration among agent-controlled UAVs. Therefore, we attempt to answer the research

question **RQ2**, "*Can collaboration with closest neighbours improve the total system's energy efficiency while minimising the total energy consumed by UAVs in a shared, dynamic and interference-limited network environment?*".

Based on our review of related work, there is growing adoption of disruptive machine learning techniques among researchers to solve complex optimisation problems in UAV-assisted networks. In particular, MARL-based algorithms have shown great potential in improving the overall system's EE in these networks. To enhance learning in MARL for UAV-assisted networks, several approaches have resorted to CTDE-based approaches, where each agent-controlled UAV is provided with the other agent-controlled UAVs' information during the centralised training phase while allowed to act independently based on its individual policies during the decentralised execution phase. Many works have embraced this shift from fully centralised control to the CTDE-based method. However, the CTDE-based method may not be well suited in dynamic environments with the presence of mobile ground users. The decentralised control of UAVs is suitable in disaster scenarios such as ours. Service downtime may occur in such disaster scenarios and this may be due to failure in existing terrestrial infrastructure or ground control station, thereby, making it difficult for UAVs to make decision via a central entity. In such circumstances, UAVs are able to interact with each other in a decentralised manner to ensure ubiquitous coverage in the network. Specifically, RL algorithms have been proposed in such multi-UAV deployment environments to improve the total systems' EE in such dynamic environments. However, using a decentralised approach introduces several challenges that make it difficult for UAVs to effectively collaborate while providing coverage to ground users. Furthermore, in quickly evolving networks with highly mobile and densely-distributed ground users, it makes the coverage task under a strict energy budget difficult. In this thesis, we adopt robust MARL strategies that allow the agent-controlled UAVs intelligently collaborate while serving highly mobile, dense and unevenly distributed users. We attempt to answer our research question **RQ3**, "*Can UAVs collaborate intelligently to improve the total system's energy efficiency in highly mobile, dense and unevenly distributed users in an urban environment?*".

In this thesis, we aim to answer the above research questions through our three contributions in Section 1.4 while meeting the design requirements specified in Section 4.1. In the next chapter, we present the system model used in our multi-UAV design and go further to formulate the problem as a multi-UAV MARL-based optimisation problem.

Chapter 3

Multi-UAV Model Design

In the previous chapter, we presented a review of the existing multi-UAV RL-based optimization techniques. In this chapter, we present the models for wireless connectivity, mobility, energy consumption, fairness, and energy efficiency (EE). We then formulate the problem as a multi-UAV MARL-based optimisation problem.

3.1 System Model

We consider a UAV-assisted network with a set U of N number of quadrotor UAVs deployed to serve ground users in an urban setting as shown in Figure 3.1. We assume that each user $\xi_i \in \xi$ is equipped with a transceiver that allows for the transmission and reception of wireless signals. In this thesis, we assume service unavailability in existing terrestrial infrastructure due to disaster, unforeseen load or failure in parts of the network.

3.1.1 Wireless Channel Model

A radio channel is one that allows a wireless device to transmit a signal directly to other wireless devices [Goldsmith and Wicker, 2002]. In this thesis, we assume guaranteed Line-of-Sight conditions between U_j^t located at $(x_j^t, y_j^t, h_j^t) \in \mathbb{R}^3$ and ξ_i^t at $(x_i^t, y_i^t) \in \mathbb{R}^2$ due to the aerial positions of the UAV. However, the wireless channel is assumed to be impaired by interference from nearby UAV cells or other access points sharing the same frequency

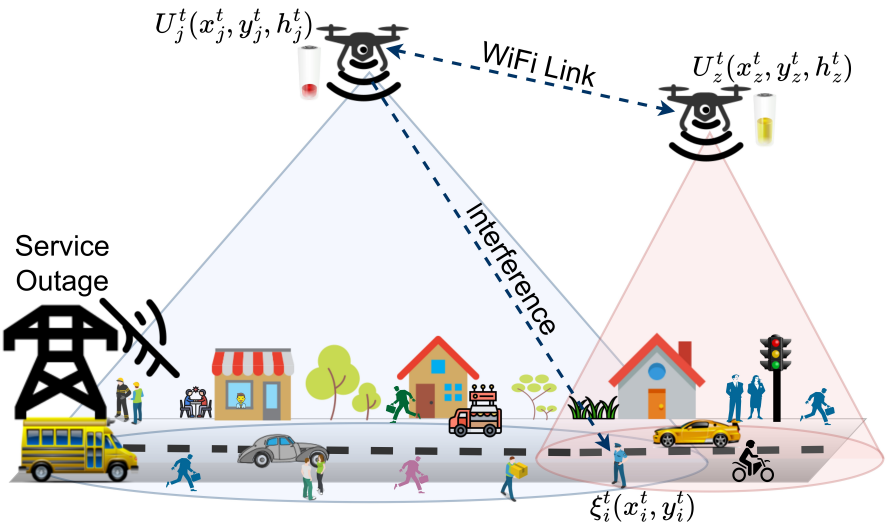


Figure 3.1: System model for UAVs providing coverage to ground users.

spectrum. Signal-to-interference-plus-noise-ratio (SINR) is a measure of signal quality. It can be defined as the ratio of the power of a certain signal of interest and the interference power from all the other interfering signals plus the noise power [Hadj-Kacem et al., 2020]. The SINR between two communicating devices will typically decrease as the distance between the devices increases and is also a function of the signal propagation and interference in the environment [Goldsmith and Wicker, 2002, Hadj-Kacem et al., 2020]. More importantly, the link SINR varies randomly over time due to the mobility of the devices that typically change the transmission distance, propagation environment, and interference characteristics.

In time-step t, each user $\xi_i \in \xi$ can be served by a single UAV $j \in U$ which provides the strongest downlink SINR. Hence, the SINR at time t is expressed as,

$$\gamma_{i,j}^t = \frac{\beta P (d_{i,j}^t)^{-\alpha}}{\Sigma_{z \in \chi_{int}} \beta P (d_{i,z}^t)^{-\alpha} + \sigma^2}, \tag{3.1}$$

where β and α are the attenuation factor and path loss exponent that characterises the wireless channel, respectively. σ^2 is the power of the additive white Gaussian noise at the receiver, $d_{i,j}^t$ is the distance between the i and j at time t, and expressed as $d_{i,j}^t = \sqrt{(x_i^t - x_j^t)^2 + (y_i^t - y_j^t)^2 + (h_j^t)^2}$, while $d_{i,z}^t$ is the distance between the i and z at time t, and given as $d_{i,z}^t = \sqrt{(x_i^t - x_z^t)^2 + (y_i^t - y_z^t)^2 + (h_z^t)^2}$. χ_{int} is the set of interfering UAVs within the coverage region (i.e., area of operation of the UAVs providing wireless connectivity to

ground users). z is the index of an interfering UAV in the set χ_{int}. P is the transmit power of the UAVs. To provide ubiquitous connectivity to the users, the UAVs must optimise their flight trajectories. Given a channel bandwidth B_w, the receiving data rate at the user can be expressed using Shannon's equation [Galkin et al., 2022b],

$$\Re_{i,j}^{t} = B_w \log_2(1 + \gamma_{i,j}^{t}). \tag{3.2}$$

The overall bandwidth in Shannon's equation is divided among users based on the SINR, which determines the capacity of the channel.

3.1.2 Connectivity Model

We consider an interference-limited system where coverage is affected by the SINR. Thus, the connectivity score of a UAV $j \in U$ at time t is calculated as [Liu et al., 2020],

$$C_j^t = \sum_{\forall i \in \xi} w_j^t(i), \tag{3.3}$$

where $w_j^t(i) \in \{0, 1\}$ represents whether user i is connected to UAV j at time t. $w_j^t(i) = 1$ if $\gamma_{i,j}^t > \gamma_{th}$, otherwise $w_j^t(i) = 0$, where γ_{th} is the SINR predefined threshold. Likewise $\mathcal{R}_{i,j}^t = 0$ if user i is not connected to UAV j. We assume that the UAVs should keep a minimal distance d_{col} from each other to avoid collision [Wang et al., 2021]. In a variety of network service scenarios, including disasters, it is desirable to have nearly all ground users connected fairly to the available UAVs. Jain's fairness index is a metric used to assess fairness on the connectivity of ground users [Liu et al., 2020]. The Jain's fairness index produces a value between 0 and 1. A higher value indicates a higher level of fairness, while a lower value indicates a more unfair distribution. As such, we define geographical fairness using Jain's fairness index as [Liu et al., 2020, Wang et al., 2021],

$$f_t^j = \frac{\left(\sum_{\forall j \in U} C_t(j)\right)^2}{N \sum_{\forall j \in U} C_t(j)^2} \tag{3.4}$$

3.1.3 Mathematical Mobility Model

Due to the difficulty in obtaining non-sparse and temporal mobility traces, several mathematical mobility models have been proposed in ad-hoc network literature to depict the realistic mobility patterns of ground users. Here, we present three widely used models [Camp et al., 2002]:

1. Random Walk (RW) mobility model: The RW was developed to mimic the stochastic behaviour of mobile ground devices [Camp et al., 2002]. In RW, the ground devices can change their speed $[V_{min}, V_{max}]$ and direction $\theta(t)$ randomly and uniformly distributed in the range $[0, 2\pi]$ in each time-step t with zero pause time [Roy, 2011]. Each movement occurs either in constant time intervals or in a constant travel distance [Biomo et al., 2014].

2. Random Way Point (RWP) mobility model: The RWP is a more realistic mobility model used ad-hoc networks with an introduction of device pause times between changes in direction and/or speed [Roy, 2011]. In this model, the travel distance varies in each time-step t [Camp et al., 2002]. However, the RW and RWP models are subject to sudden stops, sudden speed changes and sudden changes in the direction where a user can make a sharp a 180 degrees turn [Biomo et al., 2014].

3. Gauss–Markov Mobility (GMM) Model: The GMM was designed to adapt to different levels of randomness via one tuning parameter [Camp et al., 2002]. The design was inspired by the need for a more realistic mobility model, that is, it allows the users to accelerate, decelerate, or turn progressively while avoiding sharp turns. Initially, each ground user is assigned a current speed and direction. At each time step, movement occurs by updating the speed and direction of each user by following a Gaussian distribution [Camp et al., 2002, Biomo et al., 2014].

3.1.4 Energy Consumption Model

During a flight operation, UAV $j \in U$ at time t expends energy e_j^t. A UAV's total energy e_T is expressed as the sum in propulsion e_P and communication e_C energies, $e_T = e_P + e_C$. Since e_C is practically much smaller than e_P, i.e., $e_C \ll e_P$, we ignore e_C [Eom et al., 2020, Zeng and Zhang, 2017]. A closed-form analytical propulsion power consumption model for a rotary-wing UAV at time t is given as [Zeng et al., 2019],

$$P(t) = \kappa_0\Big(1 + \frac{3V^2}{U_{tip}^2}\Big) + \kappa_1\Big(\sqrt{1 + \frac{V^4}{4v_0^4}} + \frac{V^2}{2v_0^2}\Big)^{\frac{1}{2}} + \frac{\kappa_2}{2}V^3, \tag{3.5}$$

where κ_0, κ_1 and κ_2 are the UAVs' flight constants (e.g., rotor radius, disk area, drag ratio, air density, solidity or weight), U_{tip} is the rotor blade's tip speed, v_0 is the mean hovering velocity, and V is the UAVs' speed at time t. In particular, we take into account the basic operations of the UAV, such as hovering and acceleration. Therefore, we can derive the average propulsion power over all time steps as $\frac{1}{T}\sum_{t=1}^{T} P(t)$, and the total energy consumed by UAV j at time t is given as,

$$e_j^t = \delta_t \cdot P(t), \tag{3.6}$$

where δ_t is the duration of each time-step. The energy efficiency (EE) of UAV j can be expressed as the ratio of the data throughput and the energy consumed in time-step t, expressed as,

$$\eta_j^t = \frac{\sum\limits_{i \in \xi} \mathcal{R}_{i,j}^t}{e_j^t}. \tag{3.7}$$

EE is an important metric used to measure how effectively energy is utilised to achieve a desired outcome of improving the throughput in the network. The metric helps to maintain a reliable quality of service that is being offered to ground users, since the EE is a function of the throughput of the UAVs. On one hand, if the UAVs consume too much energy through flight then EE will be low which reflects that it is not a good business trade-off. On the other hand, if the UAVs consume lesser energy but there is barely any user demand in the area then EE will be low as well. We use the EE metric because it is a good, quantitative way of describing the value created by having UAVs in a given area versus how much it costs the

operator. This could be seen as an advantage to extend the coverage duration of the UAVs serving ground users. Therefore, the total systems' EE over all time-step is given as,

$$\eta_{tot} = \frac{\sum_{t=1}^{T} \sum_{j \in U} \sum_{i \in v} \mathcal{R}_{i,j}^{t}}{\sum_{t=1}^{T} \sum_{j \in U} e_j^t}. \tag{3.8}$$

3.2 Problem Formulation

Our objective is to maximise the total system's EE by jointly optimising each UAV's trajectory, number of connected users, and the energy consumed by the UAVs under a strict energy budget. Therefore, the problem is formulated as,

$$\max_{\forall j \in N:\ \mathbf{x_j^t},\ \mathbf{y_j^t},\ \mathbf{h_j^t},\ \mathbf{e_j^t},\ \mathbf{C_j^t}} \eta_{tot} \tag{3.9a}$$

$$\text{s.t.} \quad \gamma_{i,j}^{t} \geq \gamma_{th}, \quad \forall w_j^t(i) \in [0,1],\ i,\ j,\ t, \tag{3.9b}$$

$$e_j^t \leq e_{\max}, \qquad \forall j,\ t, \tag{3.9c}$$

$$x_{\min} \ \leq \ x_j^t \ \leq x_{\max}, \qquad \forall j,\ t, \tag{3.9d}$$

$$y_{\min} \ \leq \ y_j^t \ \leq y_{\max}, \qquad \forall j,\ t, \tag{3.9e}$$

$$h_{\min} \ \leq \ h_j^t \ \leq h_{\max}, \qquad \forall j,\ t, \tag{3.9f}$$

where x_{min}, y_{min}, h_{min} and x_{max}, y_{max}, h_{max} are the minimum and maximum coordinates of x, y and h, respectively. $e_{\max}$ is the UAV's maximum energy budget. The constraints in Equation (3.9b)–(3.9f) ensure that the UAVs stay within tolerable bounds. The constraint (3.9b) is to ensure that users meet the minimum SINR threshold. The constraint in Equation (3.9c) ensures that the UAVs do not exceed their maximum energy budget, while the constraints in Equation (3.9d) – (3.9f) are to keep the UAVs within the operating area. As multiple wireless transmitters sharing the same frequency spectrum are deployed in close proximity to each other, it becomes more challenging to manage interference in the network. Recall that we assume that all UAVs share the same frequency band. The amount of interference received

from other interfering sources is a function of the UAVs' locations [Mozaffari et al., 2017]. In particular, the problem in Equation (3.9a) is known to be NP-hard [Sanchez-Aguero et al., 2020, Liu et al., 2019a]. Hence, it is difficult to solve using conventional optimisation approaches [Liu et al., 2019a]. Due to the non-stationarity introduced in the environment, UAVs may become selfish by pursuing individual goals rather than collective goals. As such, it becomes imperative to investigate collaborative strategies that will improve the total system's EE while completing the coverage tasks under dynamic settings.

3.3 Summary

In this chapter, we presented the system model used throughout this thesis. We formulated the problem as a multi-UAV MARL-based optimisation problem. In the next chapter, we present the design for the Decentralised Multi-Agent Reinforcement Learning (DMARL) solution for UAV-assisted networks.

Chapter 4

DMARL for UAV-Assisted Networks

In the previous chapter, we presented the environment model and formulated the optimisation problem. In this chapter, we present a set of requirements for the Decentralised Multi-Agent Reinforcement Learning (DMARL) solution to allow each UAV equipped with an autonomous agent to intelligently serve ground users while improving the overall system's energy efficiency (EE) in a shared, dynamic and interference-limited network environment. We then present the design of the DMARL algorithm, to minimise the total energy consumed by UAVs while providing wireless connectivity to ground users. We further decompose the DMARL into five variants, as motivated by these requirements. First, we present the variant that investigates how multiple UAVs, each with an independent learning agent learn a policy that minimises the total energy consumed while serving static and mobile ground users without the knowledge of the users' locations from a Central Controller (CC). Next, we discuss two collaborative variants, direct and indirect, that attempt to improve the system's EE in a shared, dynamic and interference-limited network environment. We then present the fourth and fifth variants that allow UAVs to be density-aware by collaborating to intelligently serve dense and uneven distributed users in the environment. Lastly, we provide the complexity analysis of our algorithm.

4.1 Requirements for DMARL in Shared, Dynamic and Interference-Limited Environments

In this section, we present the requirements for DMARL in order for UAVs to provide ubiquitous coverage to ground users in a shared, dynamic and interference-limited network environment. Most multi-agent systems (MAS), like the multi-UAVs systems, are geographically distributed [Liu et al., 2019a, Cui et al., 2020, Liu et al., 2020, Chen et al., 2022], thereby making it difficult for centralised control due to scalability and real-time adaptivity concerns [Foerster et al., 2017, Lowe et al., 2017]. More specifically, in disasters, it may be impractical for a CC to provide periodic updates to UAVs while serving dynamic users. In particular, a failure in the CC may result in downtime of network service, thereby affecting the entire network via a potential single point of failure. Therefore, it becomes imperative to locally optimise the behaviour of individual UAVs to improve global coverage performance while minimising total energy consumption. However, it is challenging to provide coverage to ground users without having knowledge of their locations. As such, the design of a fully-decentralised algorithm must entail no dependency on a CC.

As agent-controlled UAVs are deployed in a shared network environment to provide wireless connectivity to ground users, they may experience interference from nearby UAV cells sharing the same frequency band, thereby impacting the overall system's EE. More specifically, the interference poses some challenges to the performance of the UAV-assisted network. Agent-controlled UAVs that are deployed to serve the ground users may follow policies that try to maximise their individual goals. However, these UAVs may impact on the performance of other UAVs while trying to maximise their goals. This results in the phenomenon called non-stationarity. As stated in Chapter 2, non-stationarity could be addressed via collaboration. Agent-controlled UAVs could benefit from collaboration with neighbours. However, designing a collaborative MARL algorithm in a decentralised environment is challenging as highlighted in Chapter 2. Furthermore, several other issues may need to be addressed to achieve collaboration among UAVs, such as how and what information is shared. As motivated in Chapter 2, collaboration needs to be allowed among agent-controlled UAVs. In

urban areas, with uneven distribution of users, such as road networks, with varying numbers of vehicles, it becomes important for agent-controlled UAVs to be aware of areas with high concentrations of ground users and find ways of intelligently serving such users. Achieving this will require a robust and adaptive algorithm that will account for past coverage performances locally experienced.

Based on these observations we derive a set of requirements for a Decentralised Multi-Agent Reinforcement Learning (DMARL) algorithm in shared, dynamic and interference-limited network environments:

1. **R1:** Decentralised control with fully-autonomous agents (without reliance on a CC for decision-making).
2. **R2:** Support for collaborative behaviours among agent-controlled UAVs in a shared environment with agents with conflicting policies (We understand that each agent-controlled UAV follows its own policy).
3. **R3:** Support for agent-controlled UAVs to directly interact with neighbours to improve the total system's EE.
4. **R4:** Support for agent-controlled UAVs to provide coverage to locations with concentrated users in the network.

The remainder of this chapter presents the DMARL solution and analyses how its design addresses the above-specified requirements. First, we present an overview of DMARL and then introduce its variants. Lastly, we present the complexity of the algorithm.

4.2 DMARL Design

In this section, we use the requirements specified above to derive the design of DMARL. We consider five DMARL variants and evaluate them against the requirements, and examine their suitability in shared, dynamic and interference-limited environments. The design of the DMARL can be decomposed into five variants to answer our main research question (RQ)[1].

[1] **RQ:** How to minimise the total energy consumed by UAVs while providing wireless connectivity to mobile ground users in an interference-limited network environment?

To effectively address our overarching RQ, we split it into three RQs. The first variant investigates how multiple UAVs, each with an independent learning agent learn a policy that minimises the total energy consumed while serving static and mobile ground users without the knowledge of the users' locations from a CC. An agent-controlled UAV can have a wider view of its environment by gaining more knowledge for better decisions when information is exchanged with closest neighbours. Therefore, we present two collaborative variants, direct and indirect, to improve the system's EE in a shared, dynamic and interference-limited network environment. The direct collaboration allows UAVs to share their telemetry (which involves sharing their coordinates and sensed observations) via existing 3GPP guidelines [3GPP, 2008, 3GPP, 2019], while the indirect variant has no such mechanism but implicitly reflects this knowledge in its reward formulation as an incentive towards collaborative behaviours as highlighted in Chapter 2 [Panait and Luke, 2005]. More importantly, UAVs' past coverage performance may influence their decision to collaborate while serving users in dense and uneven user distribution. Therefore, we present the fourth and fifth variants that allow UAVs to be density-aware by collaborating to intelligently serve densely distributed users in urban environments. Next, we present the variants as they answer our three RQs while addressing the requirements listed in Section 4.1.

4.2.1 Independent Agents with No Central Controller

This variant investigates how multiple agent-controlled UAVs learn a policy that minimises the total energy consumed while serving static and mobile ground users without the knowledge of the users' locations from a CC. It is well known that RL can easily be extended to multiple independent agents [Tan, 1993]. The deployment of multiple UAVs in a dynamic and quickly-evolving network makes energy estimation and planning complex and difficult. As such, we propose a DMARL variant [Busoniu et al., 2008] to improve the energy utilization of multiple UAVs, while maximizing the connectivity of both static and mobile ground users. In such environments, the agents share the common interest of maximizing wireless connectivity while improving energy utilization within the network. The naive approach to address multi-agent problems is to consider each agent individually such that other agents are perceived as part

of the environment [Gronauer and Diepold, 2022, Shi et al., 2022].

From Chapter 2, a MARL algorithm can be considered an independent learner (IL) algorithm if the agents learn Q-values for their actions based on Equation (4.1) [Claus and Boutilier, 1998]. We overcome key MARL challenges highlighted in Chapter 2. First, by assuming that each agent has full local observability from the environment. Furthermore, the *computational complexity* in this decentralised setting is reduced with our IL agents. We do not assume the presence of a CC or a central agent for periodic updates of decision-making. The *credit assignment* problem is addressed with our design, with each agent's goal to optimise its individual reward and neighbour reward, which are mapped to its overall reward. In this work, we focus on agents with local observability called *ILs*, since the assumption of joint action observability is unrealistic without central/global knowledge [Busoniu et al., 2008]. Recall the Q-Learning (QL) update for agent j presented in Chapter 2.

$$Q_j(s_j, a_j) \leftarrow (1-\alpha)Q_j(s_j, a_j) + \alpha\Big[r_j + \gamma \max_{a'_j} Q_i(s'_j, a'_j)\Big], \tag{4.1}$$

where s_j is the present local state observed by agent j, s'_j is the new local state observed by agent j, a_j is the action taken by agent j, r_j is the reward received by agent j in that time step, α is the learning rate and $\gamma \in [0, 1]$ is the discount factor. We assume that each UAV is equipped with an autonomous agent which takes an action and in turn, receives a reward and makes a transition to a new state as shown in Figure 4.1. We explicitly define the states, actions, and reward function of our agent.

4.2.1.1 DQLSI State Space

We consider a combination of the three-dimensional (3D) position of the UAV [Liu et al., 2019a] and the distance from neighbouring UAVs. This helps to inform the agent of its position and that of neighbouring agents in each time step. UAVs can independently get information about neighbouring UAVs through things like positioning beacons[2] which will become mandatory for all UAVs soon [Poudel and Moh, 2019]. In particular, a CC may

[2]Beacons are primarily radio signals that show the proximity or location of a device or its readiness to perform a task [Gerasenko et al., 2001].

not be required for UAVs to be informed about the present location of their neighbours. The state-space is expressed as a tuple, $\langle x^t : \{x_{min}, ..., x_{max}\},\ y^t : \{y_{min}, ..., y_{max}\},\ h^t : \{h_{min}, ..., h_{max}\},\ N_d : \{N_d^1, N_d^2, ...\} \rangle$, where x_{min}, y_{min}, h_{min} and x_{max}, y_{max}, h_{max} are the minimum and maximum 3D coordinates of the considered geographical space, respectively. N_d is the distance between the UAV and its neighbours. Recall from Chapter 2 that the QL algorithm utilises discrete states and action spaces. Therefore, we discretise the continuous state space emanating from the environment into discrete state spaces in order to reduce the state-action space. We understand that discretising the state space may not yield so desired level of accuracy, as such, in the subsequent sections, we look into methods that allow for continuous state observations to improve ion the location accuracy.

4.2.1.2 Action space

At each time-step $t \in T$, each UAV executes an action by changing its direction along the 3D coordinates. We discretise the agent's actions following the design in [Liu et al., 2019a], as follows: $(+x_s,\ 0,\ 0)$, $(-x_s,\ 0,\ 0)$, $(0, +y_s,\ 0)$, $(0, -y_s,\ 0)$, $(0,\ 0, +z_s)$, $(0,\ 0, -z_s)$ and $(0,\ 0,\ 0)$. Our rationale to discretise the action space was to ensure that the agents quickly adapt and converge to an optimal policy.

4.2.1.3 Reward

The goal of the agent is to learn a policy that maximises the number of connected ground users while minimising the total UAVs' energy consumption. As stated in Chapter 2, we want to ensure that each agent is rewarded based on its performance, while also addressing the *lazy agent problem*. Therefore, we formulate the reward function in such a way that each agent j is given a '+1' when the connectivity score in the present time-step C_j^t is greater than that in the previous time-step C_j^{t-1}. If C_j^t is equal to C_j^{t-1}, we assign a '0' reward, otherwise we assign a '-1' reward. Furthermore, we introduce ω which gives a reflection of the energy consumption by each UAV, and it is a function of the instantaneous energy consumed in the present and previous time-step. As discussed in Chapter 2, agents may be rewarded based on the performance locally. Hence, we introduce a shared collaborative factor $\mho$ to shape the

reward formulation of each agent j in each time-step $t \in T$ given as,

$$\mathcal{R}_j^t = \begin{cases} \mho + \omega + 1, & \text{if } C_j^t > C_j^{t-1} \\ \mho + \omega, & \text{if } C_j^t = C_j^{t-1} \\ \mho + \omega - 1, & \text{otherwise,} \end{cases} \tag{4.2}$$

where C_j^t and C_j^{t-1} are the connectivity score in the present and previous time-step, respectively. $\omega = \frac{e_j^{t-1} - e_j^t}{e_j^t + e_j^{t-1}}$, where e_j^t and e_j^{t-1} are the instantaneous energy consumed by agent j in present and previous time-step, respectively. To enhance collaboration, we assign each agent a '+1' incentive via a function $\mho$ only when the overall connectivity score, which is the total number of connected users by UAVs in the present time-step C_o^t exceeds that in the previous time-step C_o^{t-1}, otherwise the agent receives a '-1' incentive. We compute $\mho$ as,

$$\mho = \begin{cases} +1, & \text{if } C_o^t > C_o^{t-1} \\ -1, & \text{otherwise.} \end{cases} \tag{4.3}$$

In our multi-UAV system, we propose an algorithm called Decentralised Q-learning with Local Sensory Information (DQLSI) as shown in Algorithm 1 with strict local observability suitable when there is limited or no access to cellular infrastructure due to disaster. As stated earlier in Chapter 1, we assume the presence a of back-haul that allows the UAVs connect to the internet via satellite. However, our focus will not be on optimising the back-haul link. From Algorithm 1, we map local observations emanating from Agent j's environment to discrete states on Line 8. On Line 10, Agent j selects an action following an ϵ-greedy policy and then executes the action as seen on Line 12. Agent j then receives a reward and observes a new state. The learning procedure for Agent j is shown on Line 19.

Through this variant, we address requirement R1 as specified in Section 4.1 via our contribution C1 while proffering an answer to our first research question RQ1[3]. This variant satisfies

[3]**RQ1:** Can UAVs serving mobile ground users improve the total system's energy efficiency in a shared, dynamic and interference-limited network environment without relying on a central controller for decision-making?

Algorithm 1 Decentralised Q-learning with Local Sensory Information (DQLSI) for Agent j

Input: UAV3Dposition $(x_j^t,\ y_j^t,\ h_j^t)$, UAVneighbourProximity $N_d^t \in S$ and Output: Q-values corresponding to each possible action $(+x_s,\ 0,\ 0)$, $(-x_s,\ 0,\ 0)$, $(0, +y_s,\ 0)$, $(0, -y_s,\ 0)$, $(0,\ 0, +z_s)$, $(0,\ 0, -z_s)$, $(0,\ 0,\ 0) \in A_j$
for all $a_j \in A_j$ and $s_j \in S_j$ **do**:
    $Q_{j,max}(s_j, a_j) \leftarrow 0$, $\pi_j(s_j, a_j)$ arbitrarily
    $s_j \leftarrow$ initial state
    $1500 \leftarrow$ maxStep
        ▷ `An episode ends when` *`goal`* `is Reached or UAV` *`dies`* `or maxStep is reached. maxStep value was gotten after several experimentation to ensure that agents converge to optimal policies.`
    while *goal* not Reached and Agent *alive* and maxStep not reached **do**
        $s_j \leftarrow$ MapLocalObservationToState(*Env*)
            ▷ `From` s_j `select` a_j `according to` ϵ`-greedy method based on` π_j
        $a_j \leftarrow$ QLearning.SelectAction(s_j)
            ▷ `AgentExecutesActionInState`
        a_j.execute(*Env*)
        if a_j.execute(*Env*) is True **then**
            ▷ `MapToNewState`
            Env.UAV3Dposition
            Env.UAVneighbourProximity
            ▷ `Observe reward` r `and next state` s'
        UpdateQLearningProcedure() using Equation (4.1)
        $Q_j(s_j, a_j) \leftarrow (1-\alpha)Q_j(s_j, a_j) + \alpha\left[r_j + \gamma \max_{a'_j} Q_i(s'_j, a'_j)\right]$
        $s_j \leftarrow s'_j$
    endwhile

the requirement R1 of ensuring decentralised control of fully-autonomous agents that do not rely on a CC for decision-making. Next, we present a DMARL variant that allows UAVs to collaborate to improve the total system's EE in a shared, dynamic and interference-limited network environment. This variant adopts a deep neural network architecture that allows for continuous state observations rather than discrete states as in the DQLSI algorithm. We also look towards introducing the connectivity score and the instantaneous energy consumption information into the agents' observations to improve the overall EE in the network while meeting the requirement for supporting collaborative behaviours among agent-controlled UAVs in a shared environment with agents with conflicting policies.

4.2.2 Collaborative Agents

In this section, we look into the design of collaborative agents that allows indirect collaboration which we present in Section 4.2.2.1 and direct collaboration which we present in Section 4.2.2.3.

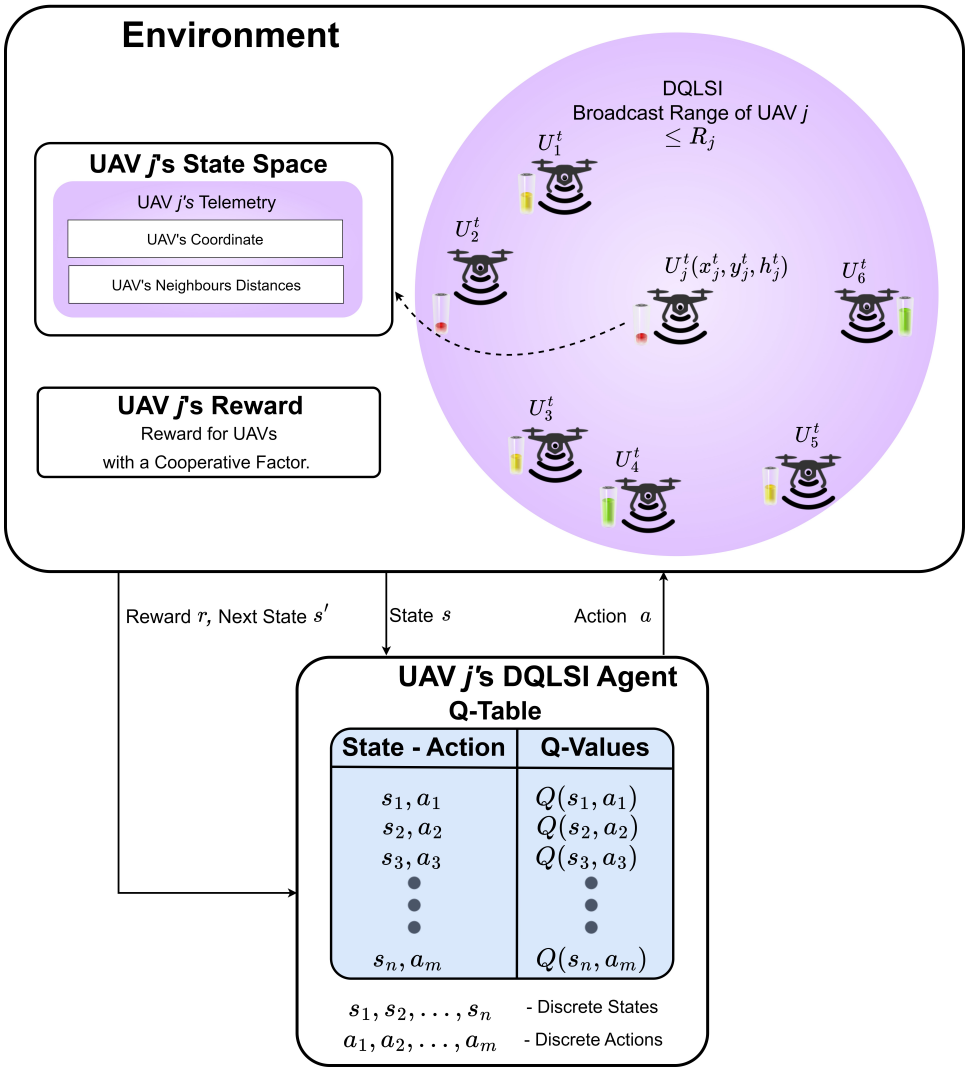


Figure 4.1: Decentralised Q-learning with Local Sensory Information (DQLSI) variant of DMARL where each UAV j equipped with a tabular Q-learning agent interacts with its environment, and provides wireless coverage to ground users without any feedback from a CC.

4.2.2.1 Collaborative Agents with Individual Knowledge

This DMARL variant investigates if and how collaboration among multiple UAVs can improve the total system's EE in a shared, dynamic and interference-limited network environment. In the *Independent Learning Agents* variant discussed in the previous Section, each agent-controlled UAV is assumed to know its position in space via a GPS mounted on the UAVs. In addition, each UAV can sense the broadcasted position information from nearby UAV cells. The agent-controlled UAV uses this local knowledge to execute decisions within its environment. Rather than have a CC or dedicated server that carries out decision-making via a

central learning agent, we propose a decentralised architecture of *Independent Learning Agent* who follow their policies to improve the overall performance in the network. We understand that collaboration in the *Independent Learning Agents* variant and proposed *Collaborative Agents with Individual Knowledge* variant may be achieved indirectly via the reward formulation, which provides some incentives for agent-controlled UAVs to collaborate in a shared, dynamic and interference-limited network environment. However, the DQLSI approach, due to its tabular Q–learning architecture, only supports discrete state observation. As such, agents only explore the available discrete states space which limits its applicability in real deployments scenario. Furthermore, as the number of state observations is increased to capture the characteristics of the agent's environment, it may be difficult to maintain the table entries in the DQLSI approach since the table mapping states to actions may become too large. Considering the importance for each UAV to be equipped with information about its coverage performance as well as its energy level, it becomes imperative for each agent-controlled UAV to have such knowledge captured in its state space. Considering the above limitations of the DQLSI approach, we look into DNN architectures that support continuous state space design. Hence, we propose the MAD–DDQN that captures the connectivity and energy level information of the agent-controlled UAVs. Overall, the *Independent Learning Agents* and the *Collaborative Agents with Individual Knowledge* variants are both indirect collaborative approaches, however, they differ in architectural design and state space composition.

As multiple wireless transmitters sharing the same frequency band are in close proximity to one another the possibility of interference is significantly increased. This interference significantly impacts the system's EE. Furthermore, it is important for UAVs to be able to infer the actions over time via collaborative mechanisms to minimise conflicting policies that degrade the system performance in the environment. This type of decentralised setting is partially observable, thereby making it challenging for agent-controlled UAVs to collaborate without any collaborative strategy [Panait and Luke, 2005]. The computational complexity of the problem in Equation (3.9a) is known to be NP-complete [Liu et al., 2019a].

The problem Equation (3.9a) is non-convex, thus having multiple local optima. For this reason, solving Equation (3.9a) is challenging. In particular, MARL has been shown to solve

complex problems in UAV-assisted networks [Liu et al., 2019a], [Liu et al., 2020]. Specifically, the problem in Equation (3.9a) will become more complex as more UAVs are deployed in a shared wireless environment, hence it is challenging to find the optimal collaborative strategies for UAVs to improve the system's EE while completing the coverage tasks under dynamic settings. This is often because UAVs may become selfish and pursue the goal of improving their individual EE while minimising the communication outage and energy consumption, rather than the collective goal of maximising the system's EE. Therefore, collaborative MARL strategies may be suitable when there is conflict in the individual and collective interest of agents [Panait and Luke, 2005]. Several works on collaborative MARL focus on equally-shared rewards among agents to motivate them to collaborate and try to avoid selfish behaviours that impact the overall performance [Gronauer and Diepold, 2022]. Specifically, it becomes imperative to explore strategies where agent-controlled UAVs are encouraged to collaborate but do not own an equally-shared reward, i.e., the reward assignment should fairly reflect both the individual and performances locally, and not necessarily be equal. On this note, we ensure that the *credit assignment* problem discussed in Chapter 2 is addressed with our design, with each agent's goal to optimise its individual reward and neighbour reward, which are mapped to its overall reward.

Figure 4.2 shows the MAD–DDQN framework where each DDQN agent-controlled UAV j interacts with its environment. We explicitly define the state space of our agent j.

4.2.2.2 MAD–DDQN State Space

We assume that Agent j acquires telemetry data via its sensors, which make up its state space. This variant considers the three-dimensional (3D) position of each UAV, the connectivity score and the UAV's instantaneous energy level at time t, expressed as a tuple, $\langle x^t : \{x_{min}, ..., x_{max}\},\ y^t : \{y_{min}, ..., y_{max}\},\ h^t : \{h_{min}, ..., h_{max}\},\ C_t,\ e_t \rangle$, where x_{min}, y_{min}, h_{min} and x_{max}, y_{max}, h_{max} are the minimum and maximum 3D coordinates of the considered geographical space, respectively.

Deep RL has been shown to perform well in decision-making tasks in UAV-assisted net-

Algorithm 2 Double Deep Q-Network (DDQN) for Agent j with Indirect Collaboration

Input: UAV3Dposition $(x_j^t,\ y_j^t,\ h_j^t)$, ConnectivityScore c_j^t, InstantaneousEnergyConsumed $e_j^t \in S$ and Output: Q-values corresponding to each possible action $(+x_s,\ 0,\ 0)$, $(-x_s,\ 0,\ 0)$, $(0, +y_s,\ 0)$, $(0, -y_s,\ 0)$, $(0,\ 0, +z_s)$, $(0,\ 0, -z_s)$, $(0,\ 0,\ 0) \in A_j$
for all $a \in A_j$ and $s \in S$ **do**:
    $Q_{(1)}(s,a)$, $Q_{(2)}(s,a)$, $\mathcal{D}$ – empty replay memory, θ – initial network parameters, θ^- – copy of θ, $\mathcal{N}_r$ – maximum size of replay memory, $\mathcal{N}_b$ – batch size, $\mathcal{N}^-$ – target replacement frequency.
    $s \leftarrow$ initial state, maxStep $\leftarrow$ maximum number of steps in the episode
    while *goal* not Reached and Agent *alive* and maxStep not reached **do**
        $s \leftarrow$ MapLocalObservationToState(*Env*)
            ▷ `Execute` ϵ-`greedy method based on` π_j
        $a \leftarrow$ DeepQnetwork.SelectAction(s)
            ▷ `Agent executes action in state` s
        a.execute(*Env*)
        if a.execute(*Env*) is True **then**
            ▷ `Map sensed observations to new state` s'
            Env.UAV3Dposition
            Env.ConnectivityScore using Equation (3.3)
            Env.InstantaneousEnergyConsumed using Equation (3.6)
        $r \leftarrow$ Env.RewardWithCollaborativeNeighbourFactor using Equation (4.2)
            ▷ `Execute UpdateDDQNprocedure()`
        Sample a minibatch of $\mathcal{N}_b$ tuples $(s,a,r,s') \sim Unif(\mathcal{D})$
        Construct target values, one for each of the $\mathcal{N}_b$ tuples:
        Define $a^{max}(s';\theta) = \arg\max_{a'} Q_{(1)}(s',a';\theta)$
        if s' is Terminal **then**
            $y_j = r$
        else
            $y_j = r + \gamma Q_{(2)}(s', a^{max}((s';\theta);\theta^-)$
        Apply a gradient descent step with loss $\| y_j - Q(s,a;\theta) \|^2$
        Replace target parameters $\theta^- \leftarrow \theta$ every $\mathcal{N}^-$ step
    endwhile

Introducing UAV's local observation

Deep Neural Network replaces the tabular Q-learning of the DQLSI variant

works [Liu et al., 2018, Liu et al., 2019a, Liu et al., 2020, Wang et al., 2021, Zhang et al., 2021b]. Hence, we adopt a collaborative deep MARL approach [Zhang et al., 2021a] to solve the system's EE optimisation problem. In our multi-UAV system, we present an algorithm called Multi-Agent Decentralised Double Deep Q-Network (MAD–DDQN), as shown in Algorithm 2, suitable to serve dynamic users when there is limited or no access to cellular infrastructure due to disaster. Through this variant, we address requirements R1 and R2 as specified in Section 4.1 via our contributions C1 and C2 while providing an answer to our second research question RQ2[4]. Here, each UAV is controlled by a Double Deep Q-Network (DDQN) agent that aims to maximise the system's EE by jointly optimising its 3D trajectory,

[4]**RQ2:** Can collaboration with closest neighbours improve the total system's energy efficiency while minimising the total energy consumed by UAVs in a shared, dynamic and interference-limited network environment?

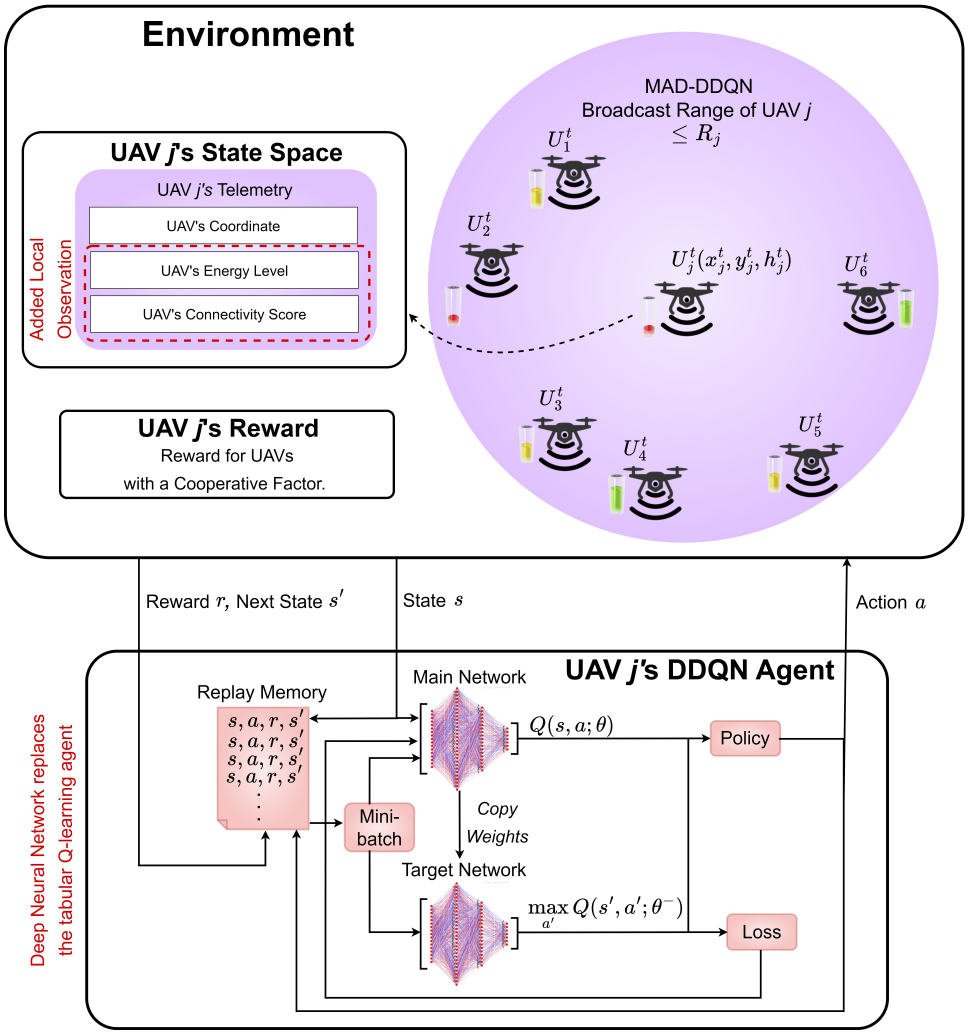


Figure 4.2: Multi-agent decentralised double deep Q-network (MAD-DDQN) framework where each UAV j equipped with a DDQN agent interacts with its environment. Each UAV indirectly collaborates via a reward [Wu et al., 2021] that reflects the coverage performance locally to improve overall EE in the network.

the number of connected users, and the energy consumed.

We assume the agents interact with each other in a shared and dynamic environment. This interaction improves the learning behaviour and addresses instabilities due to conflicting policies from other agents, hereby addressing the *non-stationarity* challenge. We address the *credit assignment* problem by ensuring that the agent-controlled UAVs are incentivised to collaborate via collaborative factor embedded in the reward formulation [Wu et al., 2021]. Furthermore, the authors [Zhang et al., 2020b] argue that collaboration among agents can be achieved naturally if each agent acts independently following its value function, by executing an action that leads to a state that is perceived to be more rewarding to itself than other

agents. As discussed in Chapter 2, the work [Freed et al., 2022] reiterated that optimal collaboration among agents can be achieved if each agent can collaborate with its neighbours. Throughout this thesis, we consider agent-controlled UAVs that are in close proximity and transmission range T_{range} of each other as neighbours. The transmission range defines the coverage range of the UAVs. That is, an agent-controlled UAV may be unable to communicate or share information to other UAVs that are beyond its transmission range. Importantly, collaboration among agent-controlled UAVs can be achieved by shaping the agents' reward to not only capture the agents' individual coverage but also reflect the coverage performance locally [Lowe et al., 2017]. From Algorithm 2, Agent j follows an ϵ–greedy policy by executing an action a, transiting from state s to a new state s' and receiving a reward given in Equation (4.2), after which DDQN procedure described on line 18–26 optimises the agent's decisions.

Despite the potential impact of this indirect collaborative variant with no communication overhead, in the next sub-section, we explore the impact of direct communication in our collaborative DMARL design on improving the total system's EE in a shared, dynamic and interference-limited network environment.

4.2.2.3 Collaborative Agents with Neighbour Knowledge

Previously, we presented the design of the collaborative MAD–DDQN algorithm that has no defined mechanism for communication. Hence, its state observation was made up of self-sensed information. Despite the DQLSI and MAD–DDQN variants having a collaborative mechanism via shaping the reward to reflect the coverage performance locally [Jaques et al., 2018], direct communication among agents may further improve collaboration [Kim et al., 2019a, Simoes et al., 2020]. The work [Zhao et al., 2022] proposed a fully distributed approach that allows agents to share information with their neighbours through a communication network and executes decisions based on its local reward and information received from their neighbours. We adopt a communication-enabled MARL design [Zhu et al., 2022] that can improve collaboration among agent-controlled UAVs while serving ground users in an energy-efficient manner.

It is expected that UAVs will be legally required to broadcast their telemetry information for safety reasons, which involves sharing their coordinates, UAV identification, flight plans (or rather velocity and direction, for security and privacy reasons), vehicle type [Vinogradov et al., 2020]. Specifically, agent-controlled UAVs can share (via direct communication) their telemetry information (i.e., coordinates, connectivity score and energy level) with closest neighbours to improve the network performance. This communication can be done through standardised WiFi or possibly 3GPP sidelink communication (enabling D2D[5] communications without going through the network infrastructure). In this type of wireless network the agent-controlled UAVs only need to communicate with their nearest neighbours (typically within proximity) [Goldsmith and Wicker, 2002], to share local observations that could improve the learning.

Enabling communication can provide better insights to other agents in the environment, especially in disaster response operations where multiple agent-controlled UAVs are deployed [Lee and Lee, 2021] and are required to collaborate to accomplish a given task. In this work, we assume that each agent-controlled UAV has full local observability and gains additional knowledge of its environment through direct interaction with its neighbours. Nevertheless, the direct collaborative communication-enabled MAD–DDQN variant introduces some communication overhead when compared to the indirect collaborative MAD–DDQN variant. The computational complexity of the problem in Equation (3.9a) is known to be NP-complete [Liu et al., 2019a]. Nevertheless, we can see a reduction in the complexity when the agents fully share their observations in every step [Becker et al., 2004]. However, sharing all observations will result in increased communication overhead. Later on in Chapter 6, we provide analysis on the overhead incurred by UAVs in communication.

Here, we assume that each UAV is equipped with an autonomous agent which takes an action and in turn, receives a reward and makes a transition to a new state. Figure 4.3 shows the CMAD–DDQN framework where each DDQN agent-controlled UAV j interacts with its environment. We explicitly define the state space of our agent j.

[5]Device-to-Device (D2D) is a radio technology that enables devices to communicate with each other with or without the involvement of network infrastructures such as an access point or base stations.

4.2.2.4 CMAD–DDQN State Space

We consider the three-dimensional (3D) position of each UAV, the UAV's connectivity score, the UAV's instantaneous energy level, closest neighbour distances using a defined communication mechanism, the neighbour connectivity score, and neighbour instantaneous energy consumed at time t, expressed as a tuple, $\langle x^t : \{x_{min}, ..., x_{max}\},\ y^t : \{y_{min}, ..., y_{max}\},\ h^t : \{h_{min}, ..., h_{max}\},\ C_j^t,\ e_j^t,\ N_d^t,\ C_z^t,\ e_z^t \rangle$, where x_{min}, y_{min}, h_{min} and x_{max}, y_{max}, h_{max} are the minimum and maximum 3D coordinates of the considered geographical space, respectively. N_d^t is the distance of neighbouring UAVs, C_z^t is the connectivity score of neighbouring UAVs, and e_z^t is the instantaneous energy level of neighbouring UAVs. However, this variant introduces some communication overhead since each agent's state space is comprised of communicated observations from neighbours as seen in Figure 4.3. The communication cost incurred by each sensory-exchanging agent per step is bounded by $U_L(t)_j \times E$ (Refer to Case-study 2, [Tan, 1993]), where $U_L(t)_j$ is the number of neighbours of agent-controlled UAV j at time t, E is the number of bits needed to represent each observation by the agent. Through the proposed direct collaborative variant, we address requirements R1, R2 and R3 as specified in Section 4.1 via our contributions C1 and C2 while providing an answer to our second research question RQ2[6]. Here, each UAV is controlled by a Double Deep Q-Network (DDQN) agent that aims to maximise the system's EE by jointly optimising its 3D trajectory, the number of connected users, and the energy consumed.

Hence, we propose a collaborative CMAD-DDQN variant that relies on a communication mechanism among neighbouring UAVs for improved system performance. Note that we assume a lossless wireless channel that allows observations sent to other agent-controlled UAVs to be received in good condition, and without any delay or distortion. In the scenario we consider, each agent's reward reflects the coverage performance locally. As seen in Figure 4.3, each UAV is controlled by a Double Deep Q-Network (DDQN) agent that aims to maximise the system's EE by jointly optimising its 3D trajectory, number of connected ground users,

[6] **RQ2:** Can collaboration with closest neighbours improve the total system's energy efficiency while minimising the total energy consumed by UAVs in a shared, dynamic and interference-limited network environment?

and the energy consumed by the UAVs. We assume that as the agents interact with each other in a shared and dynamic environment, they may observe learning instabilities due to conflicting policies from other agents. Algorithm 3 shows the DDQN for Agent j with direct collaboration with its neighbours. Agent j follows an ϵ–greedy policy by executing an action a in its present state s after which it transits to a new state s' and receives a reward that reflects the coverage performance locally as given in Equation (4.2). Furthermore, the DDQN procedure described on line 23–31 optimises the agent's decisions.

Algorithm 3 Double Deep Q-Network (DDQN) for Agent j with Direct Collaboration with its Neighbours

1: Input: UAV3Dposition $(x_j^t,\ y_j^t,\ h_j^t)$, ConnectivityScore c_j^t, InstantaneousEnergyConsumed e_j^t, UAVneighbourDistances N_d^t, NeighboursConnectionScore c_z^t, NeighboursInstantaneousEnergyConsumed e_z^t $\in S$ and Output: Q-values corresponding to each possible action $(+x_s,\ 0,\ 0)$, $(-x_s,\ 0,\ 0)$, $(0, +y_s,\ 0)$, $(0, -y_s,\ 0)$, $(0,\ 0, +z_s)$, $(0,\ 0, -z_s)$, $(0,\ 0,\ 0) \in A_j$.
2: **for all** $a \in A_j$ and $s \in S$ **do**:
3: $Q_{(1)}(s,a)$, $Q_{(2)}(s,a)$, $\mathcal{D}$ – empty replay buffer, θ – initial network parameters, θ^- – copy of θ, N_r – maximum size of replay buffer, N_b – batch size, N^- – target replacement frequency.
4: $s \leftarrow$ initial state
5: 1500 $\leftarrow$ maxStep
6: **while** *goal* not Reached and Agent *alive* and maxStep not reached **do**
7: $s \leftarrow$ MapLocalObservationToState(*Env*)
8: ▷ `Execute` ϵ-`greedy method based on` π_j
9: $a \leftarrow$ DeepQnetwork.SelectAction(s)
10: ▷ `Agent executes action in state` s
11: a.execute(*Env*)
12: **if** a.execute(*Env*) is True **then**
13: ▷ `Map sensed observations to new state` s'
14: Env.UAV3Dposition
15: Env.ConnectivityScore using Equation (3.3)
16: Env.InstantaneousEnergyConsumed using Equation (3.6)
17: ▷ `Map communicated observations from closest neighbours based on an existing ANR mechanism for UAV communication to new state` s'
18: Env.Neighbour.UAVneighbourDistances
19: Env.Neighbour.ConnectionScore
20: Env.Neighbour.InstantaneousEnergyConsumed
Introducing information received from UAV's closest neighbours
21: $r \leftarrow$ Env.RewardWithCollaborativeNeighbourFactor using Equation (4.2)
22: ▷ `Execute UpdateDDQNprocedure()`
23: Sample a minibatch of N_b tuples $(s, a, r, s') \sim Unif(\mathcal{D})$
24: Construct target values, one for each of the N_b tuples:
25: Define $a^{max}(s';\theta) = \arg\max_{a'} Q_{(1)}(s', a';\theta)$
26: **if** s' is Terminal **then**
27: $y_j = r$
28: **else**
29: $y_j = r + \gamma Q_{(2)}(s', a^{max}((s';\theta);\theta^-)$
30: Apply a gradient descent step with loss $\| y_j - Q(s,a;\theta) \|^2$
31: Replace target parameters $\theta^- \leftarrow \theta$ every N^- step
32: **endwhile**

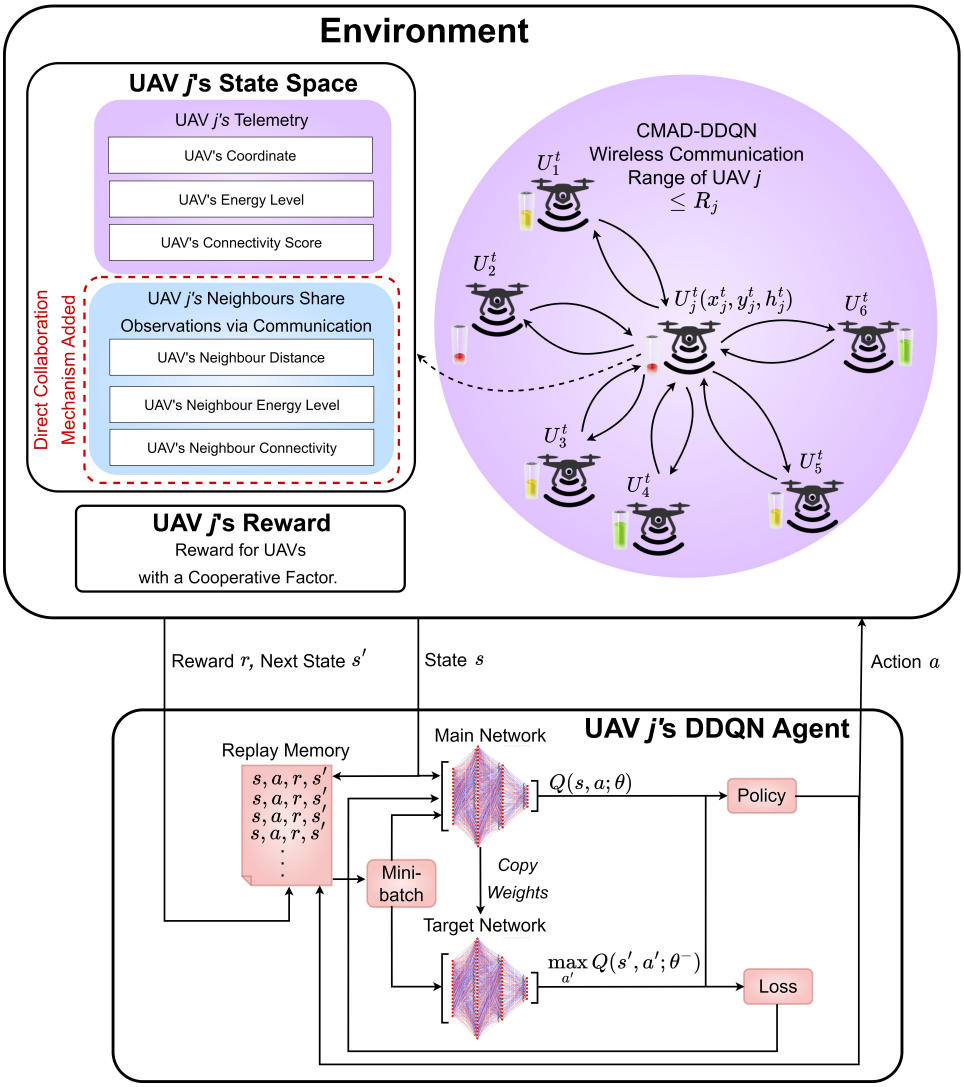


Figure 4.3: Communication-enabled multi-agent decentralised double deep Q-network (CMAD-DDQN) framework where each UAV j equipped with a DDQN agent interacts and shares knowledge with its nearest neighbours which makes up the state space. Each UAV directly collaborates to improve overall system performance.

More importantly, an agent-controlled UAV's past coverage performance may influence its decision to collaborate while serving users in dense and uneven user distribution. As such, it is imperative to explore variants of design that are suitable in such scenarios. In the next sub-sections, we aim to explore how collaborative density-aware DMARL designs improve the total system's energy efficiency in a shared, dynamic and interference-limited network environment.

4.2.3 Collaborative Density-Aware Agents

In this section, we look into the design of collaborative density-aware agents that allows indirect collaboration which we present in Section 4.2.3.1 and direct collaboration which we present in Section 4.2.3.5.

4.2.3.1 Collaborative Density-Aware Agents with Individual Knowledge

Previously, we presented the DMARL variants design that did not consider the knowledge of dense users' regions in the network for the agent-controlled UAVs serving ground users. In particular, such knowledge may be crucial for agent-controlled UAVs to serve highly mobile and uneven user distributions. Moreover, Collaborating agents may require past histories of observations to be fully synchronised and the information corresponding to the union of the histories of observations provides the agents with information about their current global state [Goldman and Zilberstein, 2003, Oliehoek and Spaan, 2012, Oliehoek and Amato, 2016]. Here, we propose the DMARL variant with *density-aware indirect collaborative agents* that is suitable in disaster scenarios with no communication overhead as compared to the CMAD–DDQN algorithm.

Our density-aware MAD–DDQN variant called DAMAD–DDQN enables each agent to observe its best-experienced connectivity score and the position where it experienced the best number of connected users to keep track of dense users' areas in the network. As stated earlier both the MAD–DDQN and CMAD–DDQN algorithms of DMARL do not have a mechanism that allows an agent to keep track of their past experiences, especially the history of dense regions in the network. On this note, we present a DAMAD–DDQN algorithm that allows the agent-controlled UAVs to keep track of such information and use this knowledge to improve the network performance in dense and uneven distribution of users. Figure 4.4 shows the DAMAD–DDQN framework where each DDQN agent-controlled UAV j interacts with its environment. We explicitly define the state space, actions space, and reward function of our agent j.

4.2.3.2 DAMAD–DDQN state space

We assume that Agent j acquires telemetry data via its sensors, which constitutes its state space. This variant considers the UAV's three-dimensional (3D) coordinates, the connectivity score, the UAV's instantaneous energy level, the ratio of the connectivity score to the best connectivity score experienced by UAV j at time-step t, and the coordinates where the UAV experienced its best connectivity score, which is made up the state space and is expressed as a tuple, $\langle x^t : \{x_{min}, ..., x_{max}\}, y^t : \{y_{min}, ..., y_{max}\},\ h^t : \{h_{min}, ..., h_{max}\},\ C_j^t,\ e_j^t,\ \frac{C_j^t}{C_j^*},\ x^*,\ y^* \rangle$, where x_{min}, y_{min}, h_{min} and x_{max}, y_{max}, h_{max} are the minimum and maximum 3D coordinates of the considered geographical space, respectively. $\frac{C_j^t}{C_j^*}$ is the ratio of the connectivity score to the best connectivity score experienced by the UAV over a series of past encounters. The x^* and y^* are the coordinates where the UAV experienced its best connectivity score.

4.2.3.3 Action space

At each time-step $t \in T$, each UAV executes an action by changing its direction along the coordinates: $(+x_s,\ 0)$, $(-x_s,\ 0)$, $(0, +y_s)$, $(0, -y_s)$, and $(0,\ 0)$. The intuition behind restricting the UAVs' actions along the x and y direction was to ensure that the UAVs make an effort to move towards dense locations rather than increasing their altitude to cover as many users.

4.2.3.4 Reward

The goal of the agent is to learn a policy that implicitly maximises the system's EE by jointly maximising the number of connected users while minimising the total UAVs' energy consumption. Here, we want to ensure that each agent is rewarded based on its performance and its ability to improve such performance based on its past experiences, while also addressing the *lazy agent problem*. Rather than assigning a +1' reward when the connectivity score in the present time-step C_j^t is greater than that in the previous time-step C_j^{t-1} as in the previous variants, we assign each agent j a '$+\frac{C_j^t}{C_j^*}$' reward. Similarly, if C_j^t is equal to C_j^{t-1}, we assign a '0' reward, otherwise we assign a '$-\frac{C_j^t}{C_j^*}$' reward.

The rationale for replacing the bipolar rewards with the ratio of the agent's present coverage performance with respect to its past best coverage performance is to motivate each UAV to pursue a goal of improving its individual best of maximising the number of connected ground users over a series of time-steps. Furthermore, we introduce ω which gives a reflection of the energy consumption by each UAV, and it is a function of the instantaneous energy consumed in the present and previous time-step. As discussed in Chapter 2, agents may be rewarded based on the performance locally. Hence, we redefine the shared collaborative factor $\mho$ to shape the reward formulation of each agent j in each time-step $t \in T$ given as,

$$\mathcal{R}_j^t = \begin{cases} \mho + \omega + \frac{C_j^t}{C_j^*}, & \text{if } C_j^t > C_j^{t-1} \\ \mho + \omega, & \text{if } C_j^t = C_j^{t-1} \\ \mho + \omega - \frac{C_j^t}{C_j^*}, & \text{otherwise,} \end{cases} \tag{4.4}$$

where C_j^*, C_j^t, and C_j^{t-1} are the best connectivity score ever experienced by Agent j during the learning cycle, connectivity score in the present and previous time-step, respectively. $\omega = \frac{e_j^{t-1} - e_j^t}{e_j^t + e_j^{t-1}}$, where e_j^t and e_j^{t-1} are the instantaneous energy consumed by Agent j in present and previous time-step, respectively. To enhance collaboration while motivating the agents to pursue a goal of providing coverage to dense areas, we compute $\mho$ as,

$$\mho = \begin{cases} +\frac{C_o^t}{C_o^*}, & \text{if } C_o^t > C_o^{t-1} \\ -\frac{C_o^t}{C_o^*}, & \text{otherwise.} \end{cases} \tag{4.5}$$

Through this variant, we address requirements R1, R2 and R4 as specified in Section 4.1 via our contributions C1, C2 and C3 while providing an answer to our third research question RQ3[7]. We assume that each UAV is equipped with a Double Deep Q-Network (DDQN) agent which can learn the density of users covered by itself, and then adjust its trajectory in such a way that will maximise the total system's EE while jointly optimising the total number of connected vehicles and the energy utilisation of the UAV. As earlier stated, it is often

[7]**RQ3:** Can UAVs collaborate intelligently to improve the total system's energy efficiency in highly mobile, dense and unevenly distributed users in an urban environment?

difficult to achieve collaboration in a typical multi-agent setting [Dafoe et al., 2020] since the interference-limited environment pushes agents to exhibit some selfish behaviors. Therefore, a robust and adaptive strategy is required to allow agents to collaborate while completing their tasks.

Algorithm 4 Double Deep Q-Network (DDQN) for Agent j with Density-Awareness and no Direct Communication Mechanism

Input: UAV3Dposition $(x_j^t,\ y_j^t,\ h_j^t)$, ConnectivityScore c_j^t, InstantaneousEnergyConsumed e_j^t, $\frac{c_j^t}{c_j^*}$, ExperiencedDensePosition $(x_j^*,\ y_j^*) \in S$ and Output: Q-values corresponding to each possible action $(+x_s,\ 0)$, $(-x_s,\ 0)$, $(0, +y_s)$, $(0, -y_s)$, $(0,\ 0) \in A_j$. Given the ConnectivityScore c_j^t, PastBestConnectivityScore c_j^*, NeighbourConnectivityScore c_o^t, BestNeighbourConnectivityScore c_o^* .

for all $a \in A_j$ and $s \in S$ **do**:

  $Q_{(1)}(s,a)$, $Q_{(2)}(s,a)$, $\mathcal{D}$ – empty replay buffer, θ – initial network parameters, θ^- – copy of θ, N_r – maximum size of replay buffer, N_b – batch size, N^- – target replacement frequency.

  $s \leftarrow$ initial state

  1500 $\leftarrow$ maxStep

  while *goal* not Reached and Agent *alive* and maxStep not reached **do**

    $s \leftarrow$ MapLocalObservationToState(*Env*)

      ▷ `Execute` ϵ-`greedy method based on` π_j

    $a \leftarrow$ DeepQnetwork.SelectAction(s)

      ▷ `Agent executes action in state` s

    a.execute(*Env*)

    if a.execute(*Env*) is True **then**

      ▷ `Map sensed observations to new state` s'

      Env.UAVposition

      Env.ConnectivityScore using Equation (3.3)

      Env.InstantaneousEnergyConsumed using Equation (3.6)

      Env.RatioOfConnectivityScore ToPastBestConnectivityScore — *Density-aware feature added to MAD–DDQN variant*

      Env.ExperiencedDensePosition

    $r \leftarrow$ Env.RewardWithCollaborativeNeighbourFactor using Equation (4.2)

    update $(x_j^*,\ y_j^*)$, c_j^*, c_o^* $\forall\ t$ — *Update knowledge locally*

    if $c_j^t > c_j^*$ **then**

      $(x_j^*,\ y_j^*) \leftarrow (x_j^t,\ y_j^t)$

      $c_j^* \leftarrow c_j^t$

    if $c_o^t > c_o^*$ **then**

      $c_o^* \leftarrow c_o^t$

      ▷ `Execute UpdateDDQNprocedure()`

    Sample minibatch of N_b tuples $(s,a,r,s') \sim Unif(\mathcal{D})$

    Construct target values, one for each of the N_b tuples:

    Define $a^{max}(s';\theta) = \arg\max_{a'} Q_{(1)}(s',a';\theta)$

    if s' is Terminal **then**

      $y_j = r$

    else

      $y_j = r + \gamma Q_{(2)}(s', a^{max}((s';\theta);\theta^-)$

    Apply gradient descent step with loss $\| y_j - Q(s,a;\theta) \|^2$

    Replace target parameters $\theta^- \leftarrow \theta$ every N^- step

  endwhile

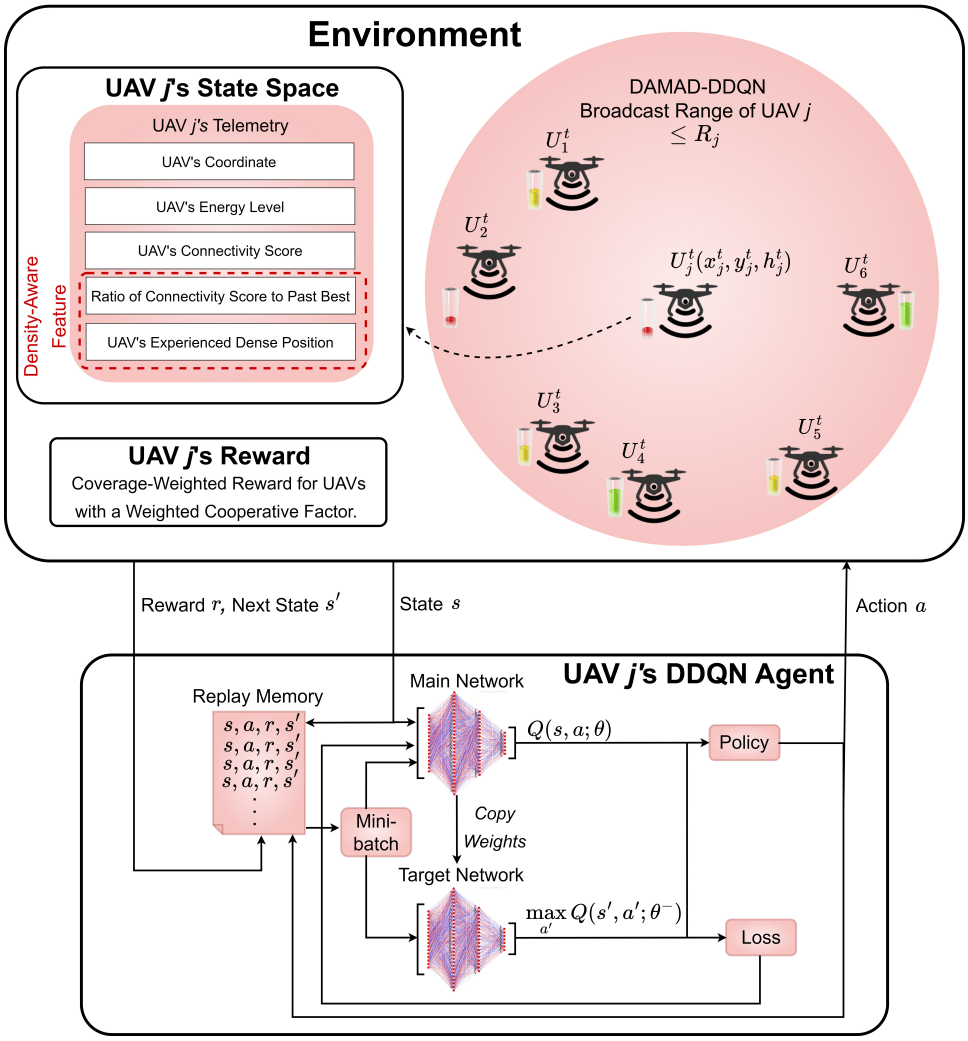


Figure 4.4: Density-aware multi-agent decentralised double deep Q-network (DAMAD-DDQN) framework where each UAV j equipped with a DDQN agent indirectly interacts with its nearest neighbours which makes up the state space. Each UAV indirectly collaborates to improve overall system performance.

Algorithm 4 shows the DAMAD–DDQN for Agent j. The DAMAD–DDQN algorithm extends the MAD–DDQN algorithm but with density awareness. Unlike the CMAD–DDQN algorithm, the DAMAD–DDQN algorithm has no direct communication mechanism, hence it has no communication overhead. The line 14–19 of Algorithm 4 shows the state space for Agent j. Unlike the MAD–DDQN algorithm, a density-aware feature is added on line 17–19. This density-aware feature provides the agent-controlled UAV j with insights into its present coverage performance with respect to its past best coverage performance, as well as keeping track of the position where it experienced its best connectivity score. The DDQN procedure described on line 28–36 optimises the agent's decisions.

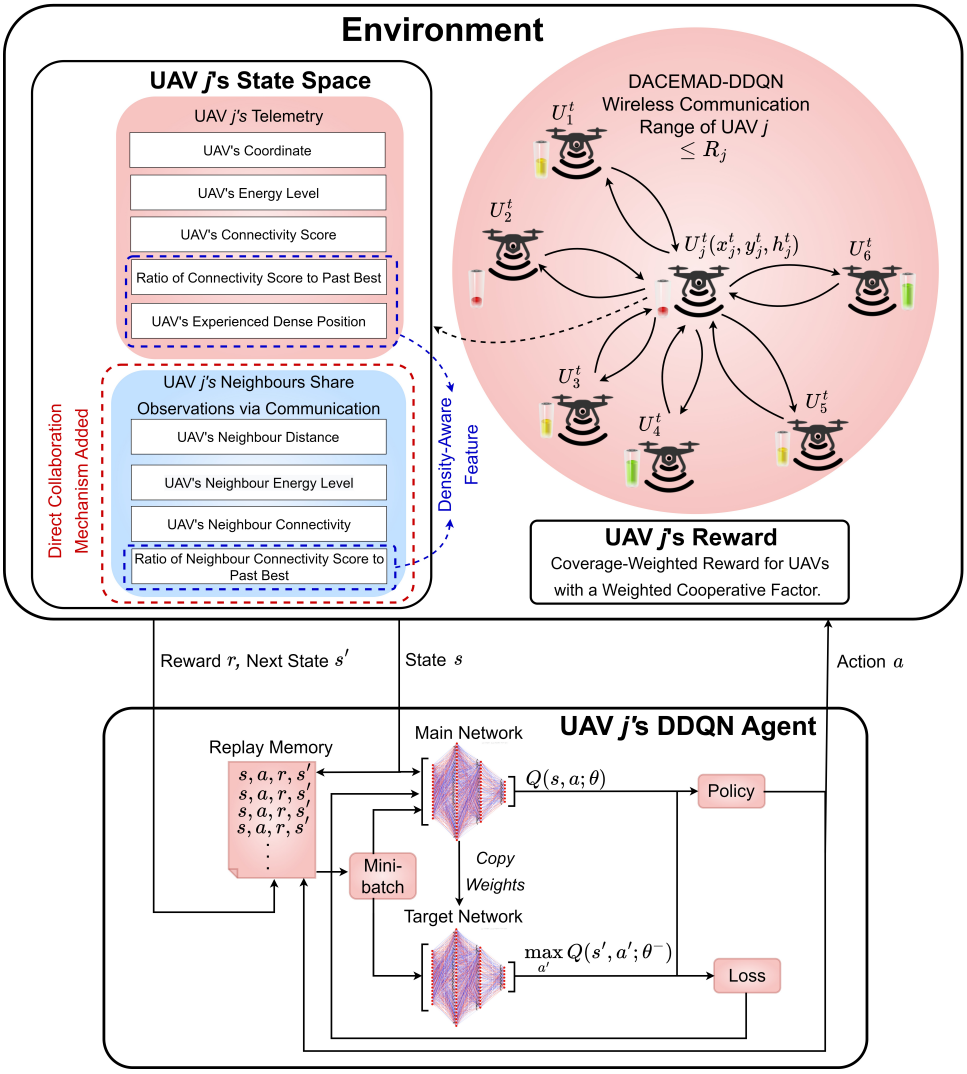


Figure 4.5: Density-aware communication-enabled multi-agent decentralised double deep Q-network (DACEMAD-DDQN) framework where each UAV j equipped with a DDQN agent interacts and share knowledge with its nearest neighbours which makes up the state space. Each UAV directly collaborates to improve overall system performance.

In the next sub-section, we are motivated to investigate the impact of information shared among neighbouring agent-controlled UAVs in intelligently serving highly mobile and densely uneven users' distribution while optimising the total system's EE in a shared, dynamic and interference-limited network environment.

4.2.3.5 Collaborative Density-Aware Agents with Neighbour Knowledge

Previously, we present the DMARL variant design that has no mechanism for sharing past coverage experience with its closest neighbours. On this note, we present a variant that allows agent-controlled UAVs to share their past coverage experiences with neighbours and

Algorithm 5 Double Deep Q-Network (DDQN) for Agent j with Density-Awareness and Direct Communication Mechanism

Input: UAV3Dposition $(x_j^t,\ y_j^t,\ h_j^t)$, c_j^t, InstantaneousEnergyConsumed e_j^t, $\frac{c_j^t}{c_j^*}$, ExperiencedDensePosition $(x_j^*,\ y_j^*)$, UAVneighbourDistances N_d^t, NeighboursConnectionScore C_z^t, $\boxed{\frac{c_o^t}{c_o^*}}$, NeighboursInstantaneousEnergyConsumed $e_z^t \in S$ and Output: Q-values corresponding to each possible action $(+x_s,\ 0)$, $(-x_s,\ 0)$, $(0, +y_s)$, $(0, -y_s)$, $(0,\ 0) \in A_j$. Given the ConnectivityScore c_j^t, PastBestConnectivityScore c_j^*, NeighbourConnectivityScore c_o^t, BestNeighbourConnectivityScore c_o^* .
for all $a \in A_j$ and $s \in S$ **do**:
    $Q_{(1)}(s,a)$, $Q_{(2)}(s,a)$, $\mathcal{D}$ – empty replay buffer, θ – initial network parameters, θ^- – copy of θ, N_r – maximum size of replay buffer, N_b – batch size, N^- – target replacement frequency.
    $s \leftarrow$ initial state
    1500 $\leftarrow$ maxStep
    while *goal* not Reached and Agent *alive* and maxStep not reached **do**
        $s \leftarrow$ MapLocalObservationToState(*Env*)
                ▷ `Execute ϵ-greedy method based on` π_j
        $a \leftarrow$ DeepQnetwork.SelectAction(s)
                ▷ `Agent executes action in state` s
        a.execute(*Env*)
        if a.execute(*Env*) is True **then**
                ▷ `Map sensed observations to new state` s'
            Env.UAVposition
            Env.ConnectivityScore using Equation (3.3)
            Env.InstantaneousEnergyConsumed using Equation (3.6)
            Env.RatioOfConnectivityScoreToPastBestConnectivityScore
            Env.ExperiencedDensePosition
                ▷ `Map communicated observations from closest neighbours based on an existing ANR mechanism for UAV communication to new state` s'
            Env.Neighbour.UAVneighbourDistances
            Env.Neighbour.ConnectivityScore
            Env.Neighbour.RatioOfNeighbourConnectivityScore
                ToPastBestNeighbourConnectivityScore
            Env.Neighbour.InstantaneousEnergyConsumed
            Density-aware feature and closest neighbours' information added to DAMAD–DDQN variant
        $r \leftarrow$ Env.RewardWithCollaborativeNeighbourFactor using Equation (4.2)
        update $(x_j^*,\ y_j^*)$, c_j^*, c_o^* $\forall\ t$
        if $c_j^t > c_j^*$ **then**
            $(x_j^*,\ y_j^*) \leftarrow (x_j^t,\ y_j^t)$
            $c_j^* \leftarrow c_j^t$
        if $c_o^t > c_o^*$ **then**
            $c_o^* \leftarrow c_o^t$
                ▷ `Execute UpdateDDQNprocedure()`
        Sample minibatch of N_b tuples $(s,a,r,s') \sim Unif(\mathcal{D})$
        Construct target values, one for each of the N_b tuples:
        Define $a^{max}(s';\theta) = \arg\max_{a'} Q_{(1)}(s',a';\theta)$
        if s' is Terminal **then**
            $y_j = r$
        else
            $y_j = r + \gamma Q_{(2)}(s', a^{max}((s';\theta);\theta^-)$
        Apply gradient descent step with loss $\| y_j - Q(s,a;\theta) \|^2$
        Replace target parameters $\theta^- \leftarrow \theta$ every N^- step
    endwhile

use this knowledge to improve the network performance in dense and uneven user distribution. The density-aware CMAD–DDQN algorithm of the DMARL is a fully decentralised MARL approach that allows agents to share observations from their closest neighbours via a communication mechanism as defined in [3GPP, 2019, 3GPP, 2008]. We recall that previous DMARL variants, MAD–DDQN and CMAD–DDQN, do not have a mechanism that enables agents to keep track of their past experiences, especially the history of dense regions in the network. Furthermore, unlike the DAMAD–DDQN algorithm, we propose the DMARL variant with *density-aware direct collaborative agents* that is suitable in disaster scenarios and has a mechanism that allows an agent directly interact with its closest neighbours.

The work [Goldman and Zilberstein, 2003] argued that a single agent cannot perform an action and receive the observation of other agents without any communication. On this note, we present a variant that allows agent-controlled UAVs to share their telemetry information (i.e., coordinates, connectivity score, best connectivity score experienced and energy level) with closest neighbours to improve the performance in the network. In particular, communication can play a crucial role in achieving synchronisation among agents in a decentralised MARL [Zhu et al., 2022]. In addition, agents may require past histories of observations to improve learning [Goldman and Zilberstein, 2003, Oliehoek and Spaan, 2012].

In this thesis, we propose a density-aware CMAD–DDQN algorithm where each agent observes its best-experienced connectivity score and the position where it experienced the best number of connected users and receives via communication the best neighbour connectivity score to keep track of dense users' area in the network. The computational complexity of the problem in Equation (3.9a) is known to be NP-complete [Liu et al., 2019a]. Notwithstanding, the complexity can be reduced when the agents share some observations [Becker et al., 2004]. However, sharing all observations will result in increased communication overhead. Unlike the DAMAD–DDQN algorithm, we address requirements R1, R2, R3 and R4 as specified in Section 4.1 via our contributions C1, C2 and C3 while providing an answer to our third research question RQ3[8]. We assume that each UAV is equipped with a Double

[8] **RQ3:** Can UAVs collaborate intelligently to improve the total system's energy efficiency in highly mobile and densely uneven users' distribution in an urban environment?

Deep Q-Network (DDQN) agent which can learn the density of users covered by itself and its closest neighbours in the network, and then adjust its trajectory in such a way that will maximise the total system's EE while jointly optimising the total number of connected users and the energy utilisation of the UAV. Nevertheless, in a typical multi-agent setting, it is often hard to achieve collaboration [Dafoe et al., 2020] since the interference-limited environment pushes agents to exhibit some selfish behaviours. Therefore, a robust and adaptive strategy is required to allow agents to collaborate while completing their tasks.

Algorithm 5 shows the DACEMAD-DDQN for Agent j. The DACEMAD-DDQN algorithm extends the CMAD-DDQN algorithm, which relies on a communication mechanism based on the existing 3GPP standard [3GPP, 2008]. Note that we assume a lossless wireless channel that allows observations sent to other agent-controlled UAVs to be received without delay or distortion. However, the DACEMAD-DDQN algorithm equips each agent with the knowledge of the number of connected users locally and keeps track of its best-experienced coverage during the training phase. From Algorithm 5, Agent j follows an ϵ–greedy policy by executing an action a (line 11), transiting from state s (line 14–24) to a new state s' and receiving a reward (line 25) given in Equation (4.2). At each time-step during the training phase, each agent keeps track of its best-experienced connectivity score and also keeps track of that position where it experienced the best number of connected users as shown on line 27–29. Furthermore, each agent keeps track of the best-experienced connectivity score locally as shown on line 30–31, which is achieved via communicating with its closest neighbours. The DDQN procedure described on line 33–41 optimises the agent's decisions. Figure 4.5 shows the DACEMAD–DDQN framework where each DDQN agent-controlled UAV j interacts with its environment. To optimise the UAVs' trajectory towards the dense areas, we design the state space, action space and reward function as follows:

4.2.3.6 DACEMAD–DDQN state space

The state space for Agent j given in line 14–24 can be expressed as a tuple, $\langle x^t : \{x_{min}, ..., x_{max}\}, y^t : \{y_{min}, ..., y_{max}\},\ h^t : \{h_{min}, ..., h_{max}\},\ C_j^t,\ e_j^t,\ \frac{C_j^t}{C_j^*},\ x^*,\ y^*,\ N_d^t, C_z^t, \frac{C_o^t}{C_o^*},\ e_z^t \rangle$, where x_{min}, y_{min}, h_{min} and x_{max}, y_{max}, h_{max} are the minimum and maximum 3D coordinates of the considered

geographical space, respectively. $\frac{C_j^t}{C_j^*}$ is the ratio of the connectivity score of UAV j at time-step t to the best connectivity score experienced by the UAV over a series of past encounters. The x^* and y^* are the coordinates where the UAV experienced its best connectivity score. N_d^t is the set of distances of the neighbouring UAVs, C_z^t is the connectivity score of neighbouring UAVs, and e_z^t is the instantaneous energy level of neighbouring UAVs. $\frac{C_o^t}{C_o^*}$ is the ratio of the connectivity score of UAV j's neighbours at time-step t to the best connectivity score experienced locally over a series of past encounters. The C_o^t is the total number of connected users by UAVs.

In the next Section, we provide the complexity analysis of our algorithm.

4.3 Complexity Analysis of the DMARL

The neural network (NN) architecture of Agent j's DDQN shown in Figures 4.2, 4.3, 4.4 and 4.5 comprises of a $\mathcal{S}$–dimensional state space input vector, densely connected to 2 layers with 128 and 64 nodes, with each using a rectified linear unit (ReLU) activation function, leading to an output layer with Q-values corresponding to the dimension of the action space. Our decentralised approach assumes agents to be independent learners while relying on collaboration with closest UAVs. We follow the analysis presented in [Hribar et al., 2022, Tan and Guan, 2022, Liu et al., 2020].

Theorem 1 *The time complexity of the decentralised Q-learning algorithm is approximately* $\mathcal{O}(N_eT)$.

Proof of Theorem 1 *The DQLSI variant listed in Algorithms 1 converges after* T *time steps and* N_e *learning episodes. The time cost of each iteration is given as* $\mathcal{O}(1 \times 1)$ *[Tan and Guan, 2022]. Given that the time complexity of the UAVs to converge at an optimal policy of jointly improving the number of connected ground users and the energy utilization over several time steps is* $\mathcal{O}(N_eT)$. *Thus, the time complexity of the DQLSI variant is approximately* $\mathcal{O}(N_eT)$.

Theorem 2 *The time complexity of the decentralised double deep Q-network algorithm is approximately* $\mathcal{O}\Big(N_eT(D_sW_1 + \sum_{k=1}^{K} W_kW_{k+1})\Big)$.

Table 4.1: Summary of DMARL Design

Design Contribution	**Decentralised Multi-Agent Reinforcement Learning**				
	Independent Learning Agent	Indirect Collaborative Agent	Direct Collaborative Agent	Density-Aware Indirect Collaborative Agent	Density-Aware Direct Collaborative Agent
Agent's Architecture	Tabular	Deep NN	Deep NN	Deep NN	Deep NN
Requirements Addressed	R1	R1, R2	R1–R3	R1, R2, R4	R1–R4
Research Question	RQ1	RQ2	RQ2	RQ3	RQ3
Contribution	C1	C1, C2	C1, C2	C1, C2, C3	C1, C2, C3
Collaborative Mechanism	Indirect	Indirect	Direct	Indirect	Direct
Communication-Enabled	Broadcast	–	Multicast	–	Multicast
State Components	x_j^t, y_j^t, z_j^t, N_d	x_j^t, y_j^t, z_j^t, e_j^t C_j^t	x_j^t, y_j^t, z_j^t, e_j^t C_j^t, N_d^t, C_z^t, e_z^t	x_j^t, y_j^t, z_j^t, e_j^t C_j^t, $\frac{C_j^t}{C_j^*}$, x^*, y^*	x_j^t, y_j^t, z_j^t, e_j^t C_j^t, $\frac{C_j^t}{C_j^*}$, x^*, y^*, N_d^t, C_z^t, $\frac{C_o^t}{C_o^*}$, e_z^t
Reward	Eqn. (4.2)	Eqn. (4.2)	Eqn. (4.2)	Eqn. (4.4)	Eqn. (4.4)
Reward Components	C_j^t, C_j^{t-1}, e_j^t, e_j^{t-1}, C_o^t, C_o^{t-1}	C_j^t, C_j^{t-1}, e_j^t, e_j^{t-1}, C_o^t, C_o^{t-1}	C_j^t, C_j^{t-1}, e_j^t, e_j^{t-1}, C_o^t, C_o^{t-1}	C_j^t, C_j^{t-1}, e_j^t, e_j^{t-1}, C_o^t, C_o^{t-1}, C_j^*, C_o^*	C_j^t, C_j^{t-1}, e_j^t, e_j^{t-1}, C_o^t, C_o^{t-1}, C_j^*, C_o^*
Complexity	Theorem 1	Theorem 2	Theorem 2	Theorem 2	Theorem 2
Communication Overhead	$E \times U_L(t)_j$	-	$3E \times U_L(t)_j$	-	$4E \times U_L(t)_j$
Communication Components	Position	-	Position, Connection, Energy Level	-	Position, Connection, Energy Level, Best Connection

$U_L(t)_j$–number of neighbours of agent-controlled UAV j at time t, E–number of bits needed to represent each observation

Proof of Theorem 2 *The DMARL variants listed in Algorithms 2, 3, 4 and 5 show that after T time steps and N_e learning episodes, the neural network parameters of DDQN algorithm converge and tend to be stable. The time complexity of neural network (NN) represents the number of operations of the network model, which is determined by the dimension of input state D_s and action D_a, the number of layers and the number of neurons in each layer of the NN [Tan and Guan, 2022]. The operation times of the DDQN in each time step can be expressed as $\mathcal{O}\Big(D_sW_1+\sum_{k=1}^{K}W_kW_{k+1}\Big)$, where K is the number of hidden layers of the NN, and W is the number nodes in each layer. Hence, the time complexity of the decentralised double deep Q-network algorithm is approximately $\mathcal{O}\Big(N_eT\big(D_sW_1+\sum_{k=1}^{K}W_kW_{k+1}\big)\Big)$. The time complexity of a closely related work and evaluation baseline [Liu et al., 2020] (MADDPG) is approximately $\mathcal{O}\Big(N_eT\big(D_sW_1+\sum_{k=1}^{K}W_kW_{k+1}\big)\Big)+\mathcal{O}\Big(N_eT\big((D_a+D_s)W_1+\sum_{k=1}^{K}W_kW_{k+1}\big)\Big)$.*

4.4 Summary

In this chapter, we presented a set of requirements for Decentralised Multi-Agent Reinforcement Learning (DMARL) to allow each UAV equipped with an autonomous agent to

intelligently serve ground users while improving the overall system's energy efficiency (EE) in a shared, dynamic and interference-limited network environment. We then mapped each of the requirements to address our specific research questions via our contributions. Table 4.1 summarises the DMARL design contribution, showing a comparison of our proposed variants. The first variant with *Independent Learning agents* is designed to address requirement R1 as specified in Section 4.1 via our contribution C1 while proffering an answer to our first research question RQ1. We provide an answer to our second research question RQ2 through our second and third variants which have a collaborative mechanism via the reward function. In particular, the second variant with *Indirect Collaborative agents* addresses requirements R1 and R2 via our contributions C1 and C2, while the third variant with *Direct Collaborative agents* which allows direct communication among UAVs addresses requirements R1, R2 and R3 via our contributions C1 and C2.

We provide an answer to our third research question RQ3 via two variants which have a density-aware mechanism added to enhance the UAVs' ability to serve densely and uneven users' distribution. Specifically, the fourth variant with *Density-Aware Indirect Collaborative agents* addresses requirements R1, R2 and R4 via our contributions C1, C2 and C3, while the fifth variant with *Density-Aware Direct Collaborative agents* which also allows direct communication among UAVs addresses requirements R1, R2, R3 and R4 via our contributions C1, C2 and C3. The design of the DMARL was provided along with the complexity of the algorithm. We present the DMARL implementation in the next chapter.

Chapter 5

Implementation of DMARL for UAV-Assisted Networks

In the previous Chapter, we presented the design of the DMARL for UAV-assisted networks to optimise the total energy efficiency (EE) of multiple UAVs deployed to provide wireless connectivity to ground users in a shared, dynamic and interference-limited network environment. In this Chapter, we present the implementation of DMARL for UAV-assisted networks. We present the libraries used to implement our proposed DMARL solution, the training phase, deployment setting of the UAVs and the ground users.

5.1 Implementation

We present the implementation of the DMARL for UAV-Assisted Networks presented in Chapter 4. The DMARL solution was decomposed into five variants. Each of these variants has unique design features that can be readily applied in disaster scenarios, where multiple UAVs can be deployed to provide wireless coverage to ground users. In this thesis, we assume that each of the UAVs is controlled by an autonomous agent that has a goal of maximising the overall system's EE while optimising the UAVs' flight trajectory, the number of connected ground users and the energy utilisation of the UAVs. In our DMARL for UAV-Assisted Networks implementation, we adopt existing reinforcement learning architectures

and libraries[1].

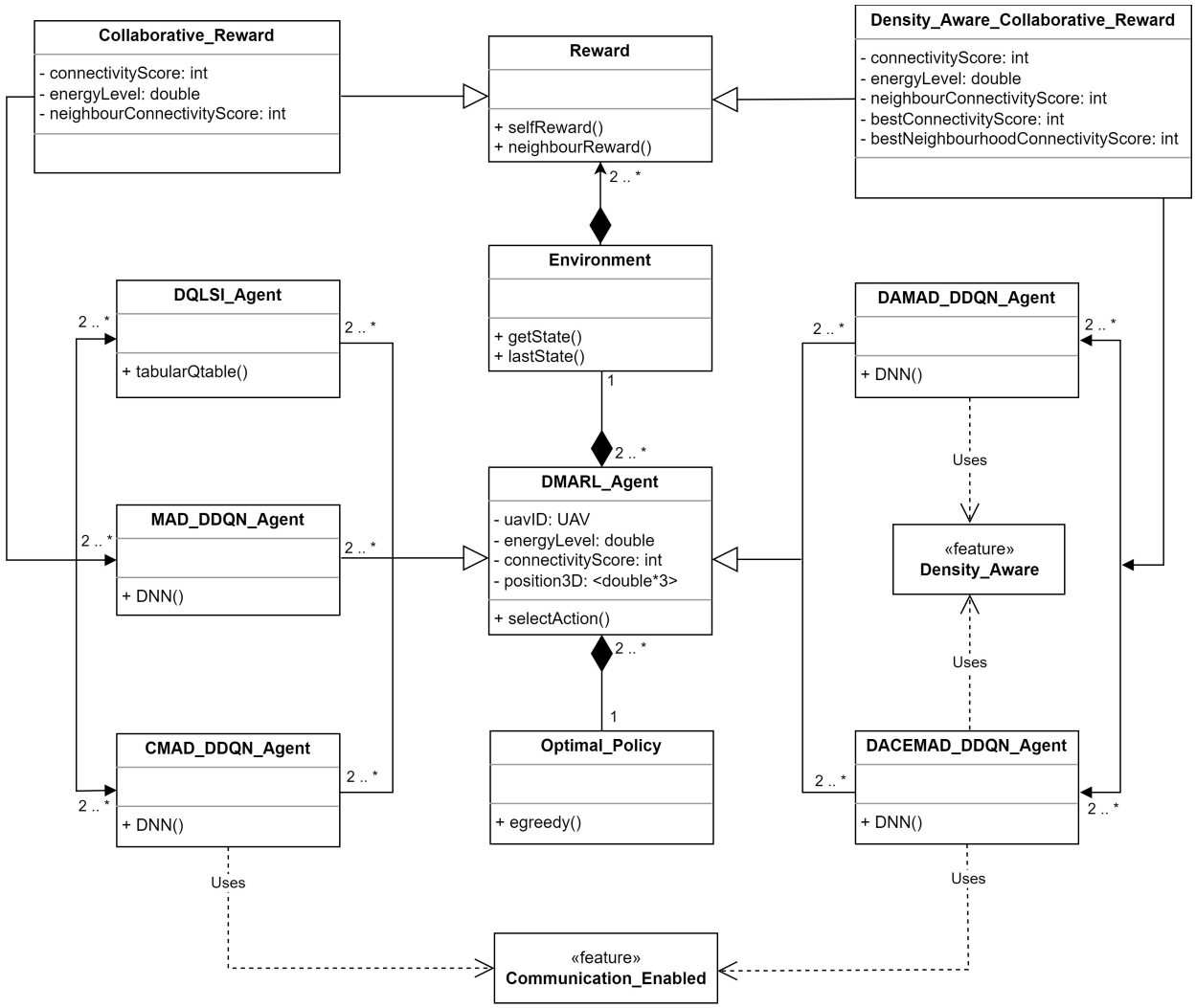


Figure 5.1: Class diagram of DMARL for UAV-assisted networks.

The class diagram of DMARL for UAV-Assisted Networks is shown in Figure 5.1. The DMARL agent `DMARL_Agent` learns an optimal policy following an ϵ–greedy policy `egreedy()`. The agent classes of each of the five variants, `DQLSI_Agent`, `MAD_DDQN_Agent`, `CMAD_DDQN_Agent`, `DAMAD_DDQN_Agent` and `DACEMAD_DDQN_Agent`, are sub-classes and all inherit the attributes of the `DMARL_Agent` class. The `DQLSI_Agent` relies on a tabular Q-Learning architecture `tabularQtable()`, while the `MAD_DDQN_Agent`, `CMAD_DDQN_Agent`, `DAMAD_DDQN_Agent` and `DACEMAD_DDQN_Agent` all rely on a Deep Neural Network architecture `DNN()`. The DMARL agent `DMARL_Agent` explores the impact of its actions on the environment `Environment` hereby transiting to a new state via `getState()` and receiving a reward `Reward`. As de-

[1]https://github.com/tunjiomoniwa/DMARL

scribed in Chapter 4, the `DQLSI_Agent`, `MAD_DDQN_Agent` and `CMAD_DDQN_Agent` receive the reward `Collaborative_Reward`, while `DAMAD_DDQN_Agent` and `DACEMAD_DDQN_Agent` receive the `Density_Aware_Collaborative_Reward`.

The `Collaborative_Reward` and `Density_Aware_Collaborative_Reward` both inherit the properties of the `Reward`, where the individual reward and neighbour reward of the agent is mapped to the overall reward. On one hand, the Figure indicates that the `DAMAD_DDQN_Agent` and `DACEMAD_DDQN_Agent` both use the `Density_Aware` feature which helps the agent-controlled UAVs keep track dense user locations during the coverage task. On the other hand, we see that the `CMAD_DDQN_Agent` and `DACEMAD_DDQN_Agent` use the `Communication_Enabled` feature to help enhance collaboration among agent-controlled UAVs.

5.2 Training Phase of DMARL

For the DQLSI variant, we train the Q-Learning algorithm by first initialising the Q-table. We then create the training algorithm that will independently update the Q-tables as the agent-controlled UAVs explore the environment over N_e episodes. While the agent-controlled UAV's battery power is not expended and the goal is not reached, the agents decide whether to pick a random action or exploit the already computed Q-values. This is achieved by using the ϵ–greedy method which helps to provide a balance between exploration and exploitation. Each agent j executes the chosen action in the environment in state s and obtains the next state s' and the reward r from performing the action of updating its trajectory. We then calculate the maximum Q-value for the actions corresponding to the next state s', and with that, we update our Q-value. However, it may be difficult to implement this variant with a Q table when the number of states in the environment becomes large.

For the MAD–DDQN, CMAD–DDQN, DAMAD–DDQN, and DACEMAD–DDQN variants, we adopt a DNN architecture for training the agents. During the training phase and given the state information as input, Agent j trains the main network to improve its decisions by yielding Q-values that match each possible action as output. The maximum Q-value obtained is a determinant of the action the agent executes. At each time step Agent j observes its

present state s and updates its trajectory by selecting an action a according to its policy. Following its action in time-step t, Agent j observes a reward r which is defined in (4.2), and transits to a new state s' [Sutton and Barto, 2018]. The information (s, a, r, s') is inputted in the replay memory as shown in Figures 4.4, 4.5. Agent j now samples the random mini-batch from the replay memory and uses the mini-batch to get y_j.

The optimisation is performed with $L(\theta)$ and θ updated accordingly. The target Q-network updates the parameters θ^- with the same parameters θ of the main network in every 100^{th} time-step. The memory size was set to 10,000 while using a mini-batch size of 1024. We perform the optimisation using a variant of the stochastic gradient descent called RMSprop to minimise the loss [François-Lavet et al., 2018, Chapter 4]. After several experimental trials to achieve values of learning rate and discount factor that improved the overall performance, the learning rate was set to 0.0001 and the discount factor of 0.95 was applied. Our Q-networks were trained by running multiple episodes, and at each training step the ϵ-greedy policy is used to have a balance between exploration and exploitation [François-Lavet et al., 2018]. In the ϵ-greedy policy, the action is randomly selected with ϵ probability, whereas the action with the largest action value is selected with a probability of $1 - \epsilon$. The initial value of ϵ was set to 1 and linearly decreased to 0.01. Table 5.1 summarises the parameters used in training the `DMARL_Agent`.

5.3 DMARL Experimental Setting

In this Section, we present the UAVs and ground users deployment settings.

5.3.1 UAVs Deployment

In this thesis, we adopt a hexagonal cellular structure, called honeycomb networks, which is popularly used in wireless communication, [Garcia Nocetti et al., 2002] as seen in Figure 5.2. Thus, we assume that each UAV can interact with up to six neighbouring UAVs. We consider the multi-rotor UAV design[2] with each UAV weighing up to 20kg. We assume that the UAV

[2]https://www.aeroexpo.online/cat/professional-drones-D.html

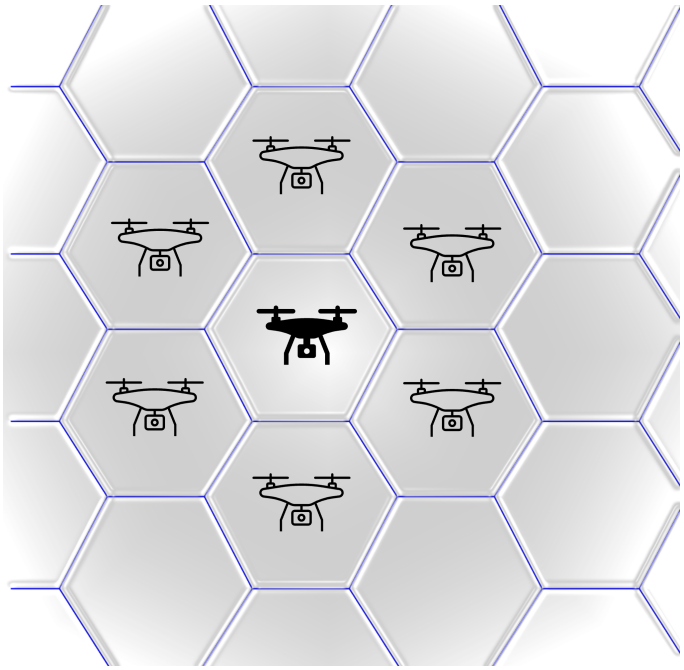


Figure 5.2: Hexagonal cellular structure with a UAV (black) having six neighbours.

can hover in space with a speed of 0 m/s and can travel at a speed of up to 20 m/s. The nominal battery capacity for the UAV is 16,000 mAh, with maximum flight endurance not exceeding 1 hour. Table 5.1 summarises the parameters used in the deployment of the agent-controlled UAVs. The parameters used for the UAV flight, such as, κ_0, κ_1, κ_2, U_{tip}, v_0 are obtained from the work [Yuan et al., 2021]. We understand that the data rate per user is a function of the Bandwidth B_w. Practically, there is no hard limit for the choice of bandwidth value been used, as this metric is determined by the type of network link being used, the amount of data that needs to be transmitted, and the capacity of the network to handle that data. Recent works in the field have applied different bandwidth values for the UAVs (e.g., 1 MHz [Liu et al., 2019a], 5 MHz [Montero et al., 2019], 20 MHz [Sun et al., 2023]). In this thesis, we adopt the bandwidth value of 1 MHz that was used in a close evaluation baseline [Liu et al., 2019a] for the sake of fair comparison. Without loss of generality, when more users are connected to a UAV, the overall throughput in the network which is shared by the ground users drops significantly [Mozaffari et al., 2019]. The authors in [Hayat et al., 2016, Refer to Table III] set a throughput requirement for network coverage provisioning of 12.2 kbps for voice and about 384 kbps for video communication. In our disaster scenario, we assume that the ground users may use the coverage service of the UAVs to send emergency

text or voice notifications across the network [Hayat et al., 2016, Section VII]. Nevertheless, we understand that if 1 MHz results in about 50 kbps for the average user, then 20 MHz under the exact same conditions would give 1 mbps, which is a workable data rate for other bandwidth-hungry applications such as video, gaming and web applications [Sun et al., 2023].

Table 5.1: Parameters Used in Implementation

Parameters	*Value*
Software platform	MATLAB (Variant 1)
”	Python (Variants 2–5)
Library	PyTorch (Variants 2–5)
Optimiser	RMSprop (Variants 2–5)
Loss function	MSELoss (Variants 2–5)
Learning rate	0.0001
Discount factor	0.95
Exploration policy	ϵ-greedy (Linear decrease)
Hidden layers	2 (128, 64) (Variants 2–5)
Activation function	ReLu (Variants 2–5)
Replay memory size	10,000 (Variants 2–5)
Batch size	1024 (Variants 2–5)
Learning episodes	100 (Variant 1)
”	250 (Variants 2–5)
Maximum step size (maxStep)	1500
Ground users models	Static, RW, RWP (Variant 1)
”	RW, RWP, GMM (Variants 1–3)
”	SUMO (Variants 4 & 5)
Coverage Area	1 km^2 (Variant 1–3)
” (DCC) [Guériau and Dusparic, 2020]	3 km^2 (Variants 4 & 5)
” (M50 motorway) [Guériau and Dusparic, 2020]	7 km (Variants 4 & 5)
” (N7 national road) [Guériau and Dusparic, 2020]	6.5 km (Variants 4 & 5)
SUMO floating car data (FCD) output	1 s (Variants 4 & 5)
Mobile ground user direction range	[0, 2π]
Mobile ground user speed range	[0, 1] m/s
UAV speed V	[0, 20] m/s
Coefficient of blade profile power κ_0 [Yuan et al., 2021]	79.85 J/s
Coefficient of induced power κ_1 [Yuan et al., 2021]	88.63 J/s
Coefficient of parasite power κ_2 [Yuan et al., 2021]	0.018 kg/m
Rotor blade's tip speed U_{tip} [Yuan et al., 2021]	120 m/s
Mean hovering velocity v_0 [Yuan et al., 2021]	4.03 m/s
Vehicle speed range (DCC) [Guériau and Dusparic, 2020]	[0, 50] km/hr
” (M50 & N7) [Guériau and Dusparic, 2020]	[0, 100] km/hr
Number of UAVs deployed	[2–12]
Weight per UAV	16 kg
Nominal battery capacity	16,000 mAh
Transmission range T_{range}	500 m
UAV Collision distance d_{col} [Wang et al., 2021]	20 m
Maximum transmit power [Liu et al., 2019a]	20 dBm
Noise power [Mozaffari et al., 2017]	-130 dBm
SINR threshold [Mozaffari et al., 2017]	5 dB
Bandwidth B_w [Liu et al., 2019a]	1 MHz
Pathloss exponent [Liu et al., 2019a]	2
UAV step distance ($\forall\ x_s, y_s, z_s$)	[0–20] m

5.3.2 Ground Users Deployment

In this thesis, we consider different ground users' deployments under different deployment scenarios. For the DQLSI variant, both static and mobile ground users were deployed. We deploy 200 static and 200 mobile ground users. We assume that the static and mobile users are pedestrians that move within a 1 km^2 area. In this variant, both the static and mobile ground users' position data are synthetically generated using `rand()`. Here, we modelled the mobility of mobile users to follow the RW and RWP mobility models.

For the MAD–DDQN and CMAD–DDQN variants, both static and mobile ground users are deployed. Again, we assume that the mobile users are pedestrians that move within a 1 km^2 area. We deploy 200 static and 200 mobile ground users. In these variants, we tried as much as possible in getting real-world data to depict real-world pedestrians. We were able to get 126 bin location data in the Drumcondra South A area of Dublin with coordinates around 53° 22' 9" N, 6° 14' 45" W [Dublin, 2021] along with 74 synthetic data generated using `random.sample(range, size)`, to make up 200 mobile ground users. We then modelled the mobility of mobile users to follow the RW, RWP, and GMM mobility models which provide a realistic depiction of pedestrians as discussed in Chapter 3.

Table 5.2: Deployment of Ground Users in SUMO

Road Network	Free flow	Saturated	Congested
DCC (Vehicles)	3179	27167	27702
DCC (Pedestrians)	1342	14756	15471
M50 Motorway	1348	23508	25316
N7 National Road	1236	12191	12769

For the DAMAD–DDQN and DACEMAD–DDQN variants, mobile users are deployed in the road networks. We assume that the mobile users are pedestrians and vehicles that move within some urban road networks. The speed range for pedestrians fall within [0, 1] m/s, while that of vehicles falls within [0, 50] km/hr [Guériau and Dusparic, 2020]. The considered road networks across Ireland are:

1. 3000×3000 m^2 area of Dublin city centre (DCC).
2. 7 km segment of the M50 motorway[3] in Ireland.

[3]M50 motorway was built to form the urban boundary of Dublin

3. 6.5 km segment of the N7 national road in Ireland.

In each of these deployment scenarios, we consider different traffic conditions:

(a) Free flow traffic condition, where there is a low number of vehicles or pedestrians. This is usually early in the morning when road traffic is quite low.

(b) Saturated traffic conditions, where the number of vehicles increases and traffic congestion begins to build.

(c) Congested traffic condition, where there is a high number of vehicles on the road and often occurs during peak hours of the day.

Table 5.2 shows the number of ground users deployed on different road networks and under different traffic conditions in SUMO. The road networks and data used are based on real-world data[4] samples from the Dublin city council[5] [Guériau and Dusparic, 2020]. We adopt the intelligent driver model (IDM) to capture traffic phenomena and road user behaviour. Furthermore, we modify the Dublin City Centre (DCC) network in [Guériau and Dusparic, 2020] to accommodate for pedestrians[6]. For the motorway and national road networks, traffic demand was generated from loop sensors data from the open dataset made available by Transport Infrastructure Ireland[7] that covers the Irish motorways and national roads. We consider a scenario where users (i.e., vehicles and/or pedestrians) in the DCC, the M50 motorway and N7 national road networks are fully mobile. We consider a realistic scenario where both vehicles and pedestrians are not confined to a geographical space but may enter or leave the network over time. After deployment, simulation was carried out and the floating car data (FCD)[8] output was aggregated, which contains the GPS data of the vehicles and pedestrians every second. The GPS data was cleaned. The data was then integrated into the Python simulation environment after which the UAVs are deployed to provide coverage.

[4]https://github.com/maxime-gueriau/ITSC2020_CAV_impact
[5]https://data.gov.ie/dataset/traffic-volumes
[6]https://sumo.dlr.de/docs/Simulation/Pedestrians.html
[7]https://data.gov.ie/dataset/traffic-counter-data
[8]The FCD export comprises location and speed along with other information for every vehicle and person in the network at every time step.

Table 5.3: Summary of DMARL Implementation

Implementation	**Decentralised Multi-Agent Reinforcement Learning**				
	Independent Learning Agent	Indirect Collaborative Agent	Direct Collaborative Agent	Density-Aware Indirect Collaborative Agent	Density-Aware Direct Collaborative Agent
Agent	`DQLSI_Agent`	`MAD_DDQN_Agent`	`CMAD_DDQN_Agent`	`DAMAD_DDQN_Agent`	`DACEMAD_DDQN_Agent`
Agent's Architecture	Tabular	Deep NN	Deep NN	Deep NN	Deep NN
Software Platform	MATLAB	Python	Python	Python	Python
Ground Users	Pedestrians	Pedestrians	Pedestrians	Pedestrians, Vehicles	Pedestrians, Vehicles
Mobility Model	RW, RWP	RW, RWP, GMM	RW, RWP, GMM	IDM (SUMO)	IDM (SUMO)
User Mobility Flow	Constrained	Constrained	Constrained	Not Constrained (in/out of region)	Not Constrained (in/out of region)
Data Source	Synthetic	Synthetic + Real-world	Synthetic + Real-world	Real-world	Real-world

5.4 Summary

In this chapter, we presented the implementation of DMARL for UAV-assisted networks. The libraries used to implement our proposed DMARL solution were also presented. The class diagram for the DMARL agent was presented. We then described the training phase of our DMARL agent. Finally, we presented the deployment setting of the UAVs and the ground users. Table 5.3 shows the summary of the implementation of DMARL for UAV-assisted networks. In this thesis, we consider both tabular and DNN architecture. Both pedestrians and vehicles are deployed to the environment as ground users. Various mobility models were used to depict human mobility patterns, with energy-constrained UAVs deployed to provide wireless connectivity to these ground users. We source data synthetically and also from real-world scenarios using SUMO. In the next chapter, we evaluate the DMARL approach to maximise the total system's energy efficiency (EE) by jointly optimising its 3D trajectory, the number of connected users, and the energy consumed. The objective of the evaluation will be to answer our research questions outlined in Chapter 1.

Chapter 6

Evaluation

In this section we present an evaluation of the DMARL approach to optimise the total system's energy efficiency (EE). We present the objectives of the evaluation, along with the metrics that we used to measure the performance of the proposed DMARL. We describe the baselines used for comparative analysis. The experimental settings and scenarios are presented. We then describe the experiments we used for the evaluation, and present and analyse their outcomes.

6.1 Evaluation Objectives

The purpose of the evaluation of DMARL for UAV-assisted networks is to answer the research questions from Chapter 1. The main objective of the DMARL design is to provide a decentralised multi-agent technique that allows each UAV equipped with an autonomous agent to intelligently serve ground users while improving the overall system's EE in a shared, dynamic and interference-limited network environment. To ensure that our DMARL design fully addresses the research questions in Chapter 1, we aim to investigate the performance of our proposed DMARL under various evaluation scenarios. The DMARL can be said to have addressed the research questions by meeting the design requirements for multi-UAVs deployment in a shared, dynamic and interference-limited network environment if it satisfies the performance requirements. We investigate the effectiveness of our proposed DMARL in

addressing the overarching research question[1]. The objective is to observe whether results hold for:

(a) Different ground user types (pedestrians, vehicles).

(b) Different ground users' deployment settings and distribution (static/mobile, even/uneven).

(c) Different UAVs configuration (varying number of deployed agent-controlled UAVs).

(d) Different mobility models (mathematical-based, SUMO-generated).

(e) Different densities and speeds of users based on different road networks.

Next, we present the metrics used to evaluate the performance of the DMARL for UAV-assisted networks and provide justification for their use.

6.2 Evaluation Metrics

We considered the following metrics that contribute to answering our overarching research question for performance evaluation:

1. Cumulative reward: This is the total amount of reward an agent accumulates over several time steps. This metric is extensively used in RL literature to demonstrate the performance of a learning agent. An RL agent is said to be learning if the value of this metric increases over a series of time steps, hence, we adopt this metric in our evaluation.

2. Total energy consumed: This metric shows the amount of energy depleted through propulsion by UAVs during flight given as Equation (3.6). The unit used for measurement is kiloJoules (kJ). The system performance is worse if the total energy consumed by the UAVs is high. On the other hand, a system with low energy consumption is desirable. This metric is very important as it directly impacts battery life and a UAV that completely depletes its battery dies out and can not provide coverage any longer. This metric also affects the total system's EE as seen in Equation (3.8). Intuitively, high

[1] **RQ:** Can UAVs deployed to provide wireless connectivity to mobile ground users minimise the total energy consumed in a shared, dynamic and interference-limited network environment?

energy consumption may often result in a low EE. Hence, it is important to consider this metric when evaluating the performance of our DMARL approach.

3. Number of connected users: This metric shows the total number of ground users connected to UAV small cells. We express this as a percentage. It gives an estimate of how well the coverage performance of the UAVs is. A low value in the number of connected users indicates poor coverage by the UAVs, hence it is desirable to have a high number of connected users. Therefore, we consider this metric in our evaluation.

4. Total energy efficiency η: This is defined as the ratio of the total throughput and the total energy consumed given as Equation (3.7). This metric gives us an insight into how much energy is expended by the UAVs to deliver certain bits of information. It is desirable to improve the total system's EE since we want as much information as possible to be delivered using a minimum amount of energy. We aim to optimise this metric to allow UAVs effectively serve ground users for an extended duration.

5. Fairness index: This metric reflects the QoS level of ground users served by UAVs from the initial time-step to the current time-step given as Equation (3.4). All ground users need to be fairly served, being connected to a UAV small cell as much as possible. For example, when most ground users are not covered or served most of the time, it leads to geographical unfairness and a poor fairness index. Fairness should be high. Hence, we adopt this metric for evaluation.

6. Area covered: This metric shows the ground area covered by the UAVs. The unit is km^2. Although UAVS need to cover as much ground area as possible, we are also particular about UAVs covering ground areas with users in them. Hence, we use this metric.

7. Connected users to deployed users ratio (CDR): This metric is useful in observing the present coverage performance with respect to the presently deployed users on the ground. It provides insight, especially in networks where there are variations in the number of deployed users over time. In particular, CDR can be useful in realistic urban

scenarios that experience the inflow and outflow of users in the coverage space. The higher the CDR, the better the coverage of ground users. On this note, during some of our evaluations, we adopt the CDR for evaluation.

8. Users (Vehicles, Pedestrians): The metric gives us a count of the number of vehicles or/and pedestrians in the network. This is useful when plotting a relationship graph between the deployed users and the covered users. The performance is desirable when the covered users closely match the deployed users. We use this metric at some point in our evaluation.

6.2.1 Baselines

In this thesis, we compare the effectiveness of the proposed DMARL against the following baselines [2]:

1. The random policy, where UAVs choose their flight directions and travel distances randomly at each time-step t.

2. Exhaustive (brute-force) search (ES) approach where the UAVs explore the entire coverage space in search of improving the overall coverage performance in the network.

3. Iterative Search (IS) [Mozaffari et al., 2017] approach where the decision-making is centralised and the locations of the ground users and UAVs are known to a control centre located at a central cloud server. This iterative algorithm was used to optimise the 3D flight trajectory of UAVs serving static ground users such that the energy consumed under their SINR constraints is minimised.

4. Clustering-based Q-Learning (CQL) [Liu et al., 2019a] approach that assumes the partitioning of the coverage area into K-clusters, and pre-assigns the UAV small cells to each centroid. The tabular Q-learning approach was used to obtain the dynamic movement of UAVs to maintain maximum mean opinion score (MOS)[3] at each time-step t. The work also neglects the impact of interference from nearby UAV cells.

[2]The baselines considered in this thesis are those that are closest to our work.
[3]It is adopted for evaluating the satisfaction of users.

5. Multi-Agent Deep Deterministic Policy Gradient (MADDPG) [Liu et al., 2020] approach that pre-partitions the network into K-cells based on prior knowledge of the static ground users' locations and neglects the impact of interference from nearby UAV cells. The work adopted the CTDE approach where information is centrally shared during the training and execution is decentralised.

Next, we introduce our evaluation scenarios and present the rationale for their use and suitability in answering our research questions.

6.3 Evaluation Scenario

In this section, we describe the scenarios we used in the evaluation of the DMARL for UAV-assisted networks. We evaluate the DMARL solution under three categories.

- **DMARL Variant with Independent Learning Agents:** Here, we consider a set of agent-controlled UAVs called *independent learners*, deployed to provide wireless coverage to ground users in a 1 km^2 geographical area. The objective here is to minimise the energy consumption of multiple UAVs while serving ground users by optimising the flight trajectory of the UAVs without the aid of a CC. Specifically, the aim is to provide an answer to the research question RQ1[4] in Section 1.3 through the contribution C1 in Section 1.4 of Chapter 1. Hence, we propose the DMARL variant called the Decentralised Q-learning with Local Sensory Information (DQLSI) algorithm to address this research question. To answer this research question, the effectiveness of the DQLSI algorithm is investigated under three scenarios:

  (a) Static setting, where we have fully decentralised agent-controlled UAVs deployed to serve randomly and evenly deployed ground users that are static within the coverage area. Several works make this assumption, hence we evaluate the performance of our proposed DQLSI algorithm using this scenario.

  (b) Dynamic setting with even randomly-distributed ground devices, where we have

[4]**RQ1:** Can UAVs serving mobile ground users improve the total system's energy efficiency in a shared, dynamic and interference-limited network environment without relying on a central controller for decision-making?

decentralised agent-controlled UAVs deployed to serve randomly and evenly distributed static and mobile ground users. This scenario investigates whether our proposed DQLSI algorithm without a CC is robust to serve when some ground users are mobile.

(c) Dynamic setting with uneven randomly-distributed ground devices, where we have decentralised agent-controlled UAVs deployed to serve both static and mobile ground users randomly and unevenly distributed in the coverage area. This scenario examines whether our proposed DQLSI algorithm without a CC is robust to serve when some ground users are mobile and unevenly distributed.

We evaluate whether the DQLSI algorithm outperforms the ES, IS [Mozaffari et al., 2017] and CQL [Liu et al., 2019a] baselines. In Section 6.4, we present the evaluation of the DMARL variant with Independent Learning Agents.

- **DMARL Variants with Collaborative Agents:** Here, we consider collaborative agent-controlled UAVs which are deployed to provide wireless coverage to pedestrians in a 1 km^2 selected area in Dublin, Ireland. The objective here is to maximise the total system's EE by jointly optimising the 3D flight trajectory, the number of connected ground users, and the total energy utilisation of multiple UAVs under a strict energy budget. The aim will be to provide an answer to the research question RQ2[5] in Section 1.3 through the contribution C2 in Section 1.4 of Chapter 1. Specifically, we propose the two collaborative DMARL variants to answer this research question. The DMARL variant with *indirect collaborative agents* is called Multi-Agent Decentralised Double Deep Q-Network (MAD–DDQN), while the variant with *direct collaborative agents* is called the Communication-enabled MAD–DDQN (CMAD–DDQN) algorithm. To effectively answer the research question, these two DMARL variants are evaluated under three scenarios:

  (a) Dynamic setting with collaborative agents with individual knowledge, where a

[5] **RQ2:** Can collaboration with closest neighbours improve the total system's energy efficiency while minimising the total energy consumed by UAVs in a shared, dynamic and interference-limited network environment?

set of *indirect collaborative agent-controlled* UAVs are deployed to serve randomly distributed static and mobile ground users. This scenario investigates whether our proposed MAD–DDQN algorithm can allow UAVs to collaborate with closest neighbours to jointly optimise the total system's EE and total energy consumed.

(b) Dynamic setting with collaborative agents with neighbour knowledge, where a set of *direct collaborative agent-controlled* UAVs are deployed to serve randomly distributed static and mobile ground users. This scenario investigates whether our proposed CMAD–DDQN algorithm can allow UAVs directly collaborate with closest neighbours to jointly optimise the total system's EE and total energy consumed.

Based on the settings above, we investigate the performance of the MAD–DDQN and CMAD–DDQN algorithms under the following conditions:

(i) Varying number of UAVs deployed over baselines.

(ii) Varying mobility models over baselines.

(iii) Varying number of UAVs deployed over mobility models.

We evaluate whether the MAD–DDQN and the CMAD–DDQN algorithms outperform the random policy and the closest evaluation baseline, MADDPG [Liu et al., 2020]. The justification for using the MADDPG is its recent application in similar environments, i.e., the intersection of vehicular networks and UAV-assisted networks [Peng and Shen, 2020]. In Section 6.5, we present the evaluation of the DMARL variant with collaborative agents.

- **DMARL Variants with Collaborative Density-Aware Agents:** Here, we consider collaborative density-aware agent-controlled UAVs which are deployed to provide wireless coverage in selected urban roads in Dublin, Ireland. We consider some selected locations in the Dublin city centre area, the M50 motorway, and the N7 national road in Ireland. The objective here is to improve the total system's EE by jointly optimising the flight trajectory, the number of connected ground users, and the total energy

utilisation of multiple UAVs serving highly mobile, densely uneven users' distribution in urban areas. The aim will be to provide an answer to the research question RQ3[6] in Section 1.3 through the contribution C3 in Section 1.4 of Chapter 1. Specifically, we propose two collaborative density-aware DMARL variants to answer this research question. The DMARL variant with *indirect collaborative density-aware agents* is called Density-Aware MAD–DDQN (DAMAD–DDQN), while the variant with *direct collaborative density-aware agents* is called the Density-Aware CMAD–DDQN (DACEMAD–DDQN) algorithm. To effectively answer the research question, these two DMARL variants are evaluated under three traffic scenarios:

(a) Urban road setting, where we have UAVs deployed to serve both vehicles and pedestrians.

(b) Motorway setting with UAVs deployed to serve the vehicles.

(c) National road setting with UAVs deployed to serve the vehicles.

We further evaluate each of these scenarios under three traffic conditions to investigate whether the DAMAD–DDQN and DACEMAD–DDQN algorithms are robust to varying densities of road users. The considered traffic conditions are:

(i) Free flow traffic condition, usually early in the morning when there is a low concentration of road users in the environment.

(ii) Saturated traffic condition, where the number of road users is increasing and moderate traffic is experienced in the environment.

(iii) Congested traffic condition, where there is a high concentration of road users on the road, and often occurs during peak hours of the day.

As highlighted in Section 5.3 of Chapter 5, the road networks and data used are based on real-world data samples from the Dublin city council[7] [Guériau and Dusparic, 2020]. However, we modify the Dublin City Centre (DCC) network in [Guériau and Dusparic,

[6] **RQ3:** Can UAVs collaborate intelligently to improve the total system's energy efficiency in highly mobile and densely uneven users' distribution in an urban environment?

[7] https://github.com/maxime-gueriau/ITSC2020_CAV_impact

2020] to accommodate pedestrians. Note that we consider realistic urban environments where users (vehicles, pedestrians) are not confined to the considered coverage area, i.e., there is a continuous flow of users in and out of the given coverage region. Hence, considering the large number of users that enter and leave the network, we consider the Covered to deployed users ratio (CDR) as a metric to provide us with insight into the coverage performance in the network. We evaluate whether the DAMAD–DDQN and DACEMAD–DDQN algorithms outperform the closest evaluation baseline, MADDPG [Liu et al., 2020]. In Section 6.6, we present the evaluation of the DMARL variant with collaborative density-aware agents.

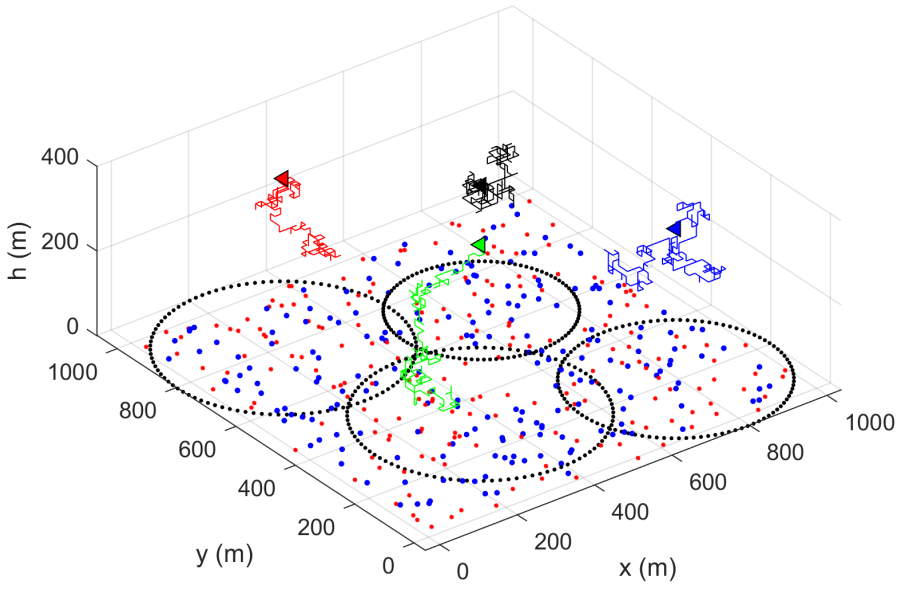


Figure 6.1: Simulation snapshot of four UAVs providing wireless coverage to 200 static (blue dots) and 200 mobile (red dots) evenly-distributed ground users. The dotted black circles represent the coverage cells of each UAV which vary according to the aerial position of the UAVs. The squiggle lines show the trajectory path of each UAV over a series of time steps. The entire coverage area is 1 km^2.

6.4 Evaluation of Independent Learning Agents

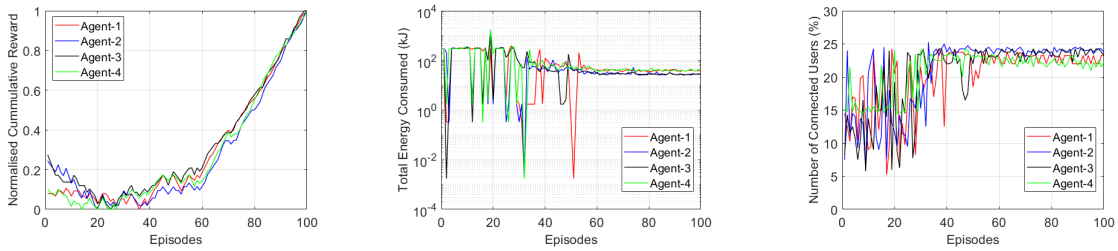
We evaluate the performance of DMARL using the variant of *independent learning agents* with no CC in 3 different scenarios. In our experiments as depicted in Figure 6.1, we consider an environment settings where static ground users and dynamic ground users are deployed

in a 1 km^2 area with four (4) UAV small cells deployed to provide wireless service. Note that the parameter values chosen are motivated by the baselines and several rounds of experimentation. The initial starting point of the UAVs is assumed to be pre-determined at the start of an episode. The mobility step size for each UAV is 20 meters. At each time step, we consider the deployment of 400 randomly distributed ground users in the coverage area, whether they be fully static or a combination of both static and mobile users as seen in Figure 6.1. To depict the mobility of ground users, we adopt the random walk mobility (RW) and random waypoint mobility (RWP) models in our experiments [Camp et al., 2002]. The mobile ground users assume a new location at every time step and this new location is calculated based on the respective mobility model used. Each UAV optimises its trajectory in such a way as to jointly maximise the number of connected ground users and the energy utilisation of the UAVs in the network. We assume a maximum connection limit of 150 active ground users per UAV [Mozaffari et al., 2017], i.e., a UAV may not have the capacity to serve more than 150 users at a time. The motivation for limiting the number of active connection is to ensure that the UAVs are not overloaded. We also understand that the capacity to serve these users may be subject to the total available bandwidth.

6.4.1 Static Setting

In this section, we consider a static setting where static ground users (i.e., stationary users) are deployed in a given coverage area. It is expected that UAV small cells are deployed to provide wireless connectivity in this static setting. A majority of works in this area of research focus on UAVs serving static users. We evaluate the effectiveness of this DMARL variant with independent learning agents using the proposed DQLSI algorithm. Figure 6.2 shows the performance of our DQLSI algorithm measured using the reward, total energy consumed, number of connected users, EE, fairness index and area covered metrics in a static setting. As seen in Figure 6.2a, all four agents try to maximise their cumulative reward through the learning episodes. After the 60^{th} episode, we observe significant convergence in the energy consumed by the UAVs, within the range of about 26 kJ – 52 kJ as seen in Figure 6.2b. Figure 6.2c shows a balance in the connection load across the four UAVs, ranging

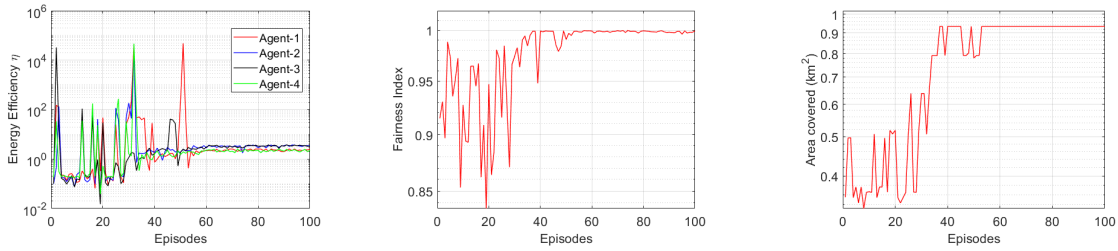
between about 20%–25% connected ground users per UAV. The total number of connected ground users for all UAVs ranges between 91%–95%. Figure 6.2d, Figure 6.2e and Figure 6.2f show the convergence of the EE, fairness index and area covered by the UAVs after the 60^{th} episode, respectively. Next, we investigate the overall system performance, i.e., all four agent-controlled UAVs and compare the proposed DQLSI algorithm with the ES, IS and CQL baselines.



(a) Cumulative reward per agent-controlled UAV vs. episodes.

(b) Total energy consumed per agent-controlled UAV vs. episodes.

(c) Number of connected users per agent-controlled UAV vs. episodes

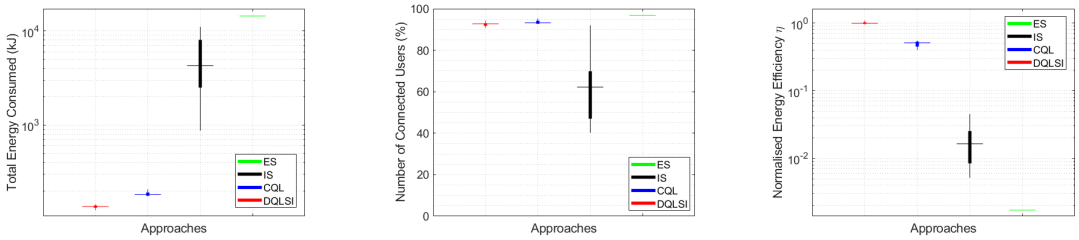


(d) Energy efficiency per agent-controlled UAV vs. episodes.

(e) Fairness Index vs. episodes.

(f) Area covered vs. episodes.

Figure 6.2: Four agent-controlled UAVs serving 400 randomly distributed static ground users.



(a) Total energy consumed vs. approaches.

(b) Number of connected users vs. approaches.

(c) Normalised Energy efficiency vs. approaches.

Figure 6.3: Comparing the proposed DQLSI with centralised baselines while deploying four agent-controlled UAVs to serve 400 randomly distributed static ground users. The plots are based on the overall performance of all four agent-controlled UAVs. 5 trained samples each were gathered from 20 independent runs.

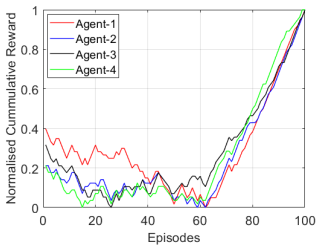
We evaluate the effectiveness of the DQLSI in addressing the research question RQ1 by

comparing it with centralised baselines that rely on a CC for decision-making. Figure 6.3 shows the comparative plots measured using the total energy consumed, number of connected users and EE metrics. From Figure 6.3b the ES method achieves the highest number of connected users at about 96%. However, this method consumed the most amount of energy in the order of hundreds of kJ as seen in Figure 6.3a. This high energy cost is due to the exploration of all possible combinations of the search space by the UAVs. Furthermore, the ES's poor energy performance is reflected in its poor EE as seen in Figure 6.3c. The proposed DQLSI approach achieved about 92% number of connected users. Interestingly, the CQL outperformed the proposed DQLSI approach in this static scenario. This is because the centroid in the CQL approach is almost always static since the ground users do not change their position over time. Nevertheless, the proposed DQLSI approach was able to reduce the total energy consumed while improving the total EE of the UAVs than the centralised baselines. This is indicative that our decentralised approach may be a preferred option suitable in energy-constrained applications. The IS performed better than the ES in terms of total energy consumed and EE, however, it achieved the least coverage among all approaches. We observe that both DQLSI and CQL which are learning-based approaches performed well in optimising the total system's EE while jointly maximising the total energy utilisation and number of connected static users in the network. However, the DQLSI stands out without relying on a CC. Next, we investigate the effectiveness of our proposed approach when mobile ground users are present in the network.

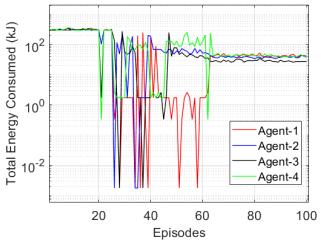
6.4.2 Dynamic Setting with Even Randomly-Distributed Ground Users

We evaluate the effectiveness of this DMARL variant with independent learning agents using the proposed DQLSI algorithm. Here, we consider a dynamic setting with even randomly-distributed ground users, where we have a combination of 200 static users and 200 mobile users that follow the random walk (RW) mobility model. Figure 6.4 shows the performance of our DQLSI algorithm measured using the reward, total energy consumed, number of connected users, EE, fairness index and area covered metrics. As seen in Figure 6.4a, all four agents try to maximise their cumulative reward through the learning episodes. After the 65^{th} episode,

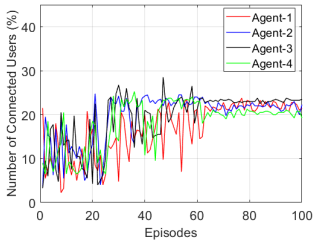
we observe significant convergence in the energy consumed by the UAVs, within the range of about 27 kJ – 53 kJ as seen in Figure 6.4b. Figure 6.4c shows a balance in the connection load across the four UAVs, ranging between about 20%–25% connected ground users per UAV. The total number of connected ground users for all UAVs ranges between 86%–91%. Figure 6.4d, Figure 6.4e and Figure 6.4f show the convergence of the EE, fairness index and area covered by the UAVs after the 65^{th} episode, respectively.



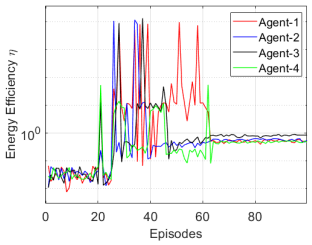
(a) Cumulative reward per agent-controlled UAV vs. episodes.



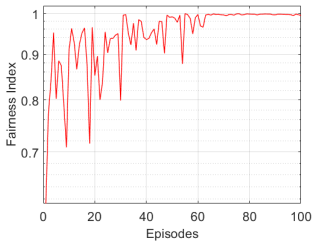
(b) Total energy consumed per agent-controlled UAV vs. episodes.



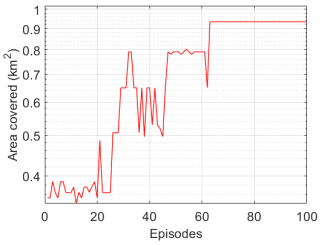
(c) Number of connected users per agent-controlled UAV vs. episodes



(d) Energy efficiency per agent-controlled UAV vs. episodes.



(e) Fairness Index vs. episodes.

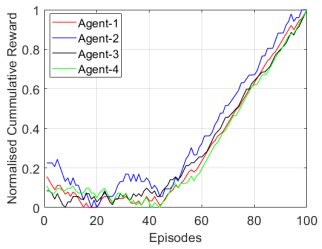


(f) Area covered vs. episodes.

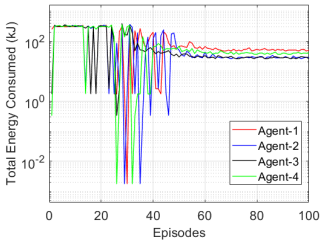
Figure 6.4: Four agent-controlled UAVs serving 400 even randomly distributed ground users (200 static and 200 mobile following the RW mobility model).

Next, we consider a dynamic setting with even randomly-distributed ground users, where we have a combination of 200 static users and 200 mobile users that follow the random waypoint (RWP) mobility model. Figure 6.5 shows the performance of our DQLSI algorithm measured using the reward, total energy consumed, number of connected users, EE, fairness index and area covered metrics. As seen in Figure 6.5a, all four agents try to maximise their cumulative reward through the learning episodes. After the 65^{th} episode, we observe significant convergence in the energy consumed by the UAVs, within the range of about 26 kJ – 46 kJ as seen in Figure 6.5b. Figure 6.5c shows a balance in the connection load across the four UAVs, ranging between about 20%–25% connected ground users per UAV. The

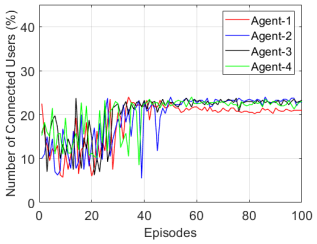
total number of connected ground users for all UAVs ranges between 84%–88%. Figure 6.5d, Figure 6.5e and Figure 6.5f show the convergence of the EE, fairness index and area covered by the UAVs after the 65^{th} episode, respectively. Next, we compare the proposed DQLSI algorithm with the ES, IS and CQL baselines under this setting.



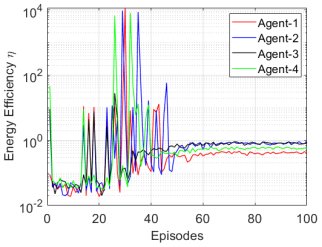
(a) Cumulative reward per agent-controlled UAV vs. episodes.



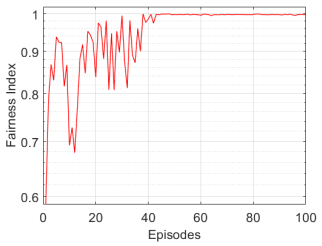
(b) Total energy consumed per agent-controlled UAV vs. episodes.



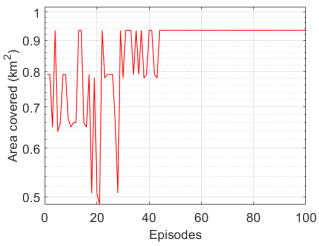
(c) Number of connected users per agent-controlled UAV vs. episodes



(d) Energy efficiency per agent-controlled UAV vs. episodes.



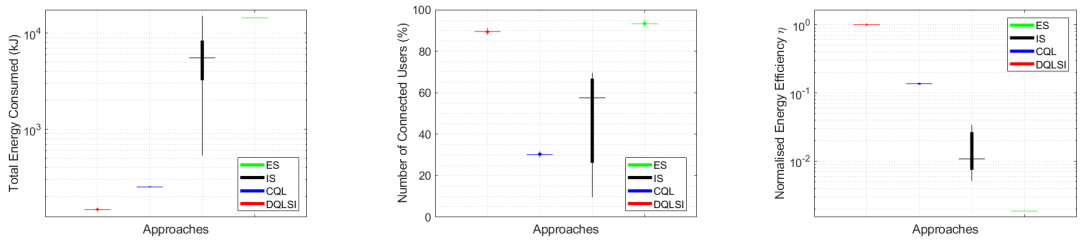
(e) Fairness Index vs. episodes.



(f) Area covered vs. episodes.

Figure 6.5: Four agent-controlled UAVs serving 400 even randomly distributed ground users (200 static and 200 mobile following the RWP mobility model).

Again, we evaluate the effectiveness of the DQLSI in addressing the research question RQ1 by comparing it with centralised baselines that rely on a CC for decision-making. This time we do not consider the deployment of only static users, but a combination of static and mobile users following the RWP model. Figure 6.6 shows the comparative plots measured using the total energy consumed, number of connected users and EE metrics. From Figure 6.6b the ES method achieves the highest number of connected users of about 96% which comes at an energy cost in the order of hundreds of KiloJoules as seen in Figure 6.6a. Intuitively, the high energy cost is due to the exploration of all possible combinations of the search space by the UAVs. Furthermore, the ES's poor energy performance is reflected in its poor EE as seen in Figure 6.6c. Interestingly, the proposed DQLSI approach which achieved about 89% in the average number of connected users outperforms the CQL approach which achieved



(a) Total energy consumed vs. approaches.

(b) Number of connected users vs. approaches.

(c) Normalised Energy efficiency vs. approaches.

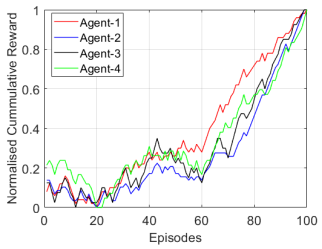
Figure 6.6: Comparing the proposed DQLSI with centralised baselines while deploying four agent-controlled UAVs to serve 200 static and 200 mobile randomly distributed even ground users (RWP model). The plots are based on the overall performance of all four agent-controlled UAVs. 5 trained samples each were gathered from 20 independent runs.

about 33%. This poor performance in the CQL may be due to the inability of the UAVs to effectively locate the quickly-changing centroids in real time. On the other hand, our proposed DQLSI does not rely on the CC to provide periodic update to the UAVs for local decision making. This allows the agent-controlled UAVs to independently learn behavioural patterns of the mobile users and provide coverage intelligently. Thus, we observe that the DQLSI is robust to the mobility of ground users. The IS performed better than the ES in terms of total energy consumed and EE, however, it achieved the least coverage among all approaches. We observe that both DQLSI and CQL which are learning-based approaches performed well in optimising the total system's EE while jointly maximising the total energy utilisation and number of connected evenly-distributed dynamic users in the network. Notwithstanding, the proposed DQLSI significantly improves the connectivity in the network and does not rely on a CC for decision-making. Next, we investigate the effectiveness of our approach when agent-controlled UAVs are deployed to serve unevenly distributed mobile ground users in the network.

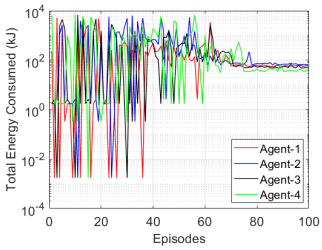
6.4.3 Dynamic Setting with Uneven Randomly-Distributed Ground Users

We evaluate the effectiveness of this DMARL variant with independent learning agents using the proposed DQLSI algorithm. Here, we consider a dynamic setting with uneven randomly distributed ground users, where we have a combination of 200 static users and 200 mobile users that follow the random walk (RW) mobility model. Figure 6.7 shows the performance of our

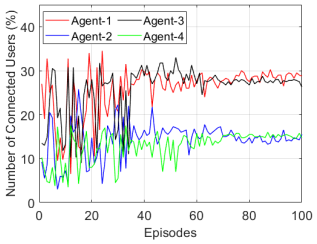
DQLSI algorithm measured using the reward, total energy consumed, number of connected users, EE, fairness index and area covered metrics. As seen in Figure 6.7a, all four agents try to maximise their cumulative reward through the learning episodes. After the 80^{th} episode, we observe significant convergence in the energy consumed by the UAVs, within the range of about 36 kJ – 67 kJ as seen in Figure 6.7b. Figure 6.7c shows the number of connected users per UAV, ranging between about 10%–35% connected ground users per UAV. We observe that Agent 1 and Agent 3 connected to twice as many users as Agent 2 and Agent 4, possibly influenced by their initial starting position. The total number of connected ground users for all UAVs ranges between 85%–90%. Figure 6.7d, Figure 6.7e and Figure 6.7f show the convergence of the EE, fairness index and area covered by the UAVs after about the 80^{th} episode, respectively.



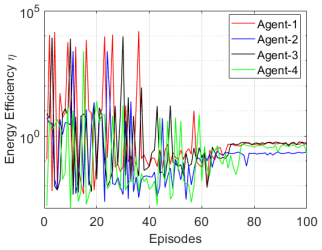
(a) Cumulative reward per agent-controlled UAV vs. episodes.



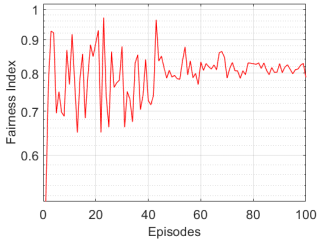
(b) Total energy consumed per agent-controlled UAV vs. episodes.



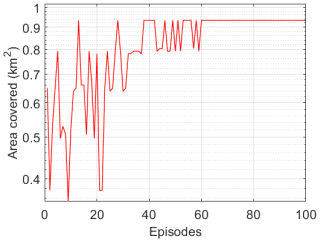
(c) Number of connected users per agent-controlled UAV vs. episodes



(d) Energy efficiency per agent-controlled UAV vs. episodes.



(e) Fairness Index vs. episodes.



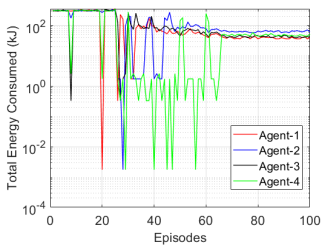
(f) Area covered vs. episodes.

Figure 6.7: Four agent-controlled UAVs serving 400 uneven randomly distributed ground users (200 static and 200 mobile following the RW mobility model).

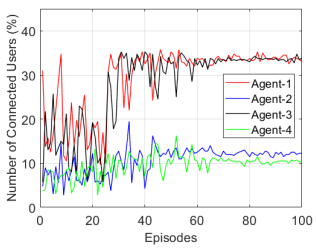
Again, we consider a dynamic setting with uneven randomly distributed ground users, where we have a combination of 200 static users and 200 mobile users following the random walk (RWP) mobility model. Figure 6.8 shows the performance of our DQLSI algorithm measured using the reward, total energy consumed, number of connected users, EE, fairness index and

area covered metrics. As seen in Figure 6.8a, all four agents try to maximise their cumulative reward through the learning episodes. After the 80^{th} episode, we observe significant convergence in the energy consumed by the UAVs, within the range of about 36 kJ – 71 kJ as seen in Figure 6.8b. Figure 6.8c shows the number of connected users per UAV, ranging between about 10%–35% connected ground users per UAV. Again, we see that Agent 1 and Agent 3 connected to twice as many users as Agent 2 and Agent 4, which is possibly influenced by their initial starting position. The total number of connected ground users for all UAVs ranges between 88%–92%. Figure 6.8d, Figure 6.8e and Figure 6.8f show the convergence of the EE, fairness index and area covered by the UAVs after about the 80^{th} episode, respectively. The fairness index significantly drops in the uneven dynamic setting compared to the even dynamic setting as seen in Figure 6.8e and Figure 6.5e. Despite the variation in the number of connected users by each UAV as seen in Figure 6.8c, we see from Figure 6.8d that each UAV's EE converges after a series of episodes. Next, we compare the proposed DQLSI algorithm with the ES, IS and CQL baselines under this setting.

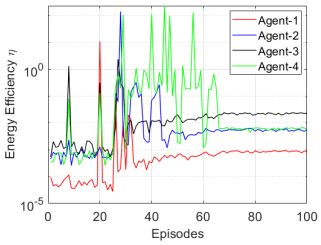
(a) Cumulative reward per agent-controlled UAV vs. episodes.



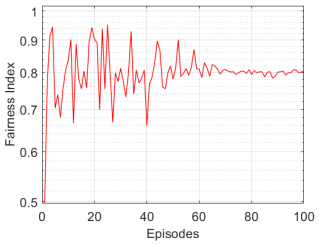
(b) Total energy consumed per agent-controlled UAV vs. episodes.



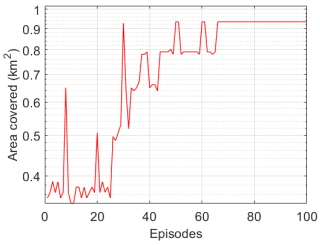
(c) Number of connected users per agent-controlled UAV vs. episodes



(d) Energy efficiency per agent-controlled UAV vs. episodes.



(e) Fairness Index vs. episodes.

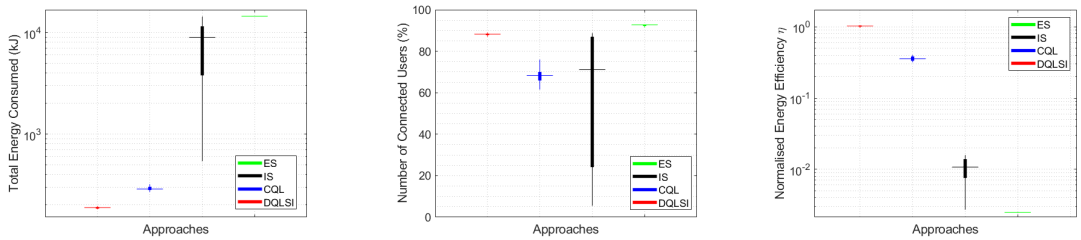


(f) Area covered vs. episodes.

Figure 6.8: Four agent-controlled UAVs serving 400 uneven randomly distributed ground users (200 static and 200 mobile following the RWP mobility model).

We further evaluate the effectiveness of the DQLSI in addressing the research question RQ1

by comparing it with centralised baselines that rely on a CC for decision-making. We consider the deployment of four agent-controlled UAVs deployed to serve a combination of static and mobile users, where the mobile users follow the RWP mobility model. Figure 6.9 shows the comparative plots measured using the total energy consumed, number of connected users and EE metrics. From Figure 6.9b the ES method achieves the highest number of connected users of about 96% which comes at an energy cost in the order of hundreds of KiloJoules as seen in Figure 6.9a. Intuitively, the high energy cost is due to the exploration of all possible combinations of the search space by the UAVs. Furthermore, the ES's poor energy performance is reflected in its poor EE as seen in Figure 6.9c. Interestingly, the proposed DQLSI approach which achieved about 88% in the average number of connected users outperforms the CQL approach which achieved about 65%. The performance in the CQL was significantly better in the unevenly distributed users scenario as compared to the evenly distributed users scenario. Intuitively, the probability of a UAV located in a highly congested user to provide significantly higher coverage is increased since the centroid will be located around densely concentrated user locations. Our proposed DQLSI agent performed in this uneven users' distribution, demonstrating the agent-controlled UAVs' ability to collaborate while improving coverage in this dynamic setting. Like in the evenly distributed users scenario, our decentralised approach was able to conserve much more energy than the centralised baselines. The DQLSI approach is robust to the mobility of ground users as well as uneven user distribution. The IS performed better than the ES in terms of total energy consumed and EE, however, it achieved the least coverage among all approaches. We observe that both DQLSI and CQL which are learning-based approaches performed well in optimising the total system's EE while jointly maximising the total energy utilisation and number of connected unevenly-distributed dynamic users in the network. The proposed DQLSI significantly improves the connectivity in the network and does not rely on a CC for decision-making. Next, we provide an evaluation summary of the DMARL with independent learning agents using the proposed DQLSI algorithm.



(a) Total energy consumed vs. approaches.

(b) Number of connected users vs. approaches.

(c) Normalised Energy efficiency vs. approaches.

Figure 6.9: Comparing the proposed DQLSI with centralised baselines while deploying four agent-controlled UAVs to serve 200 static and 200 mobile randomly distributed uneven ground users. The plots are based on the overall performance of all four agent-controlled UAVs. 5 trained samples each were gathered from 20 independent runs.

6.4.4 Evaluation Summary for Independent Learning Agents

Overall, we observe good connectivity when the proposed DQLSI algorithm is applied to all 3 scenarios. However, there was a significant drop in the number of connected ground users in the dynamic settings as compared to the static setting, due to the quickly-evolving network topology, emphasizing the importance of building approaches like the variants coming up later in this chapter that accounts for the mobility of ground users without having prior knowledge of the locations of each user via a CC. Our decentralised DQLSI outperformed the centralised baselines that rely on the CC for UAVs' decision making in terms of improving the total EE of UAVs over all settings considered. As expected, our DQLSI outperformed the cluster-based Q-learning approach in the static users, dynamic and even distribution of users, and in dynamic and uneven distribution of users by as much as 36%, 81% and 43%, respectively. Nevertheless, we observe that these centralised approaches outperformed our DQLSI approach in terms of coverage in the static settings. However, the ES outperformed all other approaches in terms of coverage over all settings. We observe that the deployment of four UAVs in the network does not guarantee a 100% coverage. We understand that our DQLSI approach may not always converge to a global optimum due to the non-stationarity in the environment. The non-stationarity occurs when ground users change positions and also when the actions of the agent-controlled UAVs conflict with already learnt policies. Nevertheless, it can be observed that the DQLSI algorithm may be a good choice in energy-constrained application, however, it may not guarantee a global optimal solution in terms

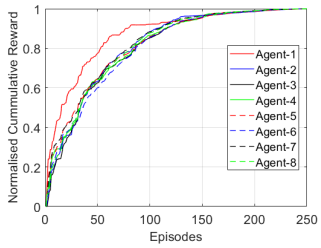
of coverage over all settings. We demonstrate that the DMARL with independent learning agents (DQLSI) answers the research question RQ1. The agent-controlled UAVs can learn to jointly maximise the number of connected static and mobile ground users while improving the total system's energy efficiency in the network. Unlike previous work, we do not assume global spatial knowledge of the locations of ground users. The performance of the proposed DQLSI approach was compared to state-of-the-art centralised approaches under realistic network conditions. Our proposed DMARL is robust in simultaneously improving the number of connections and minimizing the total energy consumed by UAVs in both static and dynamic environments.

6.5 Evaluation of Collaborative Agents

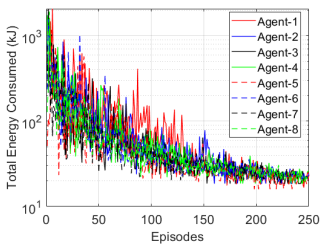
In this section, we evaluate the collaborative DMARL with two variants, *indirect collaborative agents* and *direct collaborative agents* to fully answer the research question RQ2. These DMARL variants support collaborative behaviours among agent-controlled UAVs to maximise the total system's EE while jointly optimising the UAVs' flight trajectory, the number of connected ground users and the energy utilisation of the UAVs. In a shared, dynamic and interference-limited environment like this, agent-controlled UAVs may exhibit selfish behaviours which may impact the overall system's EE. Hence, it becomes imperative to provide strategies that foster collaborations while improving performance gains. We consider an environment settings where fully decentralised agent-controlled UAVs are deployed to serve ground users in a 1 km^2 area of Dublin, Ireland. Due to the difficulty in obtaining real-world data of pedestrians' positions, we assume the deployment of 400 randomly distributed users drawn from a set of bin location data in the area[8] of Dublin with coordinates around 53° 22' 9" N, 6° 14' 45" W [Dublin, 2021] along with synthetic data[9]. The bin location data[10], which we obtained from the open data store of Smart Dublin, provides a close estimate of the likely position of users in the considered area. Furthermore, due to the difficulty in obtaining non-sparse and temporal mobility traces, we adopt three mathematical-based mobility mod-

[8]Drumcondra South A is a residential area and inner suburb on the Northside of Dublin.
[9]The synthetic data are generated using pseudo-random number generators in Python.
[10]https://data.smartdublin.ie/

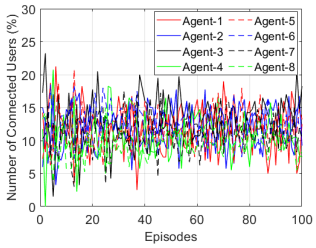
els widely used in ad-hoc network literature to depict the mobility of ground users, especially pedestrians. These models were introduced earlier in Sub-Section 3.1.3 of Chapter 3, namely: Random Walk (RW), Random Way Point mobility (RWP), Gauss–Markov Mobility (GMM) models [Camp et al., 2002]. We used the RW and RWP in the previous section to evaluate the effectiveness of the proposed DQLSI algorithm. Moreover, the RW was used in the baseline work [Liu et al., 2019a] to depict the mobility of ground users. Unless stated otherwise, we consider the GMM throughout this section due to its adaptability to different levels of randomness and ability to capture realistic movements better than the RW and RWP [Biomo et al., 2014, Solmaz and Turgut, 2019]. Next, we investigate the effectiveness of the DMARL variant with *Indirect Collaborative agents* and *Direct Collaborative agents* using the proposed MAD–DDQN and CMAD–DDQN algorithms, respectively.



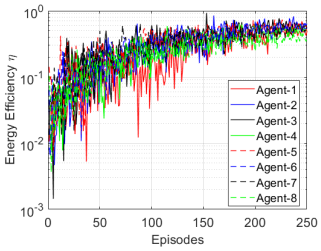
(a) Cumulative reward per agent-controlled UAV vs. episodes.



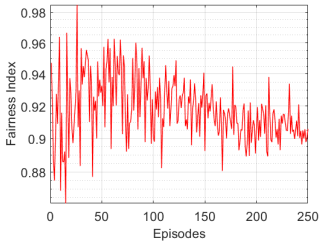
(b) Total energy consumed per agent-controlled UAV vs. episodes.



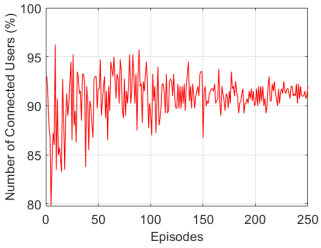
(c) Number of connected users per agent-controlled UAV vs. episodes



(d) Energy efficiency per agent-controlled UAV vs. episodes.



(e) Fairness Index vs. episodes.



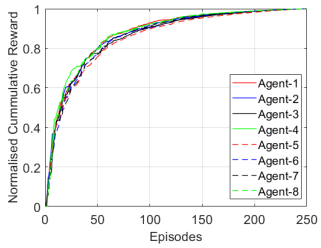
(f) Total number of connected users vs. episodes.

Figure 6.10: Learning behaviour of eight MAD–DDQN agent-controlled UAVs serving 400 randomly distributed ground users a 1 km^2 area of Dublin.

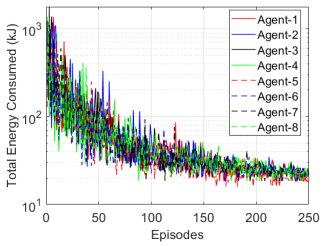
6.5.1 Dynamic Setting with Collaborative Agents with Individual Knowledge

First, we investigate the learning behaviour of collaborative agents with individual knowledge as presented in Section 4.2.2.1 of Chapter 4. We approach this using the MAD–DDQN algorithm which is a DMARL variant with *Indirect Collaborative agents*. Here, we consider the deployment of eight UAVs serving ground users to observe the performance of the agent-controlled UAVs over a series of time steps. To model the mobility of ground users, we consider a total of 200 mobile users following the GMM mobility model. This set of mobile users are comprised of 126 bin position data from an area in Dublin and padded with 74 synthetic data points. We then combine the 200 mobile users with 200 static users to make up 400 pedestrians in a 1 km^2 area. Unlike the previous section where we evaluated the effectiveness of our proposed DQLSI algorithm against the centralised baselines using four agent-controlled UAVs, here we increase the number of UAVs to eight. This helps us investigate the collaborative behaviour of the UAVs in a shared, dynamic and interference-limited environment. Interestingly, these agent-controlled UAVs do not have a direct communication mechanism for collaboration, however, they are incentivised to collaborate via the reward formulation. Figure 6.10 shows the performance of our MAD–DDQN algorithm measured using the reward, total energy consumed, number of connected users, EE, and fairness index in an environment with randomly deployed static and mobile ground users. As seen in Figure 6.10a, all eight agents try to maximise their cumulative reward through the learning episodes. Although we can observe that Agent 1 was able to maximise its reward faster than other agents, after about 180^{th} episode, they all converged. After the 200^{th} episode, we observe significant convergence in the energy consumed by the UAVs, within the range of about 16 kJ – 34 kJ as seen in Figure 6.10b. Figure 6.10c shows a balance in the connection load across the eight UAVs, ranging between about 5%–21% connected ground users per UAV. The total number of connected ground users for all UAVs ranges between 89%–93% as seen in Figure 6.10f. We observe that the deployment of eight UAVs in the network does not guarantee a 100% coverage. We understand that our collaborative agents approach may

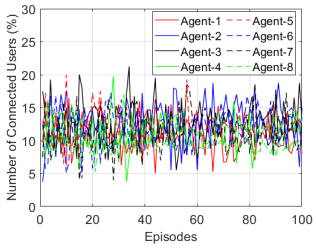
not converge to a global optimum due to the non-stationarity in the environment. Nevertheless, during experimentation, we observed that when the number of UAVs exceed 14 under same conditions, the UAVs were able to achieve the 100% coverage. However, deploying too many UAVs in a small coverage area will be counterproductive, resulting in additional cost, that is, in terms on energy and monetary cost. As such, it is desirable to have an optimal number of UAVs deployed to serve certain coverage areas. From Figure 6.10d and Figure 6.10e, we see convergence in the EE and fairness index after the 200^{th} episode, respectively. Therefore, this variant with *Indirect Collaborative agents* provides an answer to our second research question RQ2 through our MAD–DDQN algorithm. Next, evaluate the effectiveness of the DMARL variant with *Direct Collaborative agents* using the proposed CMAD–DDQN algorithm.



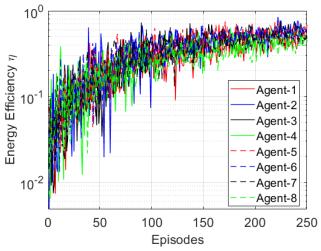
(a) Cumulative reward per agent-controlled UAV vs. episodes.



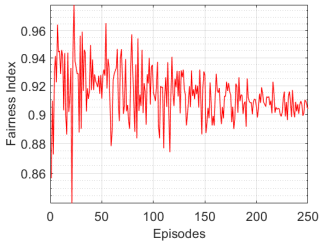
(b) Total energy consumed per agent-controlled UAV vs. episodes.



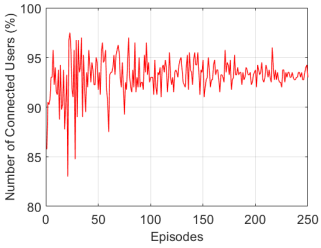
(c) Number of connected users per agent-controlled UAV vs. episodes



(d) Energy efficiency per agent-controlled UAV vs. episodes.



(e) Fairness Index vs. episodes.

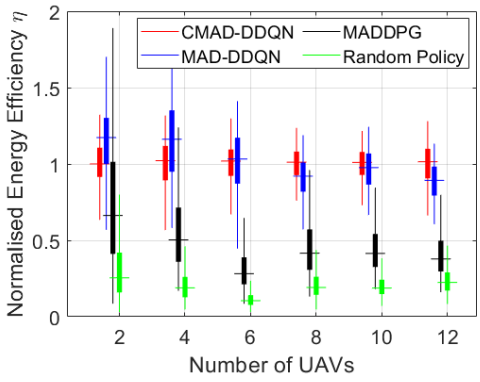


(f) Total number of connected users vs. episodes.

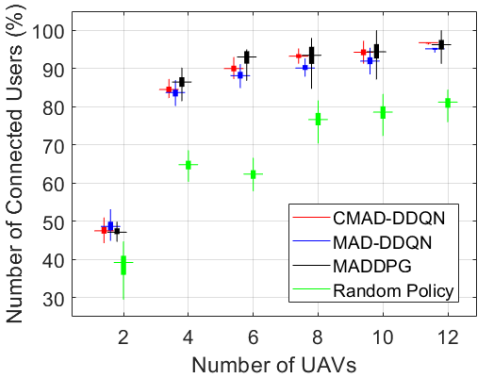
Figure 6.11: Learning behaviour of eight CMAD–DDQN agent-controlled UAVs serving 400 randomly distributed ground users a 1 km^2 area of Dublin.

6.5.2 Dynamic Setting with Collaborative Agents with Neighbour Knowledge

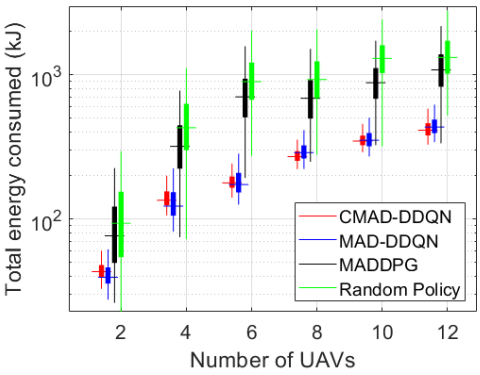
Here, we investigate the learning behaviour of collaborative agents with neighbour knowledge as presented in Section 4.2.2.3 of Chapter 4. We approach this using the CMAD–DDQN algorithm which is a DMARL variant with *Direct Collaborative agents*. Just like in the previous section, we consider the deployment of eight UAVs serving ground users to observe the performance of the agent-controlled UAVs over a series of time steps. Unlike the MAD–DDQN algorithm, the agents in this variant do have a direct communication mechanism for collaboration. Figure 6.11 shows the performance of our CMAD–DDQN algorithm measured using the reward, total energy consumed, number of connected users, EE, and fairness index in an environment with randomly deployed static and mobile ground users. As seen in Figure 6.11a, all eight agents try to maximise their cumulative reward through the learning episodes. We can see from Figure 6.11a that all agents were able to maximise their reward faster in the CMAD–DDQN algorithm as compared to Figure 6.10a which shows the MAD–DDQN algorithm. After the 200^{th} episode, we observe significant convergence in the energy consumed by the UAVs, within the range of about 16 kJ – 35 kJ as seen in Figure 6.11b. Figure 6.11c shows a balance in the connection load across the eight UAVs, ranging between about 6%–19% connected ground users per UAV. The total number of connected ground users for all UAVs ranges between 92%–95% as seen in Figure 6.11f. As discussed earlier, we do not achieve a 100% coverage when eight UAVs are deployed. This is due to the non-stationarity of the system. Nevertheless, when the number of deployed UAVs under the same conditions exceeds 14, we observed a 100% coverage. However, increasing number of UAVs may increase the deployment cost. From Figure 6.11d and Figure 6.11e, we see convergence in the EE and fairness index after the 200^{th} episode, respectively. Therefore, this variant with *Direct Collaborative agents* provides an answer to our second research question RQ2 through our CMAD–DDQN algorithm. Next, we investigate the effectiveness of our collaborative DMARL variants through the MAD–DDQN and CMAD–DDQN algorithms while answering the research question RQ2.



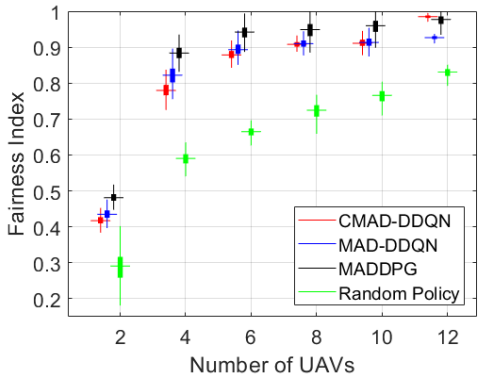
(a) Energy efficiency η vs. number of UAVs.



(b) Total number of connected ground users in the network vs. number of UAVs.



(c) Overall energy consumption by UAVs vs. number of UAVs.



(d) Fairness index vs. number of UAVs.

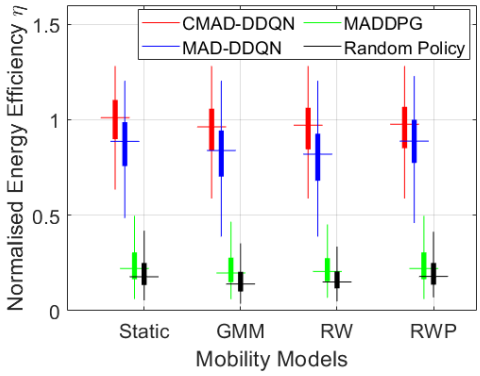
Figure 6.12: Impact of number of deployed UAVs on the UAVs' EE, number of connected ground users, fairness, and total energy consumed with 200 static and 200 mobile users deployed in a 1 km^2 area. The results shown are 2000 runs of trained agents deployed after training.

6.5.3 Investigating Number of UAVs Deployment over Baselines

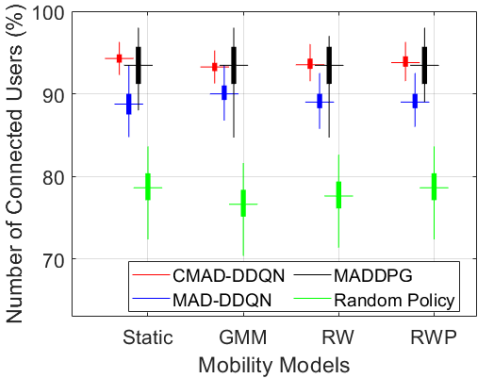
To observe how the proposed CMAD–DDQN approach performs while deploying varying numbers of UAVs in Figure 6.12, we compare the proposed CMAD–DDQN approach with baselines to evaluate the impact of different numbers of deployed UAVs on the EE, ground users connectivity and total energy consumed. Here, we vary the number of UAVs deployed to range between 2 to 12. Since we focus on comparing the EE values rather than showing their absolute values, we normalise the EE values with respect to the mean values of the proposed CMAD–DDQN approach. Figure 6.12a shows the plot of the normalised EE versus the number of deployed UAVs serving ground users. From Figure 6.12a, we observe that as more UAVs are deployed, the EE decreases in all approaches possibly because the system becomes

more unstable with more UAVs, decreasing the throughput as interference increases, and also takes longer to converge. However, the CMAD–DDQN approach outperforms the MAD–DDQN, MADDPG, and random policy approaches by approximately 15%, 65% and 85%, respectively. The proposed CMAD–DDQN approach on the other hand begins to outperform the MAD–DDQN approach only after the deployment of 8 UAVs. However, the CMAD–DDQN comes with an additional communication overhead as compared to the MAD–DDQN. From Figure 6.12, the communication overhead in the CMAD–DDQN approach results in a slight performance improvement in the evaluation metrics as the number of deployed UAVs is increased.

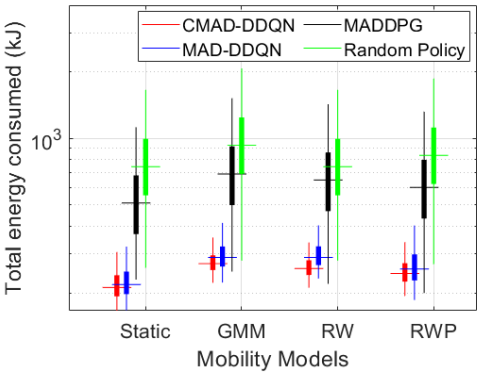
Figure 6.12b shows the plot of the number of connected users versus the number of deployed UAVs while comparing our proposed CMAD–DDQN approach with the baselines. From Figure 6.12b, we observe a marginally better performance by the MADDPG approach over the CMAD–DDQN and MAD–DDQN approaches in maximising the number of connected ground users by about 0.5% and 2%, respectively. However, the slight performance gain by the MADDPG comes at a huge computational training cost which is 8 times higher than the CMAD–DDQN and MAD–DDQN approaches. On the other hand, the random policy performed worst among the approaches in reducing connection outages, emphasizing the relevance of strategic decision-making in MARL problems. Figure 6.12c illustrates the plot of the total energy consumed versus the number of deployed UAVs serving ground users and clearly shows that the MAD–DDQN and CMAD–DDQN approaches significantly minimise the total energy consumed by all UAVs as compared to the other baselines. Although the MADDPG approach performs better in terms of improving the number of connected users than the random policy, the approach trades energy consumption for improved coverage of ground users. Figure 6.12d shows the plot of the geographical fairness versus the number of deployed UAVs serving ground users. We observe an improvement in the fairness index when the number of UAVs is increased. We observed better performance in the fairness index for the MADDPG approach than the CMAD–DDQN and MAD–DDQN approaches when 10 or fewer UAVs are deployed. As 12 UAVs are deployed, the proposed CMAD–DDQN approach



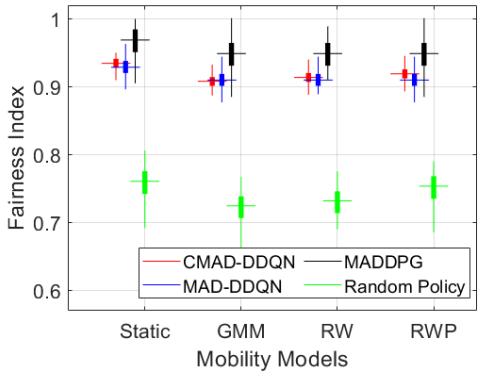
(a) Energy efficiency η vs. mobility models while under various approaches.



(b) Total number of connected ground users in the network vs. mobility models while under various approaches.



(c) Overall energy consumption by UAVs vs. mobility models while under various approaches.



(d) Fairness index vs. mobility models while under various approaches.

Figure 6.13: Impact of mobility models of 8 deployed UAVs on the EE, number of connected users, total energy consumed and fairness. For static, we consider 400 static users. For the GMM, RW and RWP we consider 200 static and 200 mobile users following the GMM, RW and RWP mobility models, respectively. The results shown are 2000 runs of trained agents deployed after training.

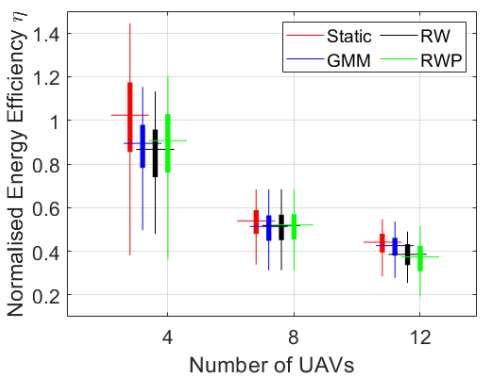
outperforms the other baselines in terms of fairness.

6.5.4 Investigating Mobility models over Baselines

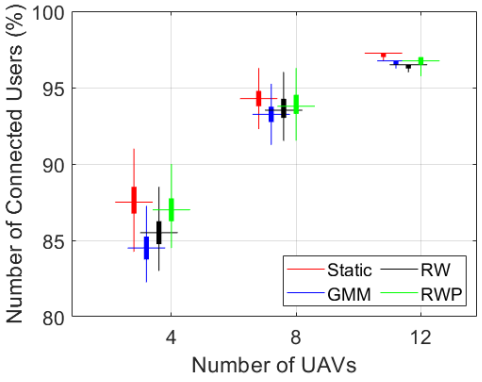
In a dynamic scenario where users may be mobile, the UAVs' locations need to be adjusted in such a way as to improve system performance. In Figure 6.13, we compare the proposed CMAD–DDQN and MAD–DDQN approaches with baselines to evaluate the impact of the various mobility models on the EE, number of connected ground users, the geographical fairness and total energy consumed when 8 UAVs are deployed to serve ground users in a 1 km^2 area. In Figure 6.12, we varied the number of deployed UAVs between 2 to 12, however,

we chose 8 UAVs as representatives to dig deeper into investigating the impact of different mobility models on the overall system's performance. Figure 6.13a shows the plot of the normalised EE versus the mobility models. The ground users' mobility models considered are the Static, GMM, RW and RWP models. Overall deployment of ground users using these mobility models, the proposed CMAD–DDQN approach outperforms the MAD–DDQN, MADDPG and Random Policy approach in terms of maximising the system's EE by about 15%, 75% and 85%, respectively. Figure 6.13b shows how the various mobility models impact the number of connected users while comparing the proposed CMAD–DDQN approach with the baselines. In all mobility models considered, the MADDQN approach performed closely to the proposed CMAD–DDQN approach. However, the MADDQN approach experience very good coverage performance, it had a larger variance than the CMAD–DDQN approach. Our proposed CMAD–DDQN approach converged to a significantly better average over multiple experimental runs.

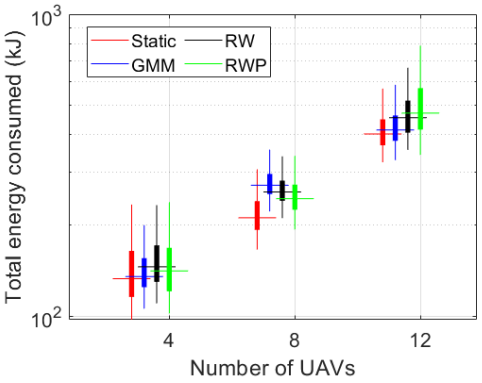
Figure 6.13c shows the plot of the total energy consumed versus the mobility models while comparing the performance of our proposed CMAD–DDQN with the baselines. Understandably, the random policy consumed the most amount of energy overall mobility models examined. The CMAD–DDQN approach consumes a lesser amount of energy in the static scenario than in the GMM, RW and RWP by about 25%, 20% and 15%, respectively. Although the MADDPG approach performed well in improving the number of connections, it performed poorly in minimizing the total energy consumed. Figure 6.13d shows the plot of the geographical fairness versus the mobility models. The CMAD–DDQN approach performed better than the MAD–DDQN and random policy but worse than the MADDPG approach. The MADDPG approach performs better due to the lengthy amount of time it takes for the achieve a good coverage performance. This implies that during this period more number of ground users will be fairly served. As to be expected, we observed that all approaches performed slightly better in the static scenario, which implies that decision-making in the scenarios that consider the mobility of ground users is worse overall approaches.



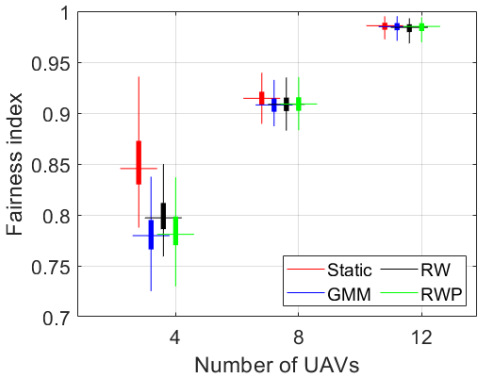
(a) Energy efficiency η vs. number of UAVs.



(b) Total number of connected ground users in the network vs. number of UAVs.



(c) Overall energy consumption by UAVs vs. number of UAVs.



(d) Fairness index vs. number of UAVs.

Figure 6.14: Impact of number of deployed UAVs on the UAVs' EE, number of connected ground users, fairness, and total energy consumed while varying mobility scenarios across Drumcondra area of Dublin. For static, we consider 400 static users. For the GMM, RW and RWP we consider 200 static and 200 mobile users following the GMM, RW and RWP mobility models, respectively. The results shown are 2000 runs of trained agents deployed after training.

6.5.5 Investigating the Deployment of UAVs over Mobility Models

Previously, we see that the mobility of users may have some significant impact on the overall system performance. From Figure 6.13 we observe that the proposed CMAD–DDQN approach outperforms other approaches in terms of improving the total EE of the UAVs. As we expected, direct communication provides the agents with better insights about the neighbours and may improve the agents' performance. However, communication comes at a cost. Here, we dive deeper to investigate the impact of deploying different numbers of UAVs while varying the mobility model while using CMAD–DDQN algorithm. Figure 6.14 shows graphs of the system's EE, number of connected users, total energy consumed and fairness

index versus the number of UAVs while varying the mobility models. Figure 6.14b shows the plot of the number of connected users versus the number of deployed UAVs while varying the mobility models. We observe improve connections as the number of UAVs are increase. Intuitively, more UAV access points can cover more ground users irrespective of the mobility scenario. However, we observe that the static scenario presents us with more connected ground users. We see the plot of the total energy consumed versus the number of deployed UAVs in Figure 6.14c, while Figure 6.14d shows the plot of the geographical fairness versus the number of deployed UAVs. Intuitively, when more UAVs are deployed to improve coverage, we observe increased geographical fairness, however this comes at an increased energy cost.

In the case of 4 and 8 UAVs deployment as seen in Figure 6.14c, we observe that the RWP model consumes slightly more energy than other models, while the static scenario consumes lesser energy. The fairness index in all mobility scenarios is even up when 12 UAVs are deployed to serve ground users. Figure 6.14a shows the plot of the normalised EE versus the number of deployed UAVs. We observe that as more UAVs are deployed, the system's EE drops across all mobility models. The intuition behind this is that as more UAVs are deployed we observe an increase in the energy consumed.

6.5.6 Evaluation Summary for Collaborative Agents

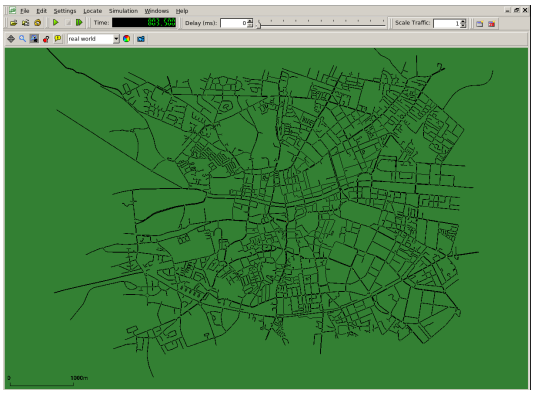
In this section, we demonstrate that the DMARL with collaborative agents provides an answer to the research question RQ2. We propose a collaborative Multi-Agent Decentralised Double Deep Q-Network (MAD–DDQN) and Communication-Enabled MAD–DDQN (CMAD–DDQN) variants of the DMARL to optimise the energy efficiency (EE) of a fleet of UAVs serving static and mobile ground users in a shared, dynamic and interference-limited environment. As we increase the number of UAVs in the network, the system's EE in the MAD–DDQN and CMAD–DDQN variants outperforms existing baselines in terms of energy utilisation and fairness, without degrading the coverage performance in the network. Both variants of the DMARL approach guarantee quick adaptability and convergence in a shared and dynamic network environment. Our proposed DMARL approach steadily converges faster

than the MADDPG approach, thereby leading to improved EE. However, the MADDPG approach outperforms our collaborative agents variants in terms of improving the number of connected ground users. This is because the MADDPG approach requires a lengthy training time and exploration of the environment to converge to good learning behaviours. This lengthy interaction of the MADDPG agent with the environment ensures improved coverage performance. However, the often comes at increased energy cost to achieve such coverage performance. We examine the robustness of the DMARL with collaborative agents over a state-of-the-art MARL approach while varying various mobility models and we observe a consistent improvement in the system's EE with a minimally deployed number of UAVs. We also demonstrate that the DMARL with collaborative agents significantly outperforms the random policy in terms of the total system's EE. This shows that our collaborative agents variants can be suitable in energy-constrained applications. The coverage improvement of the CMAD–DDQN variant over the MAD–DDQN variant comes at a communication cost. Although the periodic exchange of information among agents in the CMAD–DDQN variant can dramatically increase the entire system's communication overheads, it provides a performance guarantee for convergence in most multi-agent systems such as this one. In later sections, we will provide an analysis of the control overhead incurred.

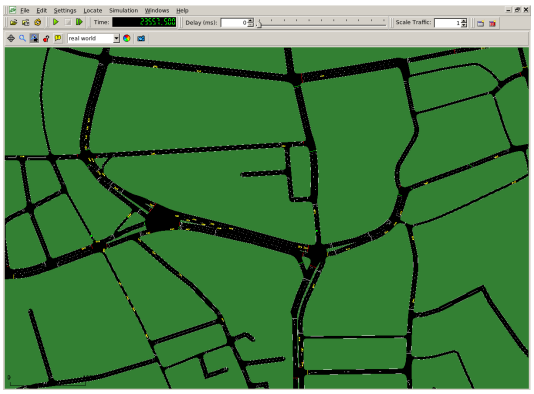
6.6 Evaluation of Collaborative Density-Aware Agents

In this section, we evaluate the collaborative density-aware DMARL variants, *indirect collaborative density-aware agents* (DAMAD–DDQN) and *direct collaborative density-aware agents* (DACEMAD–DDQN) to fully answer the research question RQ3. First, we investigate the deployment of agent-controlled UAVs to serve static toy users under different network configurations as seen in Figure A.1. Motivated by the findings, we investigate the performance of these density-aware DMARL variants in realistic urban environments. These DMARL variants support collaborative behaviours among agent-controlled UAVs to effectively serve highly mobile and densely uneven users' distribution. Specifically, these DMARL variants with collaborative density-aware agent-controlled UAVs aim to improve the total system's

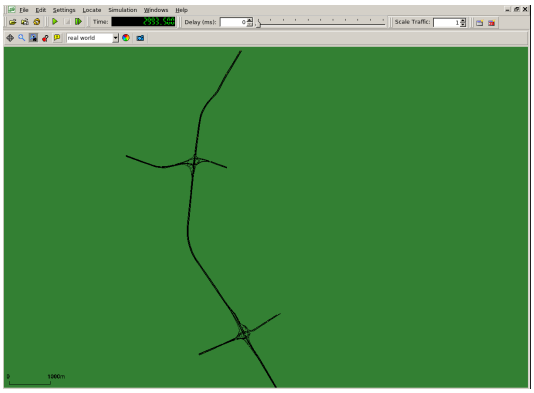
EE by jointly optimising the UAVs' flight trajectory, the number of connected ground users and the energy utilisation of the UAVs while keeping track of dense users' locations in the network. A majority of research consider the deployment of pedestrians who are confined to certain coverage area. Even in cases where the pedestrians are mobile, they move at quite low speed as compared to vehicular users that travel faster. In our previous variants, we limited the mobility of ground users to the coverage area. In this section, we investigate how agent-controlled UAVs can be deployed to serve road users. Users on road networks are often not confined to a fixed geographical space, since they may move in or out of the coverage area. Furthermore, we investigate the robustness of deploying UAVs to serve real-world vehicles and/or pedestrians using data obtained from the Dublin City council [Guériau and Dusparic, 2020]. We understand that users may be unevenly concentrated at certain locations in the road network and may require wireless connectivity in situations of outage in existing infrastructures.



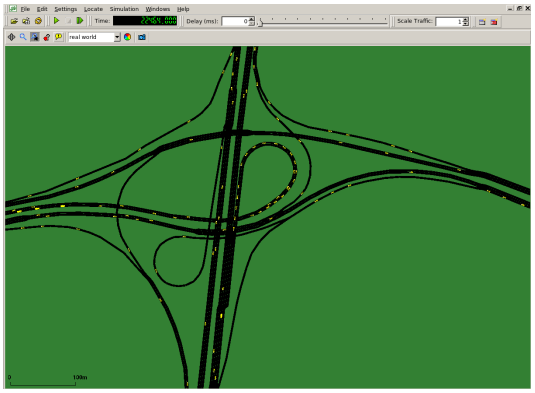
(a) Vehicles deployment in Dublin City centre.



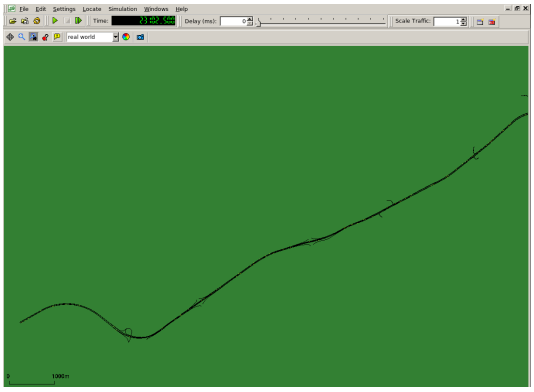
(b) Vehicles deployment in Dublin City centre (Zoomed in).



(c) Vehicles deployment along M50 motorway.



(d) Vehicles deployment along M50 motorway (Zoomed in).



(e) Vehicles deployment along the N7 national road.

(f) Vehicles deployment along the N7 national road (Zoomed in).

Figure 6.15: Screenshot of real traffic scenarios considered in Dublin, Ireland using Simulation of Urban MObility (SUMO).

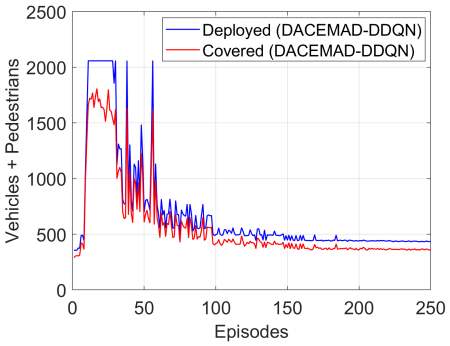
Hence, to effectively answer the research question, the DMARL variants are evaluated under three urban traffic scenarios shown in Figure 6.15, namely: *(i)* Deployment of UAVs to serve vehicles and pedestrians in a 3 km^2 area of Dublin city centre (DCC) as seen in Figure 6.15a. *(ii)* Deployment of UAVs to serve vehicles along a 7 km segment of the M50 motorway in Ireland as seen in Figure 6.15c. *(iii)* Deployment of UAVs to serve vehicles along a 6.5 km

segment of the N7 national road in Ireland as seen in Figure 6.15e. We further evaluate each of these scenarios under three traffic conditions: *(a)* Free flow traffic condition, usually early in the morning when traffic is quite low. *(b)* Saturated traffic condition, where the number of vehicles increases and traffic congestion begins to build. *(c)* Congested traffic condition, where there are a high number of vehicles on the road and often occurs during peak hours of the day.

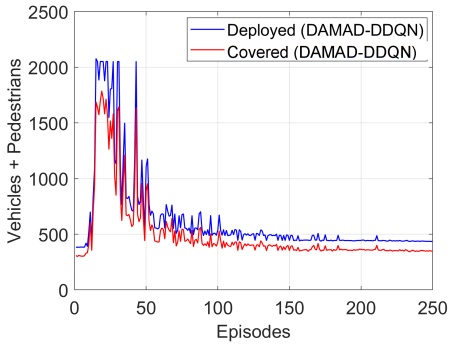
6.6.1 Urban Road Setting with Low Concentration of Vehicles and Pedestrians

To investigate the effectiveness of our DMARL solution in an urban road setting with a low concentration of vehicles and pedestrians, we deploy 10 UAVs in the DCC under the free flow scenario, where there is considerably low traffic on the road. As specified in Section 5.3, we consider 1342 pedestrians and 3179 vehicles as users injected into the considered DCC road network. From Figure 6.16, we see plots of deployed users and covered users against the learning episodes. Figures 6.16a, 6.16b, 6.16c, 6.16d show the learning behaviour of the DMARL variants with DACEMAD–DDQN, DAMAD–DDQN, CMAD–DDQN and MAD–DDQN agents over a series of time steps, respectively. After about 200^{th} episode, we observe convergence in the number of covered users with respect to the deployed users in the network.

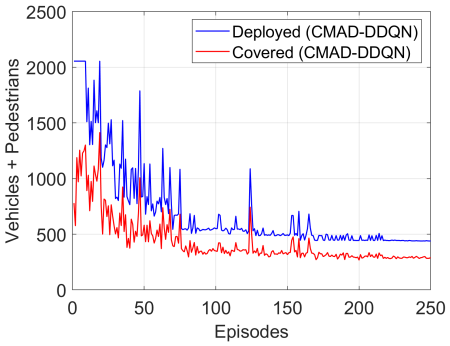
We go further to evaluate the performance of our DMARL by comparing it with the existing baseline under low traffic conditions around the DCC as seen in Figure 6.18. Figure 6.18a shows the graph of the CDR versus the learning episodes. We see a slightly better performance in the MADDPG approach as compared to our DACEMAD–DDQN variant. However, we observe performance dip at certain episodes. It should be noted that the MADDPG result shows the performance of already trained agents in the network. Hence we limit our illustration of the trained episodes of the MADDPG to Figures 6.18a, 6.18b and 6.18c. Nevertheless, the DACEMAD–DDQN variant outperforms the other variants in terms of CDR, however this performance gain comes with increased communication overhead of sharing neighbour observations. An agent-controlled UAVs may require certain information from their nearest



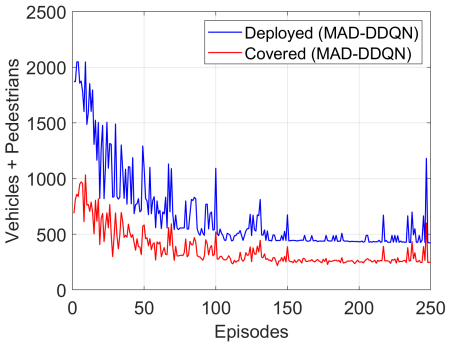
(a) Total number of road vehicles and pedestrians in the network vs. episodes (DACEMAD–DDQN).



(b) Total number of road vehicles and pedestrians in the network vs. episodes (DAMAD–DDQN).



(c) Total number of road vehicles and pedestrians in the network vs. episodes (CMAD–DDQN).



(d) Total number of road vehicles and pedestrians in the network vs. episodes (MAD–DDQN).

Figure 6.16: Impact of the proposed approach on the coverage behaviour in Low Traffic Conditions on the 3 km^2 Dublin City Centre, Ireland over learning episodes using 10 deployed UAVs.

neighbours for decision-making. Our closely related evaluation baseline [Liu et al., 2020] considered energy consumption, UAV positions, flying direction, coverage scores and distance of all UAVs. However, this implies additional overhead since each UAV has a global view of the entire state of other UAVs in the network. Our proposed approach is expected to reduce this overhead while providing UAVs with additional knowledge to improve the overall network performance. In the CMAD–DDQN variant, UAV j receives the neighbours' distances, energy levels, and connectivity score from closest neighbours within its communication range, which may help to improve its performance locally. The DACEMAD–DDQN variant on the other hand receives an additional observation, $\frac{C_o^t}{C_o^*}$, which is the ratio of the connectivity score in UAV j's neighbour at time-step t to the best neighbour connectivity score experienced over a series of past encounters.

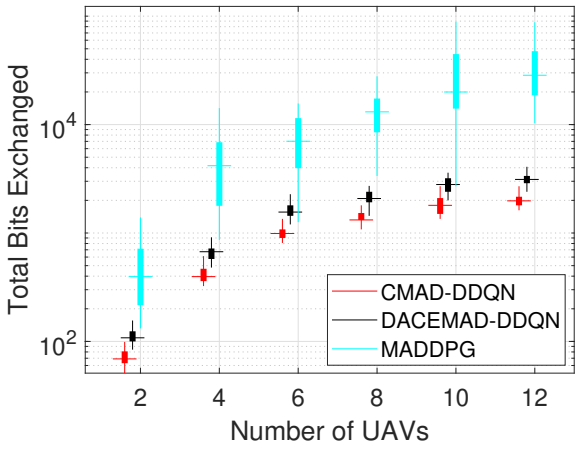
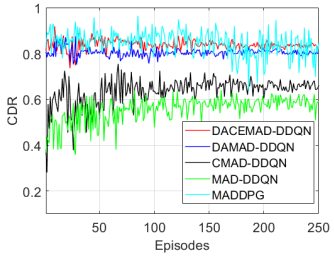


Figure 6.17: Total bits exchanged vs. number of UAVs. The result evaluates the total overhead incurred by agent-controlled UAVs for decision making. The results shown are 2000 runs of trained agents deployed after training.

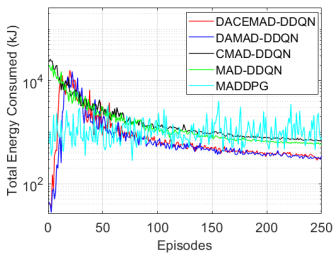
As expected, the total bits exchanged in the network is increased as the number of agent-controlled UAVs increases as seen in Figure 6.17. Furthermore, we observe that our proposed DACEMAD–DDQN and CMAD–DDQN variants performed significantly better than the MADDPG approach in terms of reducing the total amount of bits exchanged during trained deployment. This performance improvement is achieved over the number of UAVs deployments. Intuitively, the overhead incurred by the MADDPG approach is bounded by the overall number of UAVs deployed, thus leading to rapidly-growing control overhead. On the other hand, since our DACEMAD–DDQN and CMAD–DDQN variants consider the communication of an agent-controlled UAV with nearest neighbours, we observe a significant reduction in the total amount of bits exchanged after about 8 UAVs are deployed. Nevertheless, we understand that an increase in the control overhead may impact on the energy consumption of certain applications.

Interestingly, the DAMAD–DDQN variant outperforms both MAD–DDQN and the CMAD–DDQN variants, showing the significance of the agent-controlled UAVs to keep track of the best coverage locations while serving ground users which are unevenly distributed in the network. Figure 6.18d compares the CDR performance of our DMARL against the multi-agent deep deterministic (MADDPG) approach under free-flow traffic conditions in the DCC. We observe from Figure 6.18d that the MADDPG approach slightly outperforms the near-

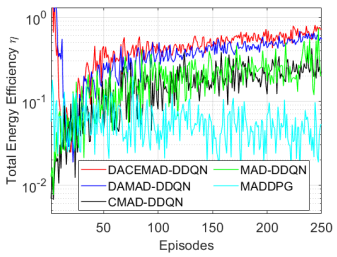
est best DMARL variant, the DACEMAD–DDQN, by an average of about 3% in terms of CDR. Figure 6.18b and Figure 6.18c show the total energy consumed and the total EE versus the learning episodes, respectively. We can see better performance in the DMARL variants over the MADDPG in terms of total energy consumed and the total system's EE. Intuitively, the performance gain of the MADDPG over our DMARL solution comes at some energy-associated cost. The DACEMAD–DDQN variant outperforms the other variants both in terms of EE and energy consumption. Figure 6.18f shows that the DACEMAD–DDQN outperforms in terms of EE the DAMAD–DDQN, CMAD–DDQN, MAD–DDQN, and MADDPG approaches by as much as 15%, 61%, 56% and 90%, respectively. From Figures 6.18e and 6.18f, we see a slightly better improvement by the MAD–DDQN over the CMAD–DDQN in terms of the total energy consumed and the total system's EE, respectively. These figures clearly show that our DMARL can jointly maximise the total system's EE and energy utilisation by UAVs much better without degrading the coverage performance as compared to the MADDPG that neglects interference from nearby UAV cells.



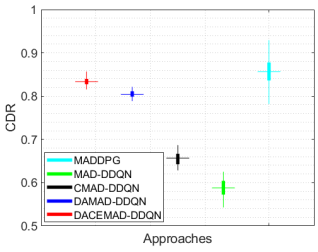
(a) CDR vs. Episodes.



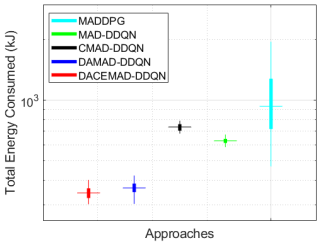
(b) Total Energy Consumed vs. Episodes.



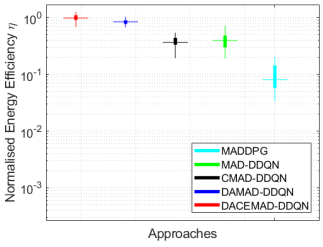
(c) Total Energy Efficiency vs. Episodes.



(d) CDR vs. Approaches.



(e) Total Energy Consumed vs. Approaches.



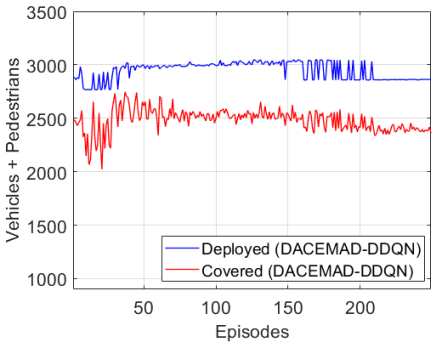
(f) Normalised Energy Efficiency vs. Approaches.

Figure 6.18: Comparative analysis using 10 deployed UAVs to serve vehicles along an area of DCC, Ireland under low traffic conditions.

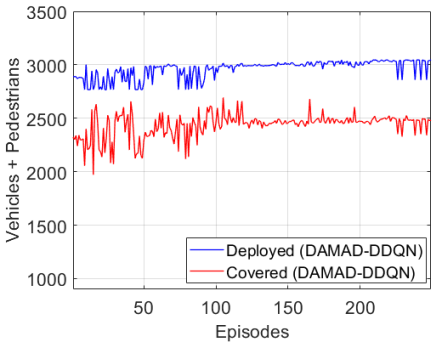
6.6.2 Urban Road Setting with Moderate Concentration of Vehicles and Pedestrians

To investigate the effectiveness of our DMARL solution in an urban road setting with a moderate concentration of vehicles and pedestrians, we deploy 10 UAVs in the DCC under saturated traffic conditions, where pedestrians and vehicles traffic begins to build. As specified in Section 5.3, we consider 14756 pedestrians and 27167 vehicles as users injected into the considered DCC road network. From Figure 6.19, we see plots of deployed users and the covered users against the learning episodes. Figures 6.19a, 6.19b, 6.19c, 6.19d show the learning behaviour of the DMARL variants with DACEMAD–DDQN, DAMAD–DDQN, CMAD–DDQN and MAD–DDQN agents over a series of time steps, respectively. After about 200^{th} episode, we observe convergence in the number of covered users with respect to the deployed users in the network.

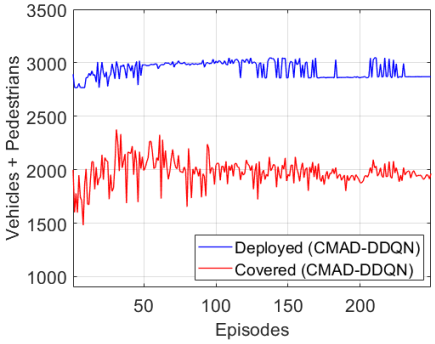
We go further to evaluate the performance of our DMARL by comparing it with the existing baseline under saturated traffic conditions around the DCC as seen in Figure 6.20. Figure 6.20a shows the graph of the CDR versus the learning episodes. The DACEMAD–DDQN variant outperforms the other variants in terms of CDR, however this performance gain comes with increased communication overhead of sharing neighbour observations. Although the performance of the DACEMAD–DDQN variant was closely matched by that of the MADDPG approach in terms of CDR. Nevertheless, the MADDPG performed better than other DMARL variants in terms of CDR. The DAMAD–DDQN variant outperforms both MAD–DDQN and the CMAD–DDQN variants, showing the significance of the agent-controlled UAVs to keep track of the best coverage locations while serving ground users which are unevenly distributed in the network. Figure 6.20d compares the CDR performance of our DMARL against the multi-agent deep deterministic (MADDPG) approach under saturated traffic conditions in the DCC. Figure 6.20b and Figure 6.20c show the total energy consumed and the total EE versus the learning episodes, respectively. The DACEMAD–DDQN variant outperforms the other variants both in terms of EE and energy consumption. Figure 6.20f shows that the DACEMAD–DDQN outperforms in terms of EE, the DAMAD–DDQN,



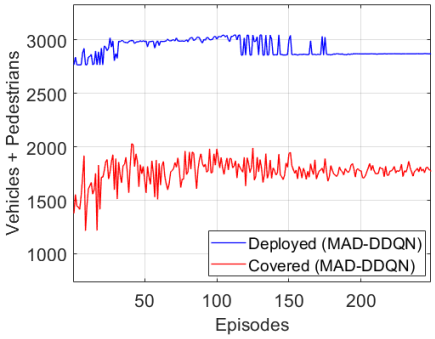
(a) Total number of road vehicles and pedestrians in the network vs. episodes (DACEMAD–DDQN).



(b) Total number of road vehicles and pedestrians in the network vs. episodes (DAMAD–DDQN).



(c) Total number of road vehicles and pedestrians in the network vs. episodes (CMAD–DDQN).



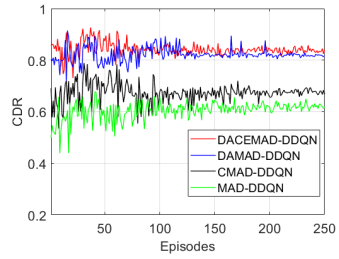
(d) Total number of road vehicles and pedestrians in the network vs. episodes (MAD–DDQN).

Figure 6.19: Impact of the proposed approach on the coverage behaviour in saturated Traffic Conditions on the 3 km^2 Dublin City Centre, Ireland over learning episodes using 10 deployed UAVs.

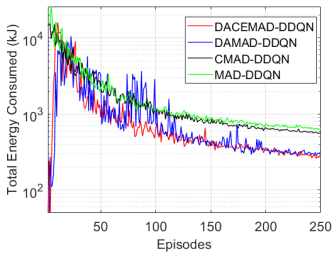
CMAD–DDQN, MAD–DDQN, and MADDPG approaches by as much as 10%, 42%, 55% and 94%, respectively. From Figures 6.20e and 6.20f, we see a slightly better improvement by the MAD–DDQN over the CMAD–DDQN in terms of the total energy consumed and the total system's EE, respectively. These figures clearly show that our DMARL can jointly maximise the total system's EE and energy utilisation by UAVs much better without degrading the coverage performance as compared to the MADDPG which neglects interference from nearby UAV cells.

6.6.3 Urban Road Setting with High Concentration of Vehicles and Pedestrians

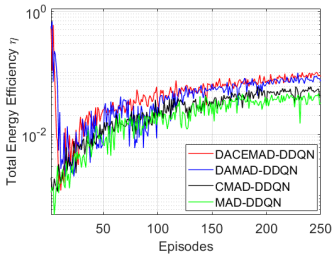
To investigate the effectiveness of our DMARL solution in an urban road setting with a high concentration of vehicles and pedestrians, we deploy 10 UAVs in the DCC under a



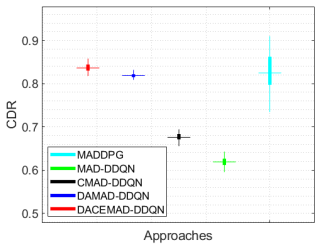
(a) CDR vs. Episodes.



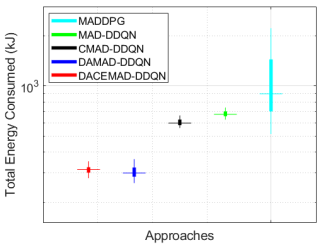
(b) Total Energy Consumed vs. Episodes.



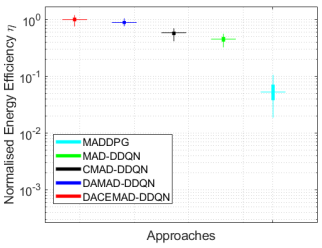
(c) Total Energy Efficiency vs. Episodes.



(d) CDR vs. Approaches.



(e) Total Energy Consumed vs. Approaches.

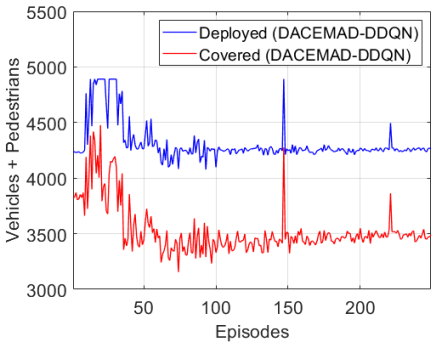


(f) Normalised Energy Efficiency vs. Approaches.

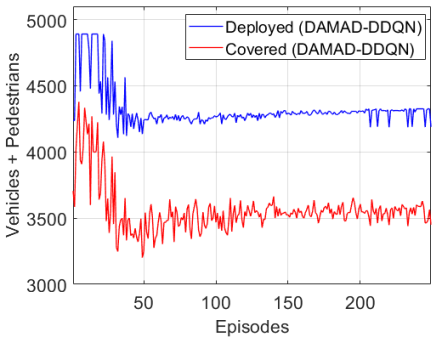
Figure 6.20: Comparative analysis using 10 deployed UAVs to serve vehicles and pedestrians along an area of DCC, Ireland under saturated traffic conditions.

congested scenario, where there is a considerably high number of users on the road. As specified in Section 5.3, we consider 15471 pedestrians and 27702 vehicles as users injected into the considered DCC road network. From Figure 6.21, we see plots of deployed users and covered users against the learning episodes. Figures 6.21a, 6.21b, 6.21c, 6.21d show the learning behaviour of the DMARL variants with DACEMAD–DDQN, DAMAD–DDQN, CMAD–DDQN and MAD–DDQN agents over a series of time steps, respectively. After about 200^{th} episode, we observe convergence in the number of covered users with respect to the deployed users in the network.

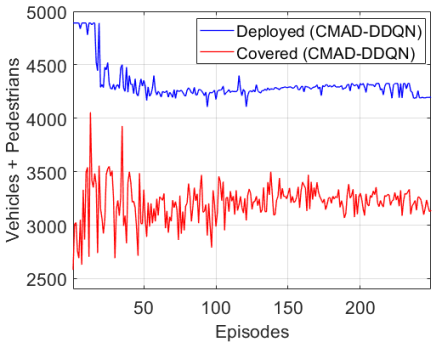
We go further to evaluate the performance of our DMARL solution by comparing it with the existing baseline under congested traffic conditions around the DCC as seen in Figure 6.22. Figure 6.22a shows the graph of the CDR versus the learning episodes. The DACEMAD–DDQN variant outperforms the other variants in terms of CDR, however this performance gain comes with increased communication overhead of sharing neighbour observations. Nevertheless, the DAMAD–DDQN variant outperforms both MAD–DDQN and the CMAD–DDQN variants, showing the significance of the agent-controlled UAVs to keep track of the best coverage locations while serving ground users which are unevenly distributed



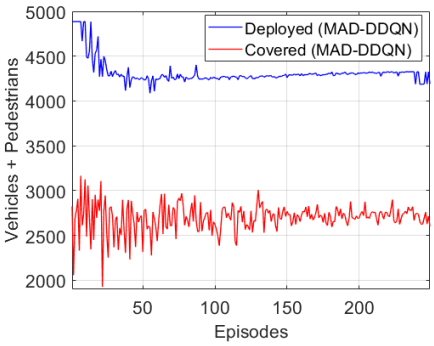
(a) Total number of road vehicles and pedestrians in the network vs. episodes (DACEMAD–DDQN).



(b) Total number of road vehicles and pedestrians in the network vs. episodes (DAMAD–DDQN).



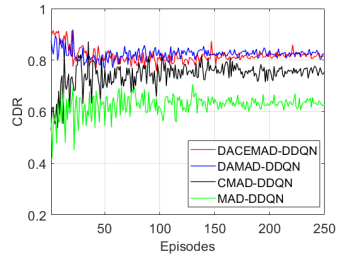
(c) Total number of road vehicles and pedestrians in the network vs. episodes (CMAD–DDQN).



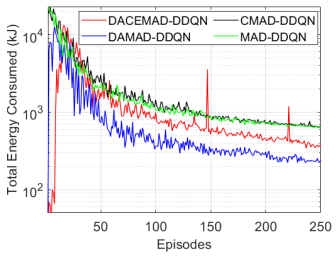
(d) Total number of road vehicles and pedestrians in the network vs. episodes (MAD–DDQN).

Figure 6.21: Impact of the proposed approach on the coverage behaviour in congested Traffic Conditions on the 3 km^2 Dublin City Centre, Ireland over learning episodes using 10 deployed UAVs.

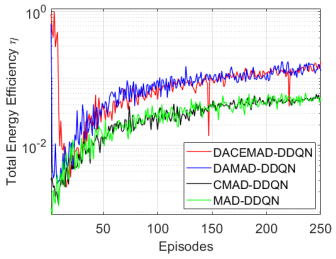
in the network. Figure 6.22d compares the CDR performance of our DMARL against the multi-agent deep deterministic (MADDPG) approach under congested traffic conditions in the DCC. Interestingly, we can see a slightly better performance of about 6% in terms of CDR of the MADDPG over the DACEMAD–DDQN and DAMAD–DDQN approaches. However, the MADDPG trades this coverage performance with higher energy cost as shown in Figure 6.22b. Figure 6.22b and Figure 6.22c show the total energy consumed and the total EE versus the learning episodes, respectively. The DACEMAD–DDQN variant outperforms the other variants in terms of the total energy consumed, however, the DAMAD–DDQN performed better than other approaches in terms of the total system's EE. Figure 6.22f shows that the DAMAD–DDQN outperforms in terms of EE, the DACEMAD–DDQN, CMAD–DDQN, MAD–DDQN, and MADDPG approaches by as much as 3%, 64%, 62% and 95%,



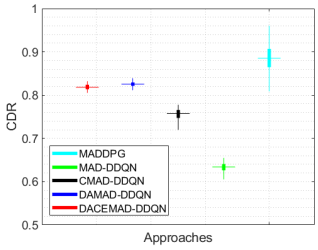
(a) CDR vs. Episodes.



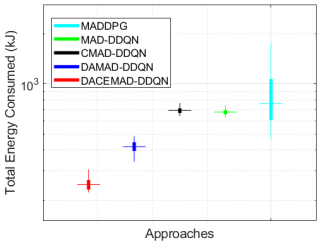
(b) Total Energy Consumed vs. Episodes.



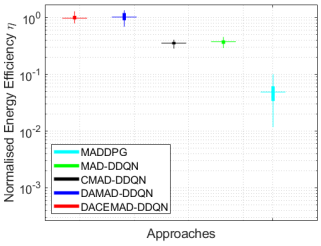
(c) Total Energy Efficiency vs. Episodes.



(d) CDR vs. Approaches.



(e) Total Energy Consumed vs. Approaches.



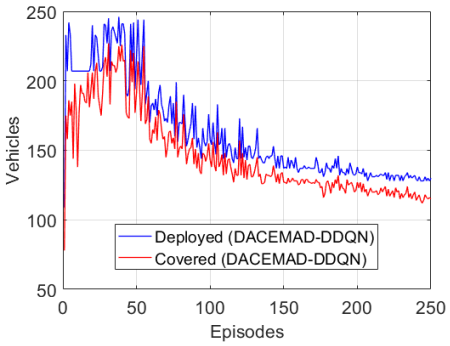
(f) Normalised Energy Efficiency vs. Approaches.

Figure 6.22: Comparative analysis using 10 deployed UAVs to serve vehicles along an area of DCC, Ireland under congested traffic conditions.

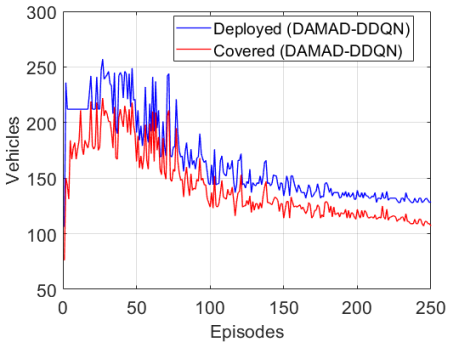
respectively. From Figures 6.22e and 6.22f, we see a slightly better improvement by the MAD–DDQN over the CMAD–DDQN in terms of the total energy consumed and the total system's EE, respectively. These figures clearly show that our DMARL can jointly maximise the total system's EE and energy utilisation by UAVs much better without degrading the coverage performance as compared to the MADDPG which neglects interference from nearby UAV cells.

6.6.4 Motorway Setting with Low Concentration of Vehicles

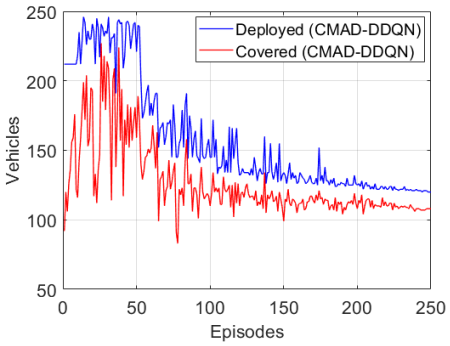
To investigate the effectiveness of our DMARL solution to serve sparse user distribution, we deploy 10 UAVs on the M50 motorway under the free flow scenario, where there is a considerably low number of vehicles on the road. Specifically, we consider vehicles as users in this section and do not consider pedestrians to be deployed on motorways. As specified in Section 5.3, we consider 1348 vehicles injected into the considered M50 motorway network. This implies the total number of vehicles that enter into the network. Nevertheless, it is only a few number of vehicles that arrive into the network in each time step. From Figure 6.23, we see plots of deployed vehicles and covered vehicles against the learning episodes. Figures 6.23a,



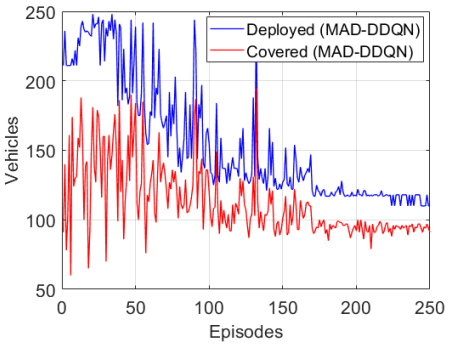
(a) Total number of road vehicles in the network vs. episodes (DACEMAD–DDQN).



(b) Total number of road vehicles in the network vs. episodes (DAMAD–DDQN).



(c) Total number of road vehicles in the network vs. episodes (CMAD–DDQN).

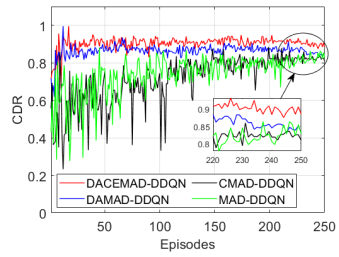


(d) Total number of road vehicles in the network vs. episodes (MAD–DDQN).

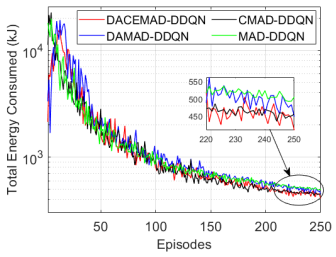
Figure 6.23: Low Traffic Conditions on the 7 km M50 motorway, Ireland over learning episodes using 10 deployed UAVs.

6.23b, 6.23c, 6.23d show the learning behaviour of the DMARL variants with DACEMAD–DDQN, DAMAD–DDQN, CMAD–DDQN and MAD–DDQN agents over a series of time steps, respectively. After about 200^{th} episode, we observe convergence in the number of covered vehicles with respect to the deployed vehicles in the network.

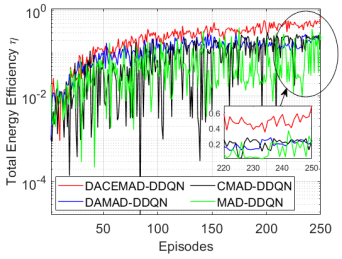
We go further to evaluate the performance of our DMARL by comparing it with the existing baseline under low traffic conditions along the M50 motorway as seen in Figure 6.24. Figure 6.24a shows the graph of the CDR versus the learning episodes. The DACEMAD–DDQN algorithm outperforms the other variants in terms of CDR, however this performance gain comes with increased communication overhead of sharing neighbour observations. Nevertheless, the DAMAD–DDQN algorithm outperforms both MAD–DDQN and the CMAD–DDQN algorithms, showing the significance of the agent-controlled UAVs to keep track of the best coverage locations while serving ground users which are unevenly distributed in the network.



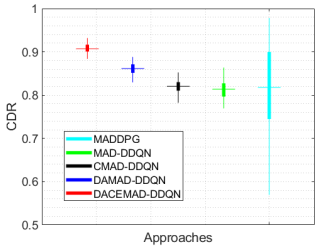
(a) CDR vs. Episodes.



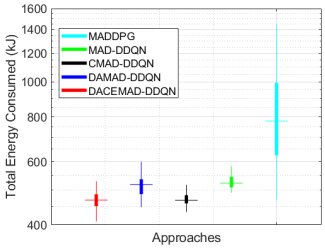
(b) Total Energy Consumed vs. Episodes.



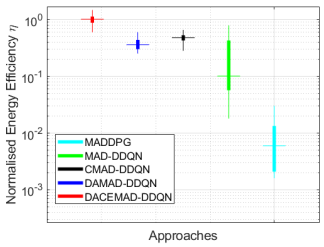
(c) Total Energy Efficiency vs. Episodes.



(d) CDR vs. Approaches.



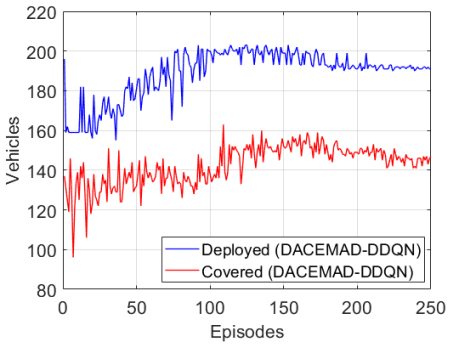
(e) Total Energy Consumed vs. Approaches.



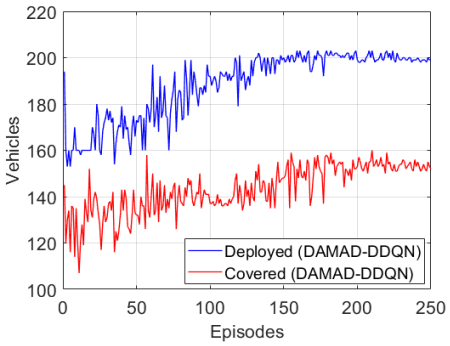
(f) Normalised Energy Efficiency vs. Approaches.

Figure 6.24: Comparative analysis using 10 deployed UAVs to serve vehicles along a segment of the M50 motorway, Ireland under low traffic conditions.

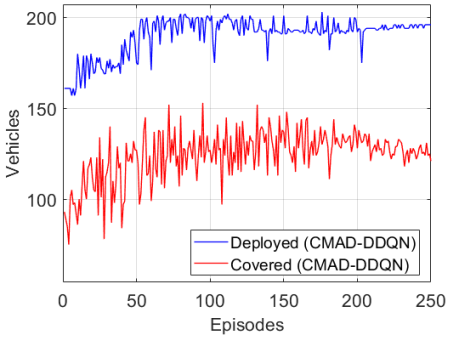
Figure 6.24d compares the CDR performance of our DMARL against the multi-agent deep deterministic (MADDPG) approach under free-flow traffic conditions on the M50 motorway. In terms of CDR, the DACEMAD–DDQN outperforms the DAMAD–DDQN, CMAD–DDQN, MAD–DDQN, and MADDPG approaches by as much as 4%, 9%, 10% and 10%, respectively. This confirms the ability of the DMARL to improve coverage performance in low-traffic conditions. Figure 6.24b and Figure 6.24c show the total energy consumed and the total EE versus the learning episodes, respectively. The DACEMAD–DDQN algorithm outperforms the other variants both in terms of EE and energy consumption. In particular, the DACEMAD–DDQN algorithm outperforms the MADDPG baseline, which performed worse, in terms of the total system's EE by about 98%. Figure 6.24e and Figure 6.24f clearly show that our DMARL can jointly maximise the total system's EE and energy utilisation by UAVs without degrading the coverage performance much better than the MADDPG that neglects interference from nearby UAV cells.



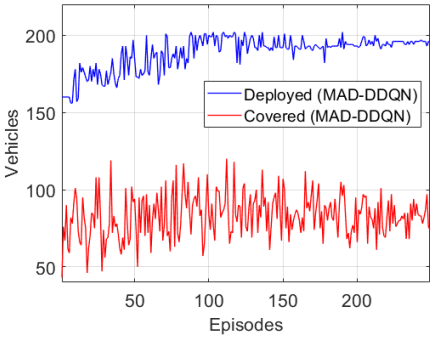
(a) Total number of road vehicles in the network vs. episodes (DACEMAD–DDQN).



(b) Total number of road vehicles in the network vs. episodes (DAMAD–DDQN).



(c) Total number of road vehicles in the network vs. episodes (CMAD–DDQN).



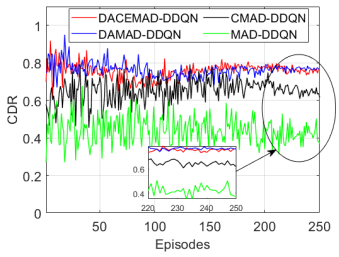
(d) Total number of road vehicles in the network vs. episodes (MAD–DDQN).

Figure 6.25: Impact of the proposed approach on the coverage behaviour in saturated traffic scenario of the 7 km M50 motorway over learning episodes using 10 deployed UAVs.

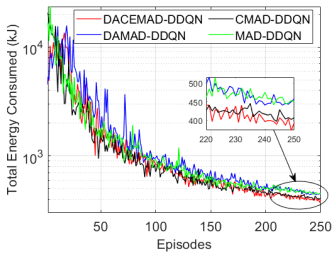
6.6.5 Motorway Setting with Moderate Concentration of Vehicles

To investigate the effectiveness of our DMARL solution in a motorway setting with a moderate concentration of vehicles, we deploy 10 UAVs on the M50 motorway under the saturated traffic condition, where the number of vehicles on the road continues to increase and traffic congestion on the road begins to build. In particular, we consider vehicles as users and do not consider pedestrians to be deployed on motorways. As specified in Section 5.3, we consider 23508 vehicles injected into the considered M50 motorway network. This implies the total number of vehicles that enter into the network. Nevertheless, it is only a few number of vehicles that arrive into the network in each time step. From Figure 6.25, we see plots of deployed vehicles and covered vehicles against the learning episodes. Figures 6.25a, 6.25b, 6.25c, 6.25d show the learning behaviour of the DMARL variants with DACEMAD–DDQN, DAMAD–DDQN, CMAD–DDQN and MAD–DDQN agents over a series of time

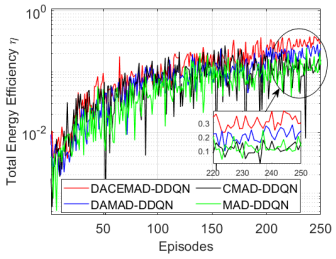
steps, respectively. After about 200^{th} episode, we observe convergence in the number of covered vehicles with respect to the deployed vehicles in the network.



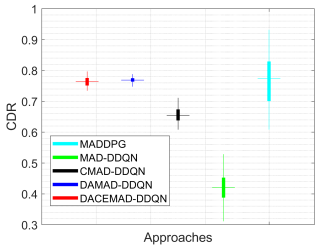
(a) CDR vs. Episodes.



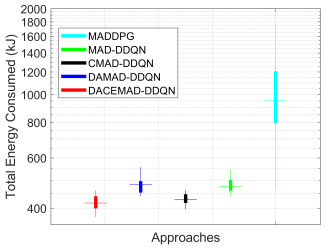
(b) Total Energy Consumed vs. Episodes.



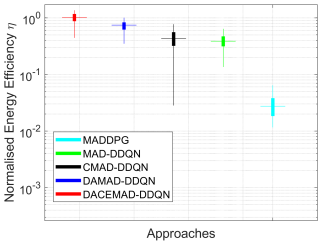
(c) Total Energy Efficiency vs. Episodes.



(d) CDR vs. Approaches.



(e) Total Energy Consumed vs. Approaches.



(f) Normalised Energy Efficiency vs. Approaches.

Figure 6.26: Comparative analysis using 10 deployed UAVs to serve vehicles along a segment of the M50 motorway under saturated traffic conditions.

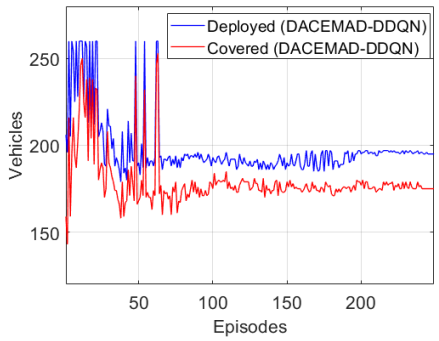
Here, we evaluate the performance of our DMARL solution by comparing it with the existing baseline under saturated traffic conditions along the M50 motorway as seen in Figure 6.26. Figure 6.26a shows the graph of the CDR versus the learning episodes. The DACEMAD–DDQN and DAMAD–DDQN variants performed closely well in terms of improving the CDR. The density-aware variants outperform both the CMAD–DDQN and MAD–DDQN variants. We observe that the CMAD–DDQN variant performed better than the MAD–DDQN variant, however, this comes at a communication cost. Figure 6.26d compares the CDR performance of our DMARL against the multi-agent deep deterministic (MADDPG) approach under saturated traffic conditions on the M50 motorway. We see a marginal performance improvement in terms of the CDR of the DACEMAD-DDQN algorithm over the DAMAD-DDQN and MADDPG approaches by an average value of about 1%. Meanwhile, the DACEMAD-DDQN algorithm outperforms the CMAD-DDQN and MAD-DDQN approaches by about 16% and 22%, respectively. Figure 6.26b and Figure 6.26c show the total energy consumed and the total EE versus the learning episodes, respectively. The DACEMAD–DDQN and CMAD–

DDQN variants, which have a direct communication mechanism, performed better than the DAMAD–DDQN and MAD–DDQN variants in terms of the total energy consumption as seen in Figure 6.26b. However, the density-aware variants achieved better total system's EE as seen in Figure 6.26c. Figure 6.26e shows that the DACEMAD–DDQN algorithm performs better in terms of the total system's EE over the DAMAD–DDQN, CMAD–DDQN, MAD–DDQN and MADDPG by as much as 27%, 57%, 58% and 96%, respectively. The Figures 6.26e and 6.26f clearly show that our DMARL can jointly maximise the total system's EE and energy utilisation by UAVs without degrading the coverage performance much better than the MADDPG that neglects interference from nearby UAV cells.

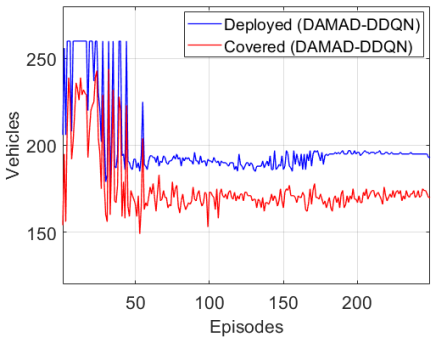
6.6.6 Motorway Setting with High Concentration of Vehicles

To verify the effectiveness of our DMARL solution in a motorway setting with a high concentration of vehicles, we deploy 10 UAVs on the M50 motorway under congested traffic conditions, where the number of vehicles on the road is at its peak. Here, we consider vehicles as users and do not consider pedestrians to be deployed on motorways. As specified in Section 5.3, we consider 25316 vehicles injected into the considered M50 motorway network. This implies the total number of vehicles that enter into the network. Nevertheless, it is only a few number of vehicles that arrive into the network in each time step. From Figure 6.27, we see plots of deployed vehicles and covered vehicles against the learning episodes. Figures 6.27a, 6.27b, 6.27c, 6.27d show the learning behaviour of the DMARL variants with DACEMAD–DDQN, DAMAD–DDQN, CMAD–DDQN and MAD–DDQN agents over a series of time steps, respectively. We observe convergence in the number of covered vehicles with respect to the deployed vehicles in the network after about 200^{th} episode.

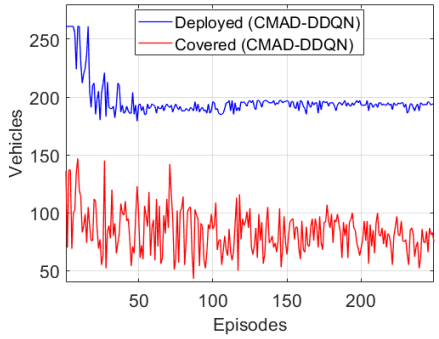
We evaluate the performance of our DMARL solution by comparing it with the existing baseline under congested traffic conditions along the M50 motorway as seen in Figure 6.28. Figure 6.28a shows the graph of the CDR versus the learning episodes. The DACEMAD–DDQN and DAMAD–DDQN variants performed closely well in terms of improving the CDR. The density-aware variants outperform both the CMAD–DDQN and MAD–DDQN variants. We see that the CMAD–DDQN variant outperforms the MAD–DDQN variant, however, in-



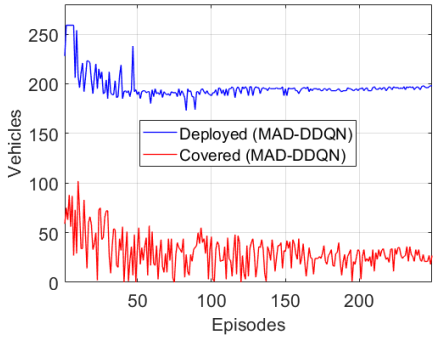
(a) Total number of road vehicles in the network vs. episodes (DACEMAD–DDQN).



(b) Total number of road vehicles in the network vs. episodes (DAMAD–DDQN).



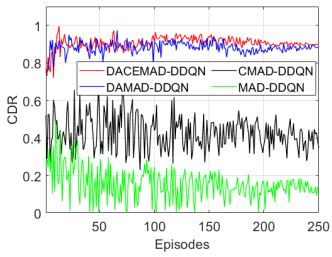
(c) Total number of road vehicles in the network vs. episodes (CMAD–DDQN).



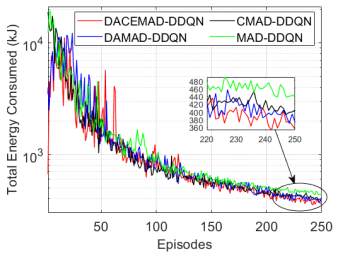
(d) Total number of road vehicles in the network vs. episodes (MAD–DDQN).

Figure 6.27: Impact of proposed approach on the coverage behaviour in congested traffic scenario of the 7 km M50 motorway, Ireland over learning episodes using 10 deployed UAVs.

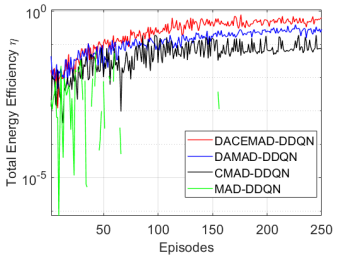
curs a higher communication overhead. Figure 6.28d compares the CDR performance of our DMARL against the MADDPG approach under congested traffic conditions of the M50 motorway. Under congested traffic conditions, we see that the density-aware variants of DMARL outperform the MADDPG in terms of CDR. Nevertheless, the MADDPG achieved better CDR than both the CMAD–DDQN and MAD–DDQN variants. Figure 6.28b and Figure 6.28c show the total energy consumed and the total system's EE versus the learning episodes, respectively. The total energy consumed in our DMARL was significantly minimised as seen in Figure 6.28b. From Figure 6.28f, the DACEMAD–DDQN outperforms the MADDPG approach by about 98% in terms of the total system's EE. Figures 6.28e and 6.28f validate that our DMARL consistently outperforms the MADDPG baseline, which performed worse in terms of the total energy consumed and the total system's EE.



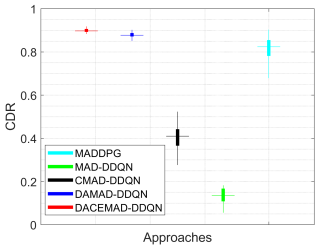
(a) CDR vs. Episodes.



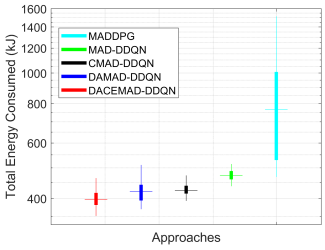
(b) Total Energy Consumed vs. Episodes.



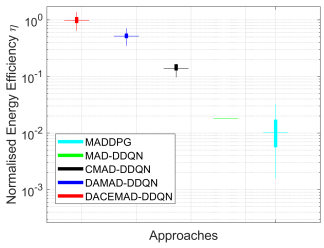
(c) Total Energy Efficiency vs. Episodes.



(d) CDR vs. Approaches.



(e) Total Energy Consumed vs. Approaches.



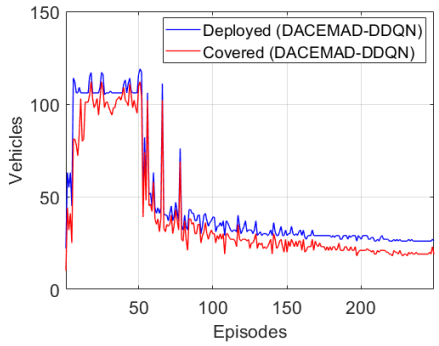
(f) Normalised Energy Efficiency vs. Approaches.

Figure 6.28: Comparative analysis using 10 deployed UAVs to serve vehicles along a segment of the M50 motorway under congested traffic conditions.

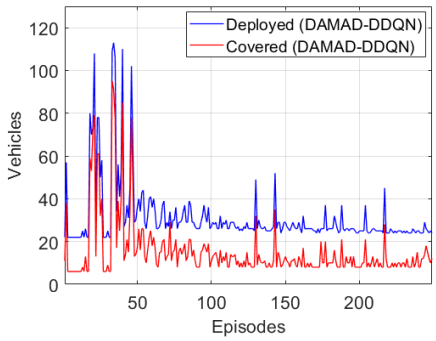
6.6.7 National Road Setting with Low Concentration of Vehicles

To investigate the effectiveness of our DMARL solution in a national road setting with a low concentration of vehicles, we deploy 10 UAVs on the N7 national road under the free flow scenario, where there is considerably low traffic on the road. The N7 differs from the M50 in terms of the traffic flow in the network and the concentration of vehicles in the network in each time step. Specifically, we consider vehicles as users and do not consider pedestrians to be deployed on the national road. As specified in Section 5.3, we consider 1236 vehicles injected into the considered N7 national road network. From Figure 6.29, we see plots of deployed vehicles and covered vehicles against the learning episodes. Figures 6.29a, 6.29b, 6.29c, 6.29d show the learning behaviour of the DMARL variants with DACEMAD–DDQN, DAMAD–DDQN, CMAD–DDQN and MAD–DDQN agents over a series of time steps, respectively. After about 200^{th} episode, we observe convergence in the number of covered vehicles with respect to the deployed vehicles in the network.

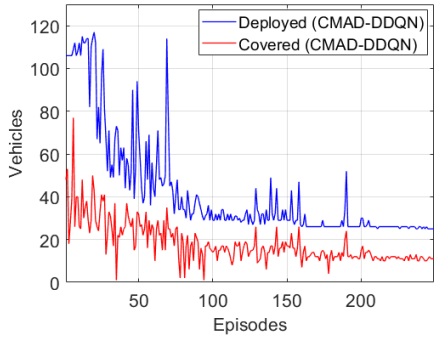
To evaluate the performance of our DMARL, we compare it with the existing baseline under free-flow traffic conditions along the N7 national road as seen in Figure 6.30. Figure



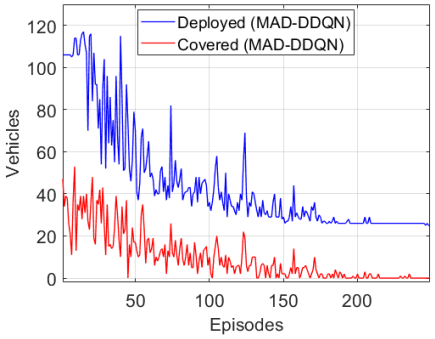
(a) Total number of road vehicles in the network vs. episodes (DACEMAD–DDQN).



(b) Total number of road vehicles in the network vs. episodes (DAMAD–DDQN).



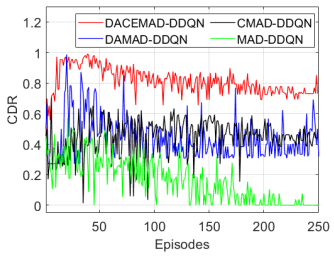
(c) Total number of road vehicles in the network vs. episodes (CMAD–DDQN).



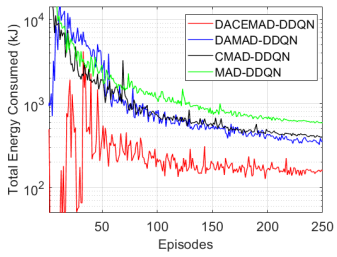
(d) Total number of road vehicles in the network vs. episodes (MAD–DDQN).

Figure 6.29: Impact of the proposed approach on the coverage behaviour in Low Traffic Conditions on the N7 national road over learning episodes using 10 deployed UAVs.

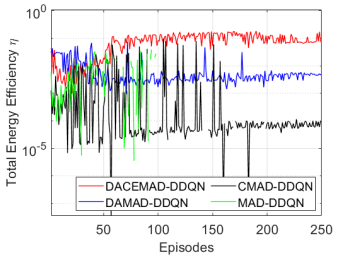
6.30a shows the graph of the CDR versus the learning episodes. In terms of CDR, the communication-enabled variants of our DMARL solution outperform their counterparts that have no direct communication mechanism. Intuitively, direct communication may be significant for collaborative behaviours that improve coverage performance. However, communication among the agent-controlled UAVs may result in increased communication overhead in the network, especially when agent-controlled UAVs share information with their nearest neighbours. We observe a very poor trend in the CDR of the MAD–DDQN variant which could be due to the UAVs' inability to serve sparsely and uneven users' distribution. On the other hand, the DACEMAD–DDQN was effectively able to keep track and provide coverage in such low traffic conditions. Figure 6.30d compares the CDR performance of our DMARL against the MADDPG approach under free-flow traffic conditions on the N7 motorway. Interestingly, we can see slightly better performance in the MADDPG approach over



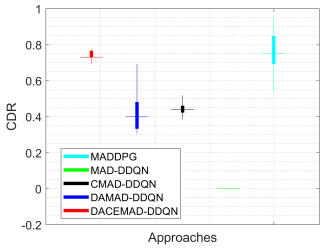
(a) CDR vs. Episodes.



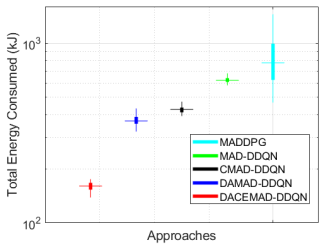
(b) Total Energy Consumed vs. Episodes.



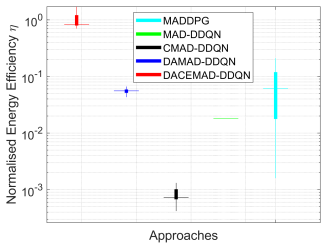
(c) Total Energy Efficiency vs. Episodes.



(d) CDR vs. Approaches.



(e) Total Energy Consumed vs. Approaches.



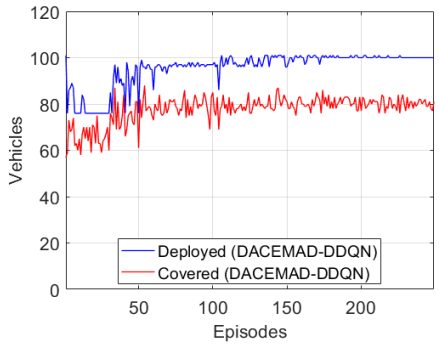
(f) Normalised Energy Efficiency vs. Approaches.

Figure 6.30: Comparative analysis using 10 deployed UAVs to serve vehicles along a segment of the N7 national road, Ireland under low traffic conditions.

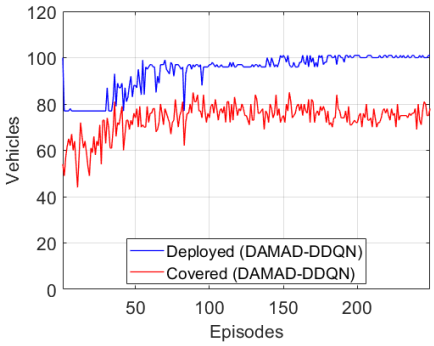
the DMARL variants in terms of CDR. Compared to the DMARL variants in Figure 6.30b, the MAD–DDQN variant performed worse. However, the MAD-DDQN variant was able to minimise the total energy consumed when compared to the MADDPG approach as seen in Figure 6.30e. Intuitively, the MADDPG approach trades its coverage gain with poor energy utilization. Figure 6.30c shows the total EE versus the learning episodes. From Figure 6.30f, the DACEMAD–DDQN algorithm outperforms the MADDPG approach by as much as 92%. We observe that our DMARL solution can jointly maximise the total system's EE and energy utilisation by UAVs without degrading the coverage performance much better than the MADDPG that neglects interference from nearby UAV cells.

6.6.8 National Road Setting with Moderate Concentration of Vehicles

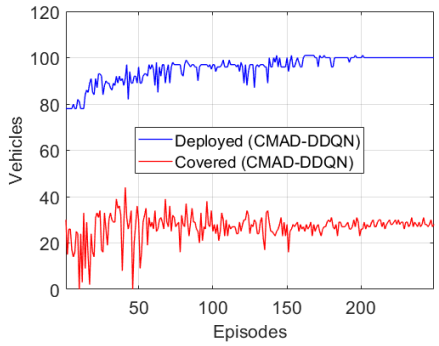
To investigate the effectiveness of our DMARL solution in a national road setting with a moderate concentration of vehicles, we deploy 10 UAVs on the N7 national road under the saturated traffic condition, where the number of vehicles on the road continues to increase and traffic congestion on the road begins to build. In particular, we consider vehicles as users and do not consider pedestrians to be deployed on the national road. As specified



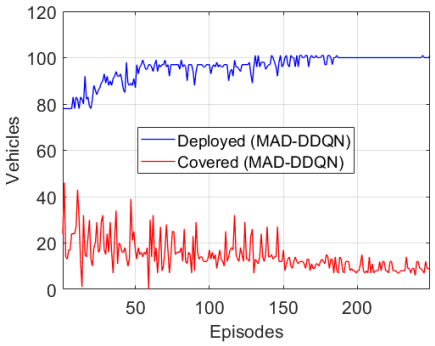
(a) Total number of road vehicles in the network vs. episodes (DACEMAD–DDQN).



(b) Total number of road vehicles in the network vs. episodes (DAMAD–DDQN).



(c) Total number of road vehicles in the network vs. episodes (CMAD–DDQN).

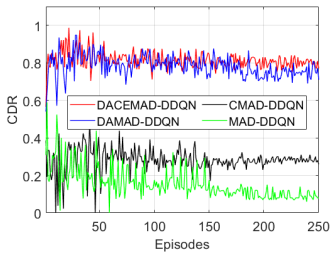


(d) Total number of road vehicles in the network vs. episodes (MAD–DDQN).

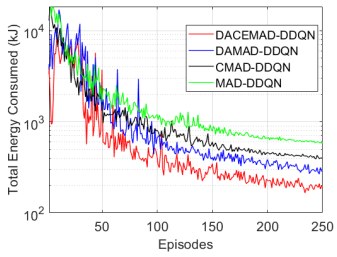
Figure 6.31: Impact of the proposed approach on the coverage behaviour in saturated traffic scenario of the N7 road, Ireland over learning episodes using 10 deployed UAVs.

in Section 5.3, we consider 12191 vehicles injected into the considered N7 national road network. From Figure 6.31, we see plots of deployed vehicles and covered vehicles against the learning episodes. Figures 6.31a, 6.31b, 6.31c, 6.31d show the learning behaviour of the DMARL variants with DACEMAD–DDQN, DAMAD–DDQN, CMAD–DDQN and MAD–DDQN agents over a series of time steps, respectively. After about 200^{th} episode, we observe convergence in the number of covered vehicles with respect to the deployed vehicles in the network.

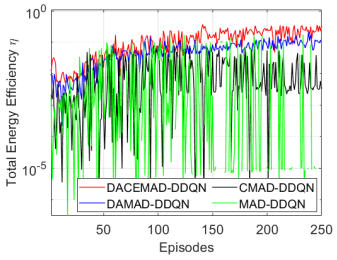
Here, we evaluate the performance of our DMARL solution by comparing it with the existing baseline under saturated traffic conditions along a segment of the N7 national road as seen in Figure 6.32. Figure 6.32a shows the graph of the CDR versus the learning episodes. The DACEMAD–DDQN and DAMAD–DDQN variants performed closely well in terms of improving the CDR. These density-aware variants outperform both the CMAD–DDQN and



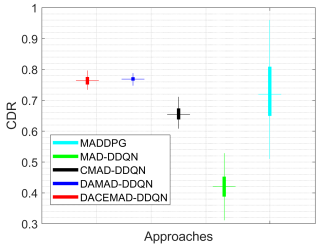
(a) CDR vs. Episodes.



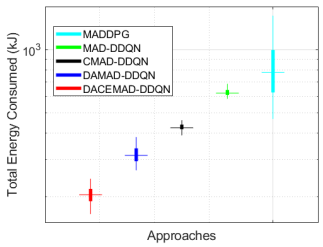
(b) Total Energy Consumed vs. Episodes.



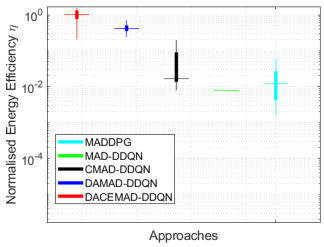
(c) Total Energy Efficiency vs. Episodes.



(d) CDR vs. Approaches.



(e) Total Energy Consumed vs. Approaches.

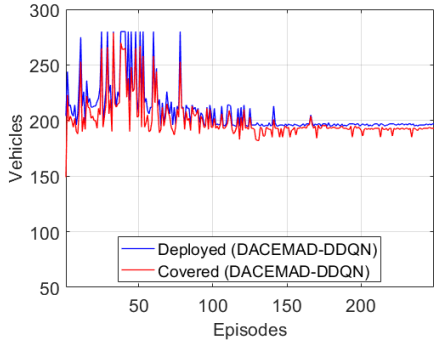


(f) Normalised Energy Efficiency vs. Approaches.

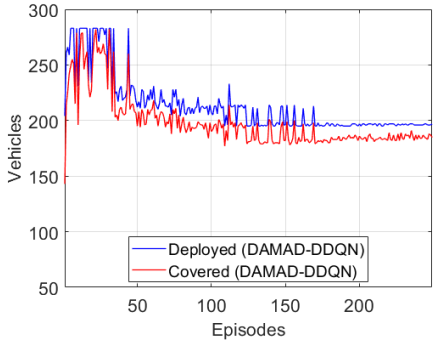
Figure 6.32: Comparative analysis using 10 deployed UAVs to serve vehicles along a segment of the N7 national road, Ireland under saturated traffic conditions.

MAD–DDQN variants, hereby showing their robustness in providing coverage to mobile and densely uneven users' distribution. We observe that the CMAD–DDQN variant performed better than the MAD–DDQN variant, however, this comes at a communication cost. Figure 6.32d compares the CDR performance of our DMARL against the multi-agent deep deterministic (MADDPG) approach under saturated traffic conditions of the N7 motorway. We observe slightly better performance improvement in the density-aware variants of DMARL over the MADDPG. Nevertheless, the MADDPG approach outperforms the CMAD–DDQN and the MAD–DDQN algorithms in terms of CDR. Figure 6.32b and Figure 6.32c show the total energy consumed and the total EE versus the learning episodes, respectively. The DACEMAD–DDQN and DAMAD–DDQN variants, which have a mechanism to track dense user locations, performed better than the CMAD–DDQN and MAD–DDQN variants in terms of the total energy consumption and total EE as seen in Figure 6.32b and Figure 6.32c, respectively. From Figure 6.32f, the DACEMAD–DDQN algorithm outperforms in terms of the total system's EE, the MADDPG approach by as much as 95%. Figure 6.32e and Figure 6.32f clearly show that our DMARL can jointly maximise the total system's EE and energy utilisation by UAVs without degrading the coverage performance much better than

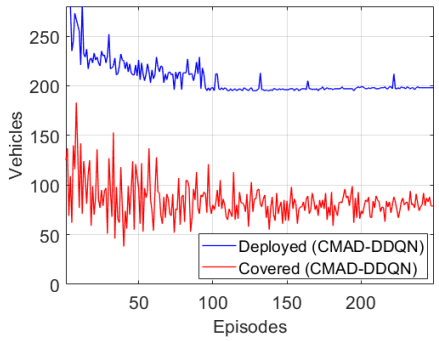
the MADDPG that neglects interference from nearby UAV cells.



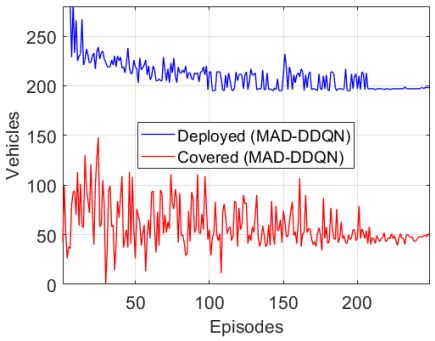
(a) Total number of road vehicles in the network vs. episodes (DACEMAD–DDQN).



(b) Total number of road vehicles in the network vs. episodes (DAMAD–DDQN).



(c) Total number of road vehicles in the network vs. episodes (CMAD–DDQN).



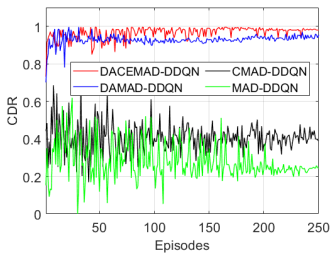
(d) Total number of road vehicles in the network vs. episodes (MAD–DDQN).

Figure 6.33: Impact of the proposed approach on the coverage behaviour in congested traffic scenario of the N7 national road, Ireland over learning episodes using 10 deployed UAVs.

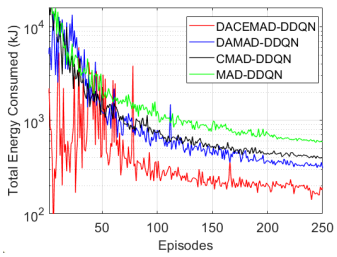
6.6.9 National Road Setting with High Concentration of Vehicles

To investigate the effectiveness of our DMARL solution in a national road setting with a high concentration of vehicles, we deploy 10 UAVs on the N7 national road under congested traffic condition, where the number of vehicles on the road is at its peak. Specifically, we consider vehicles as users and do not consider pedestrians to be deployed on the national road. As specified in Section 5.3, we consider 12769 vehicles injected into the considered N7 national road network. From Figure 6.33, we see plots of deployed vehicles and covered vehicles against the learning episodes. Figures 6.33a, 6.33b, 6.33c, 6.33d show the learning behaviour of the DMARL variants with DACEMAD–DDQN, DAMAD–DDQN, CMAD–DDQN and MAD–DDQN agents over a series of time steps, respectively. After about 200^{th} episode, we observe convergence in the number of covered vehicles with respect to the deployed vehicles in the

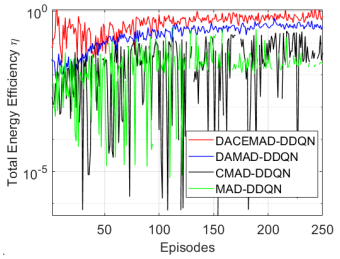
network.



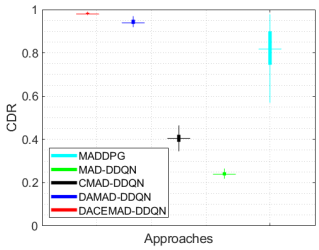
(a) CDR vs. Episodes.



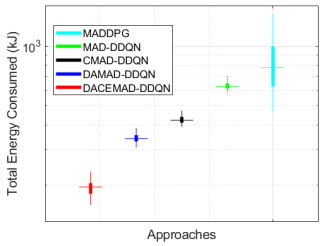
(b) Total Energy Consumed vs. Episodes.



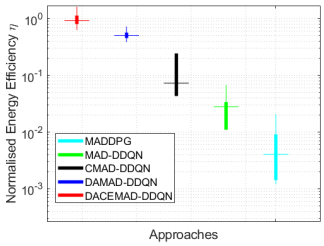
(c) Total Energy Efficiency vs. Episodes.



(d) CDR vs. Approaches.



(e) Total Energy Consumed vs. Approaches.



(f) Normalised Energy Efficiency vs. Approaches.

Figure 6.34: Comparative analysis using 10 deployed UAVs to serve vehicles along a segment of the N7 national road, Ireland under congested traffic conditions.

We evaluate the performance of our DMARL solution by comparing it with the existing baseline under congested traffic conditions along the N7 national road as seen in Figure 6.34. Figure 6.34a shows the plot of the CDR versus the learning episodes. The DACEMAD–DDQN and DAMAD–DDQN variants performed closely well in terms of improving the CDR. The density-aware variants outperform both the CMAD–DDQN and MAD–DDQN variants. We observe improvement in terms of the CDR for the DACEMAD–DDQN variant over the DAMAD–DDQN variant. Likewise, we see that the CMAD–DDQN variant outperforms the MAD–DDQN variant. Intuitively, this demonstrates that the communication-enabled mechanism allows agent-controlled UAVs to collaborate while effectively serving congested road users along the national road. However, this communication mechanism incurs higher communication costs in the network. Figure 6.34d compares the CDR performance of our DMARL against the MADDPG approach under congested traffic conditions of the M50 motorway. Under congested traffic conditions, we see that the DACEMAD–DDQN and DAMAD–DDQN algorithm outperforms the MADDPG in terms of CDR. Figure 6.34b and Figure 6.34c show the total energy consumed and the total EE versus the learning episodes,

respectively. The total energy consumed in our DMARL was significantly minimised as seen in Figure 6.34b. Figure 6.34e validates that our DMARL consistently outperforms the MADDPG baseline. From Figure 6.34f, the DACEMAD–DDQN algorithm which performed best overall approaches in terms of the total system's EE outperforms the MADDPG approach by about 98%. This performance improvement demonstrates that our DMARL solution can jointly maximise the total system's EE and energy utilisation by UAVs without degrading the coverage performance is much better than the MADDPG which neglects interference from nearby UAV cells.

6.6.10 Evaluation Summary for Collaborative Density-Aware Agents

In this section, we demonstrate that the DMARL with *Density-Aware Collaborative agents* provides an answer to the research question RQ3. We propose a *Density-Aware Direct Collaborative agents* variant and a *Density-Aware Indirect Collaborative agents* variant of the DMARL to allow for collaboration among agent-controlled UAVs to learn policies that maximise the systems' EE while providing coverage to highly mobile and densely uneven users' distribution in real-time. These variants can optimise the total systems' EE of a fleet of UAVs serving vehicles and/or pedestrians in an interference-limited environment. First, we investigate the deployment of UAVs to serve static toy users under different network configurations as seen in Figure A.1 in the Appendix. The outcome motivated us to deploy the UAVs to provide wireless coverage to highly mobile and densely uneven road users.

We considered the deployment of UAVs to serve vehicles and pedestrians in a 3 km^2 area of Dublin city centre (DCC). We then evaluated the performance under 3 traffic conditions as highlighted earlier in the evaluation scenario. Under all 3 traffic conditions in the urban setting, the MADDPG outperforms our DMARL approach in terms of CDR. Nevertheless, this performance gain comes at a high energy cost. We observed significantly better performance in the *Density-Aware Collaborative agents* variants as compared to the *Collaborative agents* variants, which have no density-aware mechanism, in terms of CDR. Likewise, the total energy consumed by the UAVs in the *Density-Aware Collaborative agents* variants was significantly minimised as compared to the other approaches. The MADDPG performed

worse in terms of the total energy consumed and the total system's EE. The *Density-Aware Direct Collaborative agents* variant outperformed the *Density-Aware Indirect Collaborative agents* variant, that has no direct communication mechanism, in terms of the total energy consumed and the total system's EE. This goes to show the importance of communication in enhancing UAVs' collaboration. Contrary to our expectation that communication among the UAVs should improve the system performance by minimising the total energy consumed, we observed that the *Indirect Collaborative agents* variant consumed less energy than the *Direct Collaborative agents* variant in most traffic conditions. However, this gain was at a cost of poorer coverage performance. As expected, the *Density-Aware Collaborative agents* variants jointly improved the total energy usage and the total system's EE without degrading the coverage performance in the DCC road network. This shows the relevance of having a density-aware feature in addition to a collaborative mechanism, especially in urban road networks with highly mobile user distribution. Overall, our DMARL with *Density-Aware Collaborative agents* solution outperforms the MADDPG approach and effectively answers the research question RQ3.

We go further in our evaluation and considered the deployment of UAVs to serve vehicles along a 7 km segment of the M50 motorway. Unlike the DCC scenario, the M50 motorway presents highly uneven user distribution. We evaluated the system performance under the 3 traffic conditions. In low, moderate and high concentrations of deployed vehicles, we observed that our *Density-Aware Collaborative agents* variants outperform the baseline approaches in terms of CDR. This shows that irrespective of the traffic conditions our approach is robust enough to provide coverage to highly mobile and uneven users' distribution. We also observed that communication played a significant role in optimising the total energy consumed and the total system's EE. Interestingly, the MADDPG approach slightly outperformed the *Collaborative agents* variants in terms of CDR, however, performed worse in terms of the total energy consumed and the total system's EE. Most notably, our *Density-Aware Collaborative agents* variants outperformed the baseline approach under different traffic conditions and effectively answers the research question RQ3.

Lastly, we considered the deployment of UAVs to serve vehicles along a 6.5 km segment of the N7 national road. Like the M50 motorway, the N7 national road is also characterised by highly uneven user distribution, though with a lesser traffic volume as compared to the M50 motorway. In low, moderate and high concentrations of deployed vehicles, we observed that our *Density-Aware Direct Collaborative agents* variant outperforms the baseline approaches in terms of CDR. Nevertheless, the MADDPG approach closely matched the average value of the *Density-Aware Direct Collaborative agents* in terms of CDR under low traffic conditions. The *Direct Collaborative agents* variant also outperformed the *Density-Aware Indirect Collaborative agents* variant under low traffic conditions in terms of CDR. This reveals that communication has some significant impact on improving the coverage performance in the network. The *Indirect Collaborative agents* performed worse in terms of improving the coverage performance under all traffic conditions. Under moderate and high traffic conditions, we observe that the MADDPG approach outperforms the *Collaborative agents* variants but not the *Density-Aware Collaborative agents* variants. As expected, our DMARL approach outperformed the MADDPG baseline in terms of the total energy consumed in all 3 traffic conditions. Interestingly, the MADDPG baseline outperformed the *Collaborative agents* variants in terms of the total system's EE in low traffic conditions, and closely matches in moderate traffic conditions. In high-traffic conditions, we observed improved performance in our DMARL solution over the MADDPG approach. As expected, our *Density-Aware Collaborative agents* variants outperformed the baseline approach under different traffic conditions and again effectively answers the research question RQ3.

Our *Density-Aware Collaborative agents* variants guarantee quick adaptability and convergence in a shared and dynamic network environment. We compared the effectiveness of the DMARL with *Density-Aware Collaborative agents* with state-of-the-art decentralised MARL approaches under the same network conditions. The results consistently show that the *Density-Aware Collaborative agents* variants jointly maximise the total system's EE and energy utilisation by UAVs without degrading the coverage performance in real-life road networks.

Table 6.1: Summary of Results Addressing our Research Questions

Addressing RQ1	**Normalised EE – DMARL vs. MARL Baseline (Synthetic Data)**					
	DQLSI	MAD–DDQN	CMAD–DDQN	DAMAD–DDQN	DACEMAD–DDQN	CQL
Static	100%	–	–	–	–	64%
Dynamic Even Distributed	100%	–	–	–	–	19%
Dynamic Uneven Distributed	100%	–	–	–	–	57%
Addressing RQ2	**Normalised EE – DMARL vs. MARL Baseline (Synthetic + Real-World Data)**					
	DQLSI	MAD–DDQN	CMAD–DDQN	DAMAD–DDQN	DACEMAD–DDQN	MADDPG
2 UAVs Deployment	–	100%	87%	–	–	58%
4 UAVs Deployment	–	100%	88%	–	–	43%
6 UAVs Deployment	–	100%	99%	–	–	27%
8 UAVs Deployment	–	92%	100%	–	–	41%
10 UAVs Deployment	–	97%	100%	–	–	41%
12 UAVs Deployment	–	89%	100%	–	–	38%
Addressing RQ3	**Normalised EE – DMARL vs. MARL Baseline (Real-World Data)**					
	DQLSI	MAD–DDQN	CMAD–DDQN	DAMAD–DDQN	DACEMAD–DDQN	MADDPG
DCC - Low traffic	–	44%	39%	84%	100%	8%
DCC - Moderate traffic	–	45%	58%	90%	100%	6%
DCC - Congested traffic	–	38%	36%	100%	97%	5%
M50 - Low traffic	–	24%	62%	54%	100%	2%
M50 - Moderate traffic	–	42%	43%	73%	100%	4%
M50 - Congested traffic	–	14%	20%	59%	100%	2%
N7 - Low traffic	–	2%	0%	6%	100%	6%
N7 - Moderate traffic	–	0%	2%	42%	100%	3%
N7 - Congested traffic	–	3%	7%	83%	100%	2%

6.7 Evaluation Summary

In this chapter, we presented details of the evaluation of DMARL for UAV-assisted networks. We presented the evaluation objectives, metrics and baselines, as well as the evaluation scenarios, and then presented and analysed the results. We provide a summary of our results in Table 6.1 where we explicitly addressed our 3 research questions by demonstrating that our proposed DMARL solution can significantly improve the total EE of UAVs deployed to serve ground users. The table shows the evaluation of our proposed DMARL approach against the closest MARL approaches under different use-cases and scenarios. From the analysis of the results, we conclude that DMARL is a suitable algorithm for multi-UAV deployment in emergency situations where there is a service outage due to failure in existing terrestrial infrastructure or central controller. We investigated the performance of the DMARL variants to address our research questions under three categories: DMARL with *Independent Learning agents*, DMARL with *Collaborative agents*, and DMARL with *Density-Aware Collaborative agents*.

DMARL with *Independent Learning agents* outperforms existing centralised baselines that rely on a CC for decision-making in terms of EE by as much as 80%. In the static settings with randomly distributed immobile ground users, the centralised baselines performed well in improving the coverage performance. This is because the centralised controllers are able to

master the locations of the ground users to improve the coverage performance in the network. However, in all dynamic settings, we observe a significant drop in the number of connected users by the centralised baselines. Results show that our DMARL with *Independent Learning agents* is capable of effectively serving mobile ground users and suitable to serve mobile pedestrians in a given area. Importantly, our DMARL with *Independent Learning agents* significantly improves the total system's energy efficiency of the agent-controlled UAVs in the network. In this chapter, we demonstrated that the DMARL with *Independent Learning agents* answers the research question RQ1.

The DMARL with *Collaborative agents* supports collaborative behaviours among agent-controlled UAVs in a shared, dynamic and interference-limited environment. This approach is very suitable when the number of UAVs in the network is increased. Specifically, the dynamic and interference-limited environment may induce some selfish tendencies among the UAVs, thus making it crucial for UAVs to collaborate. We achieved collaboration via the *Direct Collaborative agents* variant which allows UAVs to share their telemetry via existing 3GPP guidelines, and the *Indirect Collaborative agents* variant that has no such mechanism but implicitly reflects this knowledge in its reward formulation as an incentive towards collaborative behaviours. The DMARL with *Collaborative agents* outperforms the multi-agent deep deterministic policy gradient (MADDPG) approach that ignores the impact of interference from nearby UAV cells in terms of total system EE by as much as 55%–75%. In this chapter, we demonstrated that the DMARL with *Collaborative agents* answers the research question RQ2.

DMARL with *Density-Aware Collaborative agents* is suitable for deployment in highly mobile and densely uneven users' distribution. The DMARL outperforms the existing multi-agent deep deterministic policy gradient approach that neglects the impact of interference from nearby UAV cells in terms of total system EE by as much as 65%–98%. We investigated the effectiveness of our DMARL under different urban traffic scenarios and conditions. We observed that the *Density-Aware Collaborative agents* variants consistently outperformed *Collaborative agents* variants in terms of maximising the total system's EE by jointly optimising

the UAVs' flight trajectory, the number of connected users, and the total energy consumed by the UAVs in different road networks and traffic conditions. While *Collaborative agents* variants performed well with evenly distributed pedestrians confined in a given coverage area, the approach may not be as suitable to be deployed in road networks with highly mobile and densely uneven users' distribution. In this chapter, we demonstrated that the DMARL with *Density-Aware Collaborative agents* answers the research question RQ3.

In conclusion, our DMARL approach is robust enough to provide UAVs deployed in an environment with the intelligence to provide coverage in an energy-efficient manner. In the next chapter, we present the conclusion to the thesis and discuss open research issues.

Chapter 7

Conclusion

In this chapter, we summarise the thesis and highlight our findings. We then discuss open research issues that pertain to this work.

7.1 Thesis Contribution

This thesis proposes a decentralised multi-agent reinforcement learning (DMARL) solution for UAV-assisted networks. The main aim of this thesis is to maximise the total system's energy efficiency (EE) while optimising the UAVs' flight trajectories, the number of connected users, and the energy consumed by UAVs in a shared, dynamic and interference-limited environment. In Chapter 1, we motivated this problem. Specifically, this work focuses on emergency scenarios, where multiple UAVs are deployed to provide wireless connectivity to ground users, during which there is service downtime due to failure in existing terrestrial infrastructures or centralised controllers. We then examined several challenges, such as difficulty in serving users due lack of apriori knowledge of the locations of ground users, dynamic changes in the network due to mobility of users, performance degradation due to interference from nearby UAV cells, conservation of UAVs energy during prolonged flight, and the ability for UAVs to effectively collaborate in shared, dynamic and interference-limited environments. We go further to extract pertinent research questions from identified gaps in existing work. We decomposed our overarching research question (**RQ**), "Can UAVs deployed to provide

wireless connectivity to mobile ground users improve the total system's energy efficiency in a shared, dynamic and interference-limited network environment?", into 3 **RQs** to specifically address the research gaps. We formulated our first research question (**RQ1**) as, "Can UAVs serving mobile ground users improve the total system's energy efficiency in a shared, dynamic and interference-limited network environment without relying on a central controller for decision-making?", the second research question (**RQ2**) as, "Can collaboration with closest neighbours improve the total system's energy efficiency while minimising the total energy consumed by UAVs in a shared, dynamic and interference-limited network environment?", and our third research question (**RQ3**) as, "Can UAVs collaborate intelligently to improve the total system's energy efficiency in highly mobile and densely uneven users' distribution in an urban environment?". We then presented our research contributions to proffer answers to the research questions raised. We then provided the outline for the thesis.

In Chapter 2, we present a state-of-the-art review of Reinforcement Learning (RL) in UAV-assisted networks. We discussed concepts of RL, in order to provide our readers with the needed background to understand the DMARL approach. We introduced the tabular Q–Learning (QL) and the Double Deep Q–Network (DDQN) algorithms which are used later on in our design in Chapter 4. We go further to discuss the Deep Deterministic Policy Gradient (DDPG) algorithm which was used by our closest evaluation baseline. We then introduced the Multi-Agent Systems and discuss in detail the Multi-Agent Reinforcement Learning (MARL). We highlighted some of the challenges faced in MARL environments in light of recent contributions in the field. We then discussed the motivation for collaboration among multiple agents in a shared environment. We understood that collaboration can be achieved through strategic mechanisms, such as, reward assignment and communication.

To allow our readers to have some insight into research development in the area of UAV-assisted networks, we discuss the applications of UAVs as aerial base stations, relays, and data sinks/disseminators. We dive deeper to discuss our specific use case scenario, highlighting the importance of deploying multiple UAVs as base stations in disaster scenarios. We got insights into the challenges and gaps in this area of study, such as, the need for UAVs to be fully autonomous and capable of intelligent decision-making while providing wireless

connectivity to the ground users. We provide a summary of the challenges faced in deploying multiple UAVs as aerial base stations in a shared, dynamic and interference-limited environment like ours. With growing research interest towards agent-based control in UAV-assisted networks, we were able to classify related works into centrally controlled where a central controller carries out the decision-making operation and a decentralised control that involves the UAVs managing the decision-making process locally. We then provided a summary of Chapter 2, highlighting our scope on the deployment of multiple rotary-wing UAVs serving as an aerial base station to serve ground users in emergencies, where there is a service outage due to failure in existing cellular infrastructure or increased service demand on limited available infrastructure.

In Chapter 3, we presented the multi-UAV system model design. First, we presented a brief overview of the deployment scenario of multiple UAVs deployed to provide wireless service to ground users due to service unavailability in existing terrestrial infrastructure resulting from possible disaster, unforeseen load or failure in parts of the network. We then presented the wireless channel model, highlighting our assumption for guaranteed Line-of-Sight conditions due to the aerial positions of the UAVs. Considering that frequency spectrum is a scarce radio resource, as such we anticipate that most cellular providers may have to reuse this frequency resource, implying that UAVs may have to share the same radio frequency. However, sharing the same frequency spectrum introduces interference from nearby UAVs or APs. Therefore, our wireless channel model takes into account the interference from nearby UAV small cells. We apply Shannon's equation to compute the receiving data rate at the user. We then presented our connectivity model which allows us to compute the number of connected users by each UAV. To ensure that all users are fairly connected to available UAVs, we apply Jain's fairness index. We presented three mathematical-based mobility models widely used in ad-hoc networks literature to depict the mobility of ground users, especially pedestrians. These models were useful in Chapter 6 to evaluate the performance of our proposed DMARL. Next, we presented the energy consumption model used, and go further to provide an expression for the total system's EE. We then formulate our problem with an objective to maximise the total system's EE by jointly optimising each UAV's trajectory, number of connected users,

and the energy consumed by the UAVs under a strict energy budget.

In Chapter 4, we propose a DMARL solution for UAV-assisted networks that allows each UAV equipped with an autonomous agent to intelligently serve ground users while improving the total system's EE in a shared, dynamic and interference-limited network environment. We presented the requirements for DMARL in order for UAVs to provide ubiquitous coverage to ground users in a shared, dynamic and interference-limited network environment. Next, we derived the design of our proposed DMARL solution using the requirements (Section 4.1). To effectively answer our overarching research question (**RQ**), we decompose our DMARL design into five variants. The variants include the *Independent Learning agent*, the *Indirect Collaborative agent*, the *Direct Collaborative agent*, the *Density-Aware Indirect Collaborative agent*, and the *Density-Aware Direct Collaborative agent*. The *Independent Learning agent* through our proposed Decentralised Q-Learning with Local Sensory Information (DQLSI) algorithm is designed to answer our first research question (**RQ1**).

The *Direct Collaborative agent* and *Indirect Collaborative agent* through our proposed Multi-Agent Decentralised Double Deep Q-Network (MAD–DDQN) and Communication-enabled Multi-Agent Decentralised Double Deep Q-Network (CMAD–DDQN), respectively, are designed to answer our second research question (**RQ2**). To answer our third research question (**RQ3**), the *Density-Aware Indirect Collaborative agent* and *Density-Aware Direct Collaborative agent* variants through the Density-Aware MAD–DDQN (DAMAD–DDQN) and Density-Aware CMAD–DDQN (DACEMAD–DDQN) algorithms, respectively, are designed. We presented the complexity analysis of our proposed DMARL for UAV-assisted networks. First, we present the time complexity of the DQLSI algorithm. Next, we present the time complexity of the MAD–DDQN, CMAD–DDQN, DAMAD–DDQN, and DACEMAD–DDQN algorithms. We then provide a summary of our design contributions.

In Chapter 5, we present the implementation of the DMARL for UAV-Assisted Networks. We presented the class diagram of our DMARL solution. The training phase for the DMARL approach is presented. Here, we discuss in detail the procedure taken to train the agents. We then discuss the deployment of the agent-controlled UAVs and the ground users in the environment. In our UAVs' deployment, we consider that UAVs may interact with neighbouring

UAVs. Each UAV is capable of optimising its trajectory while hovering and providing wireless connectivity. We consider the deployment of ground users, which can be static or mobile. In our *Independent Learning agent* variant, we consider both static and mobile settings and model the mobile users to follow some mathematical-based mobility model. We again deploy both static and mobile ground users in our *Collaborative agent* variants. For our *Density-Aware Collaborative agent* variants, we use real-world traffic data from SUMO to provide realistic deployments of ground users in an urban environment. We presented the different road traffic networks and traffic conditions that were considered in our implementation. We then provided a summary of the implementation.

In Chapter 6, we evaluated the performance of our proposed DMARL solution and investigate its effectiveness in answering the research questions specified in Chapter 1. We present the evaluation objectives. Here, we aim to observe whether results hold for different ground users types (pedestrians, vehicles), different ground users' deployment settings and distribution (static/mobile, even/uneven), different UAVs configuration (varying number of agent-controlled UAVs), different mobility models (mathematical-based, SUMO-generated), different traffic network conditions (low, saturated, congested), and different road networks (city roads, motorway, national road). The performance metrics and the baselines considered were presented. Next, we presented the scenarios considered in evaluating our proposed DMARL for UAV-assisted networks. To evaluate the robustness of our solution, the DMARL agents were trained, and execution was carried out in the environment in parallel with other operating agents.

We deployed the *Independent Learning agents* and measured the performance using our evaluation metrics. To ensure the effectiveness of the *Independent Learning agents* algorithm in addressing **RQ1**, we compared with centralised baselines that assume global knowledge with insights gotten from a CC. We observed that the *Independent Learning agents* can jointly maximise the number of connected ground users and the energy utilisation of the UAVs while improving the total system's energy efficiency without relying on a CC. The *Independent Learning agents* outperformed centralised approaches in terms of the total system's EE by as much as 80% over all settings considered. This solution effectively answered the **RQ1**.

This thesis provides insights that could help in the deployment of multiple UAVs serving as aerial base stations. In particular, We gained insights to the decentralised deployment of the agent-controlled UAVs, and conclude that the UAVs can provide coverage without necessarily relying on a CC for local decision making. Interestingly, the centralised approaches outperformed the DMARL solution in improving the number of connected ground users. However, our proposed DMARL solution outperforms the centralised approaches in improving the total EE of the UAVs. We see that our DMARL approach is particularly suitable in disaster scenarios where a possible failure in the CC may occur, thereby, affecting the decision making of the UAVs. Addressing the first research question provides sufficient backing that UAVs can effectively serve ground users in a decentralised manner and without the reliance on a central entity.

We then deployed the *Collaborative agents* and measured the performance using the metrics. We observed that the *Collaborative agents* can effectively collaborate to maximise the total system's EE while jointly optimising the UAVs' flight trajectory, the number of connected users and the energy consumption in a shared, dynamic and interference-limited environment. Although the *Direct Collaborative agents* exhibited slightly better performance over the *Indirect Collaborative agents* in most cases, this performance improvement comes at an increased communication cost. Overall, our DMARL solution outperformed the closest evaluation baseline, the Multi-Agent Deep Deterministic Policy Gradient (MADDPG) approach and the random policy in terms of EE by as much as 55% – 75%. This solution effectively answered the **RQ2**. Comparing our *Collaborative agents* variants with the MADDPG provided us with an opportunity to evaluate how well our DMARL solution improves the overall system performance. Despite the MADDPG outperforming our DMARL solution in terms of improving the number of connected users, the MADDPG approach was not as energy efficient as our DMARL solution. Through experimentation, we can categorically come to the conclusion that our proposed DMARL solution will be suitable in energy-constrained UAVs base station applications. Addressing the second research question clearly reveals that UAVs can collaborate to improve the total EE of the UAVs without degrading the coverage performance in the network.

We deployed the *Density-Aware Collaborative agents* and measured the performance using our evaluation metrics. We observed that the *Density-Aware Collaborative agents* can effectively serve dense and uneven users' distribution while maximising the total system's EE by jointly optimising the UAVs' flight trajectory, the number of connected users and the energy consumption in a shared, dynamic and interference-limited environment. Although the *Density-Aware Direct Collaborative agents* outperformed the *Density-Aware Indirect Collaborative agents* in most cases, this performance gain comes with increased communication overhead. We investigated the effectiveness of our proposed DMARL solution and observed that the *Density-Aware Collaborative agents* variants outperform their counterparts that do not have any mechanism of keeping track of dense users' locations, i.e., the *Collaborative agents* variants. Furthermore, we compare the DMARL solution against our closest evaluation baseline, the Multi-Agent Deep Deterministic Policy Gradient (MADDPG) approach. Although the MADDPG approach outperformed our DMARL approach in terms of improving coverage, our DMARL solution performed better than the MADDPG in terms of EE by as much as 65%–98%. This solution effectively answered the **RQ3**. Providing coverage to highly mobile and unevenly distributed ground users comes with its challenges since UAVs must intelligently keep track of dense users' location in this dynamic environment. Our *Density-Aware Collaborative agents* variants may not guarantee total coverage due to the non-stationarity induced by the interacting agents and that from the environment. Nevertheless, our DMARL solution significantly improves the total energy efficiency of the UAVs without degrading the coverage performance in the network. Addressing the third research question clearly demonstrates the effectiveness of our proposed DMARL approach in providing the intelligence for the UAVs to serve the ground users under realistic road traffic conditions.

Our proposed solution can be directly applied to real-world settings only if certain conditions are met, including reliable communication between agents and full observability of agent-controlled UAVs. With our assumptions of perfect channel conditions among interacting UAVs, and modeling our agents to have full local observability, we demonstrated the ef-

fectiveness of our DMARL solution under certain road traffic conditions. Our decentralised approach which supports autonomous control of UAVs may suffice in disaster scenarios where manual or centralised control mechanisms may be unfeasible. However, delayed or lossy communication which impacts the overall performance of some wireless communication networks may be a bottleneck when our proposed solution is deployed in real-world settings. Nevertheless, we investigated the effectiveness of our DMARL solution under certain dynamic settings where the set of neighbouring agent-controlled UAVs and connected ground users change over time. In such dynamic scenarios where quick and timely decisions need to be made, our DMARL solution offers such capabilities, and allows the agent-controlled UAVs to make more informed decision in real-time while improving the overall performance in the network. Crucially, UAV-assisted networks are energy-constrained, and as such, may require intelligent strategies to reduce the energy cost while improving the total energy efficiency of the UAVs. Throughout the thesis, we demonstrated the capability of our approach to address this challenge. Our solution is expected to yield good performance when deployed in real-world settings under certain conditions. Overall, we conclude that our DMARL solution is robust enough to provide UAVs deployed in urban environments with the intelligence to provide wireless coverage to ground users in an energy-efficient manner.

7.2 Limitations and Future Work

This thesis shows that our proposed DMARL for UAV-assisted networks solution can improve the overall system's EE while optimising the UAVs' flight trajectories, number of connected users, and the energy consumed by UAVs under a strict energy budget. Although it outperforms existing approaches, it has some limitations.

1. **Investigating the downside of delayed or lossy communication:** In circumstances where UAVs communicate with neighbours, we do not consider delayed or lossy communication, which we understand may be a source of additional complexity. The environment may not be without its obstacles. Therefore, taking into account channel

impairments such as, shadowing[1] and fading[2] effects of wireless communication channels is fundamental to the efficient design of ultra-reliable low latency wireless networks. We understand that these channel impairments may lead to loss of packets and increased delays in the network. In particular, delay and possible loss of packets may impact on the learning process of the agent-controlled UAVs. To account for delay and possible loss in packets, the system model is expected to capture Non-line-of-sight (NLoS)[3] links. we need to We hope to account for this in our future works.

2. **Impact of heterogeneous agents in the network:** In this work, we only consider a set of homogeneous agent-controlled UAVs. With the growth of AI technologies, we may see deployments of different RL agents on UAVs. In particular, different autonomous agents may control UAVs with peculiar use-case application, for example, agent-controlled UAV small cells may be deployed alongside agent-controlled UAV relays with different underlying technologies. It is crucial to have a common standard that allows for seamless operability among heterogeneous agents having unique goals. Neglecting the impact of nearby agents operating in the shared heterogeneous environment may be detrimental to the overall system performance. We understand that collaboration among heterogeneous agents can be achieved through communication. Interoperability among agents is crucial for collaboration and achieving common goals. Hence, we envisage that these agents may require some communication mechanism that allow them work together a shared network environment.

3. **Partial Observability:** In this thesis, we model each agent to have full local observability. Hence, we do not use POMDP approaches to solve this problem. POMDPs extend MDPs to environments where the intentions of other agents cannot be directly observed and are often encoded in hidden variables. POMDPs can also be used to model decision-making and collaboration among multiple agents in decentralised partially ob-

[1] An effect that causes fluctuation in the received signal power due to the presence of obstacles obstructing the propagation path between transmitter and receiver.

[2] A variation of the attenuation of a signal with various variables such as, time, geographical position, and radio frequency.

[3] NLoS links refer to radio propagation that occurs outside of the typical line-of-sight (LoS) between the transmitter and receiver, such as in ground reflections or partial obstruction by physical objects present.

servable settings. The merit of this approach is that it only requires to consider belief states that are reachable from the current belief state. In agent-controlled UAV-assisted networks, the assumption of full local observability may not always hold. Hence, our future work will explore such solutions.

4. **Real Environment Implementation:** This thesis evaluates the DMARL for UAV-assisted network in simulated environments. This is due to the high cost and risk of deploying real world UAVs in the environment. Deploying real UAVs may be expensive for experimentation, however, it is crucial to be able to test the performance of the system on real world deployments. Flying and operating UAVs in the Republic of Ireland is subject to European Union Regulation 2019/947, with the supervision of the Irish Aviation Authority (IAA). The IAA also provides guidance via regulations [4] regarding the deployment of UAVs in order to ensure public safety. In line with the regulations, our future work will focus on testing our DMARL solution on real world UAV-assisted networks. More importantly, this deployment should provide immense support in emergencies, especially where there is failure in existing terrestrial infrastructure or service outage due to increased network load.

[4] https://www.iaa.ie/general-aviation/drones

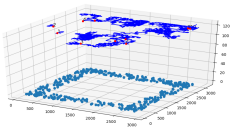
Appendix A

Appendix

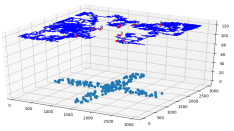
A.1 Investigating Density-Aware Collaborative Variant on Toy Scenarios

We consider 3 different network configurations with 10 UAVs serving static ground users. The objective is to verify the effectiveness of the Density-Aware Direct Collaborative agent approach in providing coverage to ground users with different spatial distributions. Figures A.1a – A.1c show different distributions of static ground users served by 10 UAVs and their trajectories over a series of time steps. Figure A.1a shows scenario 1, where we deployed a set of static ground users circularly. We can see the trajectories of the 10 UAVs in blue and in each episode, the UAVs are assigned a random take-off point. The red dot indicated the present location of the UAVs. Figure A.1b shows scenario 2, where we deployed a set of static ground users in a crossroad intersection manner. Figure A.1c shows scenario 3, where we deployed a set of static ground users in an edge-like distribution. Figures A.1a – A.1c show the trajectory as UAVs learn in the 10^{th} learning episode. As expected, we observe a high degree of exploration by the UAVs, leading to random and uncertain policies. However, during the 250^{th} episode, the UAVs' actions are more definite and from Figures A.1d – A.1i, we see that the UAVs are aware of the dense user locations, seeking to move towards those regions. This behaviour was also observed when testing subsequently trained agents.

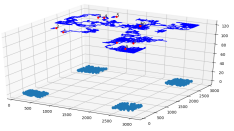
Figures A.1j – A.1l show the plots of CDR and total EE against the learning episodes on the different toy scenarios. Despite different learning behaviours across the considered scenarios, we observe convergence after the 200^{th} episode. The results show that the UAVs are capable of collaborating amongst themselves to improve the CDR and the total EE in a static setting. Nevertheless, our interest is to investigate the effectiveness of our DMARL solution in serving highly mobile and densely uneven users.



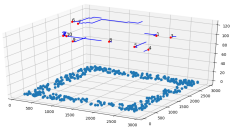
(a) Simulation scenario 1 at 10^{th} episode.



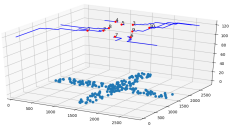
(b) Simulation scenario 2 at 10^{th} episode.



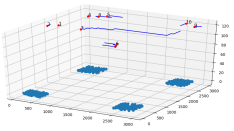
(c) Simulation scenario 3 at 10^{th} episode.



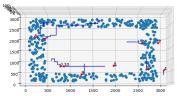
(d) Simulation scenario 1 at 250^{th} episode.



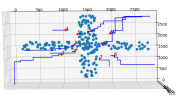
(e) Simulation scenario 2 at 250^{th} episode.



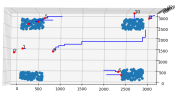
(f) Simulation scenario 3 at 250^{th} episode.



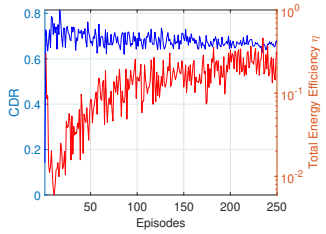
(g) Top view of scenario 1 at 250^{th} episode.



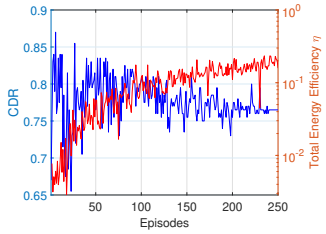
(h) Top view of scenario 2 at 250^{th} episode.



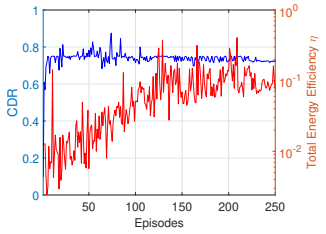
(i) Top view of scenario 3 at 250^{th} episode.



(j) Scenario 1's connected users to deployed users ratio (CDR) vs. episodes.



(k) Scenario 2's connected users to deployed users ratio (CDR) vs. episodes.



(l) Scenario 3's connected users to deployed users ratio (CDR) vs. episodes.

Figure A.1: Pre-trials of the DACEMAD-DDQN with flight directory of 10 UAVs deployed to provide coverage to static toy-case users in different density scenarios

Bibliography

[3GPP, 2008] 3GPP (2008). Automatic Neighbour Relation Management. https://https://www.3gpp.org/ftp/Specs/archive/32_series/32.511. Accessed: 2021-07-05.

[3GPP, 2019] 3GPP (2019). Enhancement for Unmaned Aerial Vehicles (UAVs). https://portal.3gpp.org/desktopmodules/Specifications/\SpecificationDetails.aspx?specificationId=3557. Accessed: 2021-07-05.

[Amato et al., 2013] Amato, C., Chowdhary, G., Geramifard, A., Üre, N. K., and Kochenderfer, M. J. (2013). Decentralized control of partially observable markov decision processes. In *52nd IEEE Conference on Decision and Control*, pages 2398–2405.

[Azari et al., 2018] Azari, M. M., Rosas, F., Chen, K.-C., and Pollin, S. (2018). Ultra reliable uav communication using altitude and cooperation diversity. *IEEE Transactions on Communications*, 66(1):330–344.

[Bayerlein et al., 2021] Bayerlein, H., Theile, M., Caccamo, M., and Gesbert, D. (2021). Multi-UAV path planning for wireless data harvesting with deep reinforcement learning. *IEEE Open Journal of the Communications Society*, 2:1171–1187.

[Becker et al., 2004] Becker, R., Zilberstein, S., and Lesser, V. (2004). Decentralized markov decision processes with event-driven interactions. In *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems, 2004. AAMAS 2004.*, pages 302–309.

[Betalo et al., 2022] Betalo, M. L., Leng, S., Zhou, L., and Fakirah, M. (2022). Multi-uav data collection optimization for sink node and trajectory planning in wsn. In *2022*

IEEE 2nd International Conference on Computer Communication and Artificial Intelligence (CCAI), pages 1–7.

[Biomo et al., 2014] Biomo, J.-D. M. M., Kunz, T., and St-Hilaire, M. (2014). An enhanced gauss-markov mobility model for simulations of unmanned aerial ad hoc networks. In *2014 7th IFIP Wireless and Mobile Networking Conference (WMNC)*, pages 1–8.

[Bouk et al., 2015] Bouk, S. H., Ahmed, S. H., Omoniwa, B., and Kim, D. (2015). Outage minimization using bivious relaying scheme in vehicular delay tolerant networks. *Wirel. Pers. Commun.*, 84(4):2679–2692.

[Boutilier, 1999] Boutilier, C. (1999). Sequential optimality and coordination in multiagent systems. In *Proceedings of the 16th International Joint Conference on Artifical Intelligence - Volume 1*, IJCAI'99, page 478–485, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

[Busoniu et al., 2006] Busoniu, L., Babuska, R., and De Schutter, B. (2006). Multi-agent reinforcement learning: A survey. In *2006 9th International Conference on Control, Automation, Robotics and Vision*, pages 1–6.

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

[Camp et al., 2002] Camp, T., Boleng, J., and Davies, V. (2002). A survey of mobility models for ad hoc network research. *Wireless Communications and Mobile Computing*, 2(5):483–502.

[Camps-Mur et al., 2021] Camps-Mur, D., Gavras, A., Ghoraishi, M., Hrasnica, H., Kaloxylos, A., Anastasopoulos, M., Tzanakaki, A., Srinivasan, G., Antevski, K., Baranda, J., Schepper, K., Casetti, C., Chiasserini, C., Garcia-Saavedra, A., Guimares, C., Kondepu, K., Li, X., Magoula, L., Malinverno, M., and Cogalan, T. (2021). Ai and ml – enablers for beyond 5g networks.

[Canese et al., 2021] Canese, L., Cardarilli, G. C., Di Nunzio, L., Fazzolari, R., Giardino, D., Re, M., and Spanò, S. (2021). Multi-agent reinforcement learning: A review of challenges and applications. *Applied Sciences*, 11(11).

[Challita et al., 2019] Challita, U., Saad, W., and Bettstetter, C. (2019). Interference management for cellular-connected uavs: A deep reinforcement learning approach. *IEEE Transactions on Wireless Communications*, 18(4):2125–2140.

[Chen et al., 2022] Chen, B., Liu, D., and Hanzo, L. (2022). Decentralized trajectory and power control based on multi-agent deep reinforcement learning in uav networks. In *ICC 2022 - IEEE International Conference on Communications*, pages 3983–3988.

[Chen et al., 2018a] Chen, Y., Liu, X., Zhao, N., and Ding, Z. (2018a). Using multiple uavs as relays for reliable communications. In *2018 IEEE 87th Vehicular Technology Conference (VTC Spring)*, pages 1–5.

[Chen et al., 2018b] Chen, Y., Zhao, N., Ding, Z., and Alouini, M.-S. (2018b). Multiple uavs as relays: Multi-hop single link versus multiple dual-hop links. *IEEE Transactions on Wireless Communications*, 17(9):6348–6359.

[Cicek et al., 2019] Cicek, C. T., Gultekin, H., Tavli, B., and Yanikomeroglu, H. (2019). Uav base station location optimization for next generation wireless networks: Overview and future research directions. In *2019 1st International Conference on Unmanned Vehicle Systems-Oman (UVS)*, pages 1–6.

[Cicek et al., 2020] Cicek, C. T., Gultekin, H., Tavli, B., and Yanikomeroglu, H. (2020). Backhaul-aware optimization of uav base station location and bandwidth allocation for profit maximization. *IEEE Access*, 8:154573–154588.

[Cisco, 2018] Cisco (2018). Cisco Annual Internet Report (2018–2023). `https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.pdf`. Accessed: 2022-10-20.

[Claus and Boutilier, 1998] Claus, C. and Boutilier, C. (1998). The dynamics of reinforcement learning in collaborative multiagent systems. In *Proceedings of the Fifteenth Na-*

tional/Tenth Conference on Artificial Intelligence/Innovative Applications of Artificial Intelligence, AAAI '98/IAAI '98, page 746–752, USA. American Association for Artificial Intelligence.

[Cui et al., 2020] Cui, J., Liu, Y., and Nallanathan, A. (2020). Multi-agent reinforcement learning-based resource allocation for uav networks. *IEEE Transactions on Wireless Communications*, 19(2):729–743.

[Dafoe et al., 2020] Dafoe, A., Hughes, E., Bachrach, Y., Collins, T., McKee, K. R., Leibo, J. Z., Larson, K., and Graepel, T. (2020). Open problems in collaborative ai.

[Demir et al., 2020] Demir, U., Toker, C., and Ekici, O. (2020). Energy-efficient deployment of uav in v2x network considering latency and backhaul issues. In *2020 IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom)*, pages 1–6.

[Devlin et al., 2014] Devlin, S., Yliniemi, L., Kudenko, D., and Tumer, K. (2014). Potential-based difference rewards for multiagent reinforcement learning. In *Proceedings of the 2014 International Conference on Autonomous Agents and Multi-Agent Systems*, AAMAS '14, page 165–172, Richland, SC. International Foundation for Autonomous Agents and Multiagent Systems.

[Dorri et al., 2018] Dorri, A., Kanhere, S. S., and Jurdak, R. (2018). Multi-agent systems: A survey. *IEEE Access*, 6:28573–28593.

[Dublin, 2021] Dublin, S. (2021). Dockland Bins Data. `https://data.smartdublin.ie/dataset`. Accessed: 2021-10-17.

[Dusparic, 2010] Dusparic, I. (2010). *Multi-policy optimization in decentralized autonomic systems*. PhD thesis, School of Computer Science & Statistics, Trinity College (Dublin, Ireland).

[Dusparic et al., 2015] Dusparic, I., Taylor, A., Marinescu, A., Cahill, V., and Clarke, S. (2015). Maximizing renewable energy use with decentralized residential demand response. In *2015 IEEE First International Smart Cities Conference (ISC2)*, pages 1–6.

[Eom et al., 2020] Eom, S., Lee, H., Park, J., and Lee, I. (2020). Uav-aided wireless communication designs with propulsion energy limitations. *IEEE Transactions on Vehicular Technology*, 69(1):651–662.

[Foerster et al., 2017] Foerster, J., Nardelli, N., Farquhar, G., Afouras, T., Torr, P. H. S., Kohli, P., and Whiteson, S. (2017). Stabilising experience replay for deep multi-agent reinforcement learning.

[Foerster et al., 2016] Foerster, J. N., Assael, Y. M., de Freitas, N., and Whiteson, S. (2016). Learning to communicate with deep multi-agent reinforcement learning. *CoRR*, abs/1605.06676.

[Fotouhi et al., 2019] Fotouhi, A., Ding, M., Galati Giordano, L., Hassan, M., Li, J., and Lin, Z. (2019). Joint optimization of access and backhaul links for uavs based on reinforcement learning. In *2019 IEEE Globecom Workshops (GC Wkshps)*, pages 1–6.

[François-Lavet et al., 2018] François-Lavet, V., Henderson, P., Islam, R., Bellemare, M. G., and Pineau, J. (2018). *An Introduction to Deep Reinforcement Learning*, volume 11. Now.

[Freed et al., 2022] Freed, B., Kapoor, A., Abraham, I., Schneider, J., and Choset, H. (2022). Learning collaborative multi-agent policies with partial reward decoupling. *IEEE Robotics and Automation Letters*, 7(2):890–897.

[Galkin, 2019] Galkin, B. (2019). *On the Performance and Design Tradeoffs of Low Altitude UAV Small Cells in Urban Environments*. PhD thesis, School of Engineering, Trinity College (Dublin, Ireland).

[Galkin, 2021] Galkin, B. (2021). Consumer and Commercial Drones: How a technological revolution is impacting Irish society. `https://data.oireachtas.ie/ie/oireachtas/libraryResearch/2021/2021-02-11_spotlight-consumer-and-commercial-drones-how-a-technological-revolution-is-impacting-irish-society_en.pdf`. Accessed: 2022-10-19.

[Galkin et al., 2022a] Galkin, B., Fonseca, E., Amer, R., A. DaSilva, L., and Dusparic, I. (2022a). Reqiba: Regression and deep q-learning for intelligent uav cellular user to base

station association. *IEEE Transactions on Vehicular Technology*, 71(1):5–20.

[Galkin et al., 2016] Galkin, B., Kibilda, J., and DaSilva, L. A. (2016). Deployment of uav-mounted access points according to spatial user locations in two-tier cellular networks. In *2016 Wireless Days (WD)*, pages 1–6.

[Galkin et al., 2019a] Galkin, B., Kibilda, J., and DaSilva, L. A. (2019a). Uavs as mobile infrastructure: Addressing battery lifetime. *IEEE Communications Magazine*, 57(6):132–137.

[Galkin et al., 2019b] Galkin, B., Kibiłda, J., and DaSilva, L. A. (2019b). A stochastic model for uav networks positioned above demand hotspots in urban environments. *IEEE Transactions on Vehicular Technology*, 68(7):6985–6996.

[Galkin et al., 2022b] Galkin, B., Omoniwa, B., and Dusparic, I. (2022b). Multi-agent deep reinforcement learning for optimising energy efficiency of fixed-wing uav cellular access points. In *ICC 2022 - IEEE International Conference on Communications*, pages 1–6.

[Gao et al., 2021] Gao, L., Wang, S., Guan, Z., and Xu, W. (2021). Optimum deployment of uav relaying with mobile ground user system. In *2021 IEEE/CIC International Conference on Communications in China (ICCC)*, pages 1183–1188.

[Garcia Nocetti et al., 2002] Garcia Nocetti, F., Stojmenovic, I., and Zhang, J. (2002). Addressing and routing in hexagonal networks with applications for tracking mobile users and connection rerouting in cellular networks. *IEEE Transactions on Parallel and Distributed Systems*, 13(9):963–971.

[Gerasenko et al., 2001] Gerasenko, S., Joshi, A., Rayaprolu, S., Ponnavaikko, K., and Agrawal, D. (2001). Beacon signals: what, why, how, and where? *Computer*, 34(10):108–110.

[Goldman and Zilberstein, 2003] Goldman, C. V. and Zilberstein, S. (2003). Optimizing information exchange in collaborative multi-agent systems. In *Proceedings of the Second International Joint Conference on Autonomous Agents and Multiagent Systems*, AAMAS '03, page 137–144, New York, NY, USA. Association for Computing Machinery.

[Goldsmith and Wicker, 2002] Goldsmith, A. and Wicker, S. (2002). Design challenges for energy-constrained ad hoc wireless networks. *IEEE Wireless Communications*, 9(4):8–27.

[Gronauer and Diepold, 2022] Gronauer, S. and Diepold, K. (2022). Multi-agent deep reinforcement learning: a survey. *Artificial Intelligence Review*, 55(2):895–943.

[Guériau and Dusparic, 2020] Guériau, M. and Dusparic, I. (2020). Quantifying the impact of connected and autonomous vehicles on traffic efficiency and safety in mixed traffic. In *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, pages 1–8.

[Hadiwardoyo et al., 2019] Hadiwardoyo, S. A., Calafate, C. T., Cano, J.-C., Ji, Y., Hernández-Orallo, E., and Manzoni, P. (2019). Evaluating uav-to-car communications performance: From testbed to simulation experiments. In *2019 16th IEEE Annual Consumer Communications & Networking Conference (CCNC)*, pages 1–6.

[Hadiwardoyo et al., 2020] Hadiwardoyo, S. A., Calafate, C. T., Cano, J.-C., Krinkin, K., Klionskiy, D., Hernández-Orallo, E., and Manzoni, P. (2020). Three dimensional uav positioning for dynamic uav-to-car communications. *Sensors*, 20(2).

[Hadj-Kacem et al., 2020] Hadj-Kacem, I., Braham, H., and Jemaa, S. B. (2020). Sinr and rate distributions for downlink cellular networks. *IEEE Transactions on Wireless Communications*, 19(7):4604–4616.

[Hanna et al., 2019] Hanna, S., Yan, H., and Cabric, D. (2019). Distributed uav placement optimization for collaborative line-of-sight mimo communications. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4619–4623.

[Hasselt et al., 2016] Hasselt, H. v., Guez, A., and Silver, D. (2016). Deep reinforcement learning with double q-learning. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, AAAI'16, page 2094–2100. AAAI Press.

[Hayat et al., 2016] Hayat, S., Yanmaz, E., and Muzaffar, R. (2016). Survey on unmanned aerial vehicle networks for civil applications: A communications viewpoint. *IEEE Com-*

munications Surveys & Tutorials, 18(4):2624–2661.

[Hribar et al., 2022] Hribar, J., Marinescu, A., Chiumento, A., and Dasilva, L. A. (2022). Energy-aware deep reinforcement learning scheduling for sensors correlated in time and space. *IEEE Internet of Things Journal*, 9(9):6732–6744.

[Hu et al., 2020] Hu, J., Zhang, H., Song, L., Schober, R., and Poor, H. V. (2020). Collaborative internet of uavs: Distributed trajectory design by multi-agent deep reinforcement learning. *IEEE Transactions on Communications*, 68(11):6807–6821.

[Huang and Xu, 2021] Huang, Z. and Xu, X. (2021). Dqn-based relay deployment and trajectory planning in consensus-based multi-uavs tracking network. In *2021 IEEE International Conference on Communications Workshops (ICC Workshops)*, pages 1–7.

[Islam et al., 2022] Islam, M. M., Saad, M. M., Raza Khan, M. T., and Shah, S. H. A. (2022). Proactive uavs placement in vanets. In *ICC 2022 - IEEE International Conference on Communications*, pages 1–7.

[Jaques et al., 2018] Jaques, N., Lazaridou, A., Hughes, E., Gulcehre, C., Ortega, P. A., Strouse, D., Leibo, J. Z., and de Freitas, N. (2018). Social influence as intrinsic motivation for multi-agent deep reinforcement learning.

[Jiang et al., 2018] Jiang, J., Dun, C., and Lu, Z. (2018). Graph convolutional reinforcement learning for multi-agent cooperation. *CoRR*, abs/1810.09202.

[Kakade, 2003] Kakade, M. S. (2003). *On the Sample Complexity of Reinforcement Learning*. PhD thesis, Gatsby Computational Neuroscience Unit, University College London.

[Kalantari et al., 2017] Kalantari, E., Shakir, M. Z., Yanikomeroglu, H., and Yongacoglu, A. (2017). Backhaul-aware robust 3d drone placement in 5g+ wireless networks. In *2017 IEEE International Conference on Communications Workshops (ICC Workshops)*, pages 109–114.

[Kalantari et al., 2016] Kalantari, E., Yanikomeroglu, H., and Yongacoglu, A. (2016). On the number and 3d placement of drone base stations in wireless cellular networks. In *2016*

IEEE 84th Vehicular Technology Conference (VTC-Fall), pages 1–6.

[Kim et al., 2019a] Kim, D., Moon, S., Hostallero, D., Kang, W. J., Lee, T., Son, K., and Yi, Y. (2019a). Learning to schedule communication in multi-agent reinforcement learning. *CoRR*, abs/1902.01554.

[Kim et al., 2019b] Kim, W., Cho, M., and Sung, Y. (2019b). Message-dropout: An efficient training method for multi-agent deep reinforcement learning. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 6079–6086. AAAI Press.

[Lee and Lee, 2021] Lee, H.-R. and Lee, T. (2021). Multi-agent reinforcement learning algorithm to solve a partially-observable multi-agent problem in disaster response. *European Journal of Operational Research*, 291(1):296–308.

[Lee et al., 2021a] Lee, I., Babu, V., Caesar, M., and Nicol, D. (2021a). Deep reinforcement learning for uav-assisted emergency response. In *MobiQuitous 2020 - 17th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*, MobiQuitous '20, page 327–336, New York, NY, USA. Association for Computing Machinery.

[Lee et al., 2021b] Lee, W., Jeon, Y., Kim, T., and Kim, Y.-I. (2021b). Deep reinforcement learning for uav trajectory design considering mobile ground users. *Sensors*, 21(24).

[Lesser, 1999] Lesser, V. (1999). Collaborative multiagent systems: a personal view of the state of the art. *IEEE Transactions on Knowledge and Data Engineering*, 11(1):133–142.

[Li et al., 2022] Li, P., Tang, H., Yang, T., Hao, X., Sang, T., Zheng, Y., Hao, J., Taylor, M. E., Tao, W., and Wang, Z. (2022). Pmic: Improving multi-agent reinforcement learning with progressive mutual information collaboration.

[Lillicrap et al., 2015] Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., and Wierstra, D. (2015). Continuous control with deep reinforcement learning.

[Lin et al., 2020] Lin, N., Fu, L., Zhao, L., Min, G., Al-Dubai, A., and Gacanin, H. (2020). A novel multimodal collaborative drone-assisted vanet networking model. *IEEE Transactions on Wireless Communications*, 19(7):4919–4933.

[Liu et al., 2018] Liu, C. H., Chen, Z., Tang, J., Xu, J., and Piao, C. (2018). Energy-efficient uav control for effective and fair communication coverage: A deep reinforcement learning approach. *IEEE Journal on Selected Areas in Communications*, 36(9):2059–2070.

[Liu et al., 2020] Liu, C. H., Ma, X., Gao, X., and Tang, J. (2020). Distributed energy-efficient multi-uav navigation for long-term communication coverage by deep reinforcement learning. *IEEE Transactions on Mobile Computing*, 19(6):1274–1285.

[Liu et al., 2019a] Liu, X., Liu, Y., and Chen, Y. (2019a). Reinforcement learning in multiple-uav networks: Deployment and movement design. *IEEE Transactions on Vehicular Technology*, 68(8):8036–8049.

[Liu et al., 2019b] Liu, X., Liu, Y., Chen, Y., and Hanzo, L. (2019b). Trajectory design and power control for multi-uav assisted wireless networks: A machine learning approach. *IEEE Transactions on Vehicular Technology*, 68(8):7957–7969.

[Lowe et al., 2017] Lowe, R., Wu, Y., Tamar, A., Harb, J., Abbeel, P., and Mordatch, I. (2017). Multi-agent actor-critic for mixed collaborative-competitive environments.

[Lyu et al., 2017] Lyu, J., Zeng, Y., Zhang, R., and Lim, T. J. (2017). Placement optimization of uav-mounted mobile base stations. *IEEE Communications Letters*, 21(3):604–607.

[Mannion et al., 2018] Mannion, P., Devlin, S., Duggan, J., and Howley, E. (2018). Reward shaping for knowledge-based multi-objective multi-agent reinforcement learning. *The Knowledge Engineering Review*, 33:e23.

[Marinescu, 2016] Marinescu, A. (2016). *Prediction-Based Multi-Agent Reinforcement Learning for Inherently Non-Stationary Environments*. PhD thesis, School of Computer Science & Statistics, Trinity College (Dublin, Ireland).

[Marini et al., 2022] Marini, R., Park, S., Simeone, O., and Buratti, C. (2022). Continual Meta-Reinforcement Learning for UAV-Aided Vehicular Wireless Networks. *arXiv e-prints*, page arXiv:2207.06131.

[Mnih et al., 2015] Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., and Hassabis, D. (2015). Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533.

[Montero et al., 2019] Montero, E., Rosário, D., and Santos, A. (2019). Clustering users for the deployment of uav as base station to improve the quality of the data. In *2019 IEEE Latin-American Conference on Communications (LATINCOM)*, pages 1–6.

[Mozaffari et al., 2017] Mozaffari, M., Saad, W., Bennis, M., and Debbah, M. (2017). Mobile unmanned aerial vehicles (uavs) for energy-efficient internet of things communications. *IEEE Transactions on Wireless Communications*, 16(11):7574–7589.

[Mozaffari et al., 2019] Mozaffari, M., Saad, W., Bennis, M., Nam, Y.-H., and Debbah, M. (2019). A tutorial on uavs for wireless networks: Applications, challenges, and open problems. *IEEE Communications Surveys & Tutorials*, 21(3):2334–2360.

[Oliehoek and Amato, 2016] Oliehoek, F. A. and Amato, C. (2016). *A Concise Introduction to Decentralized POMDPs*. Springer Publishing Company, Incorporated, 1st edition.

[Oliehoek and Spaan, 2012] Oliehoek, F. A. and Spaan, M. T. J. (2012). Tree-based solution methods for multiagent pomdps with delayed communication. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*, AAAI'12, page 1415–1421. AAAI Press.

[Omidshafiei et al., 2017] Omidshafiei, S., Pazis, J., Amato, C., How, J. P., and Vian, J. (2017). Deep decentralized multi-task multi-agent reinforcement learning under partial observability. *CoRR*, abs/1703.06182.

[Omoniwa et al., 2019] Omoniwa, B., Hussain, R., Javed, M. A., Bouk, S. H., and Malik, S. A. (2019). Fog/edge computing-based iot (feciot): Architecture, applications, and research issues. *IEEE Internet of Things Journal*, 6(3):4118–4149.

[Oubbati et al., 2019] Oubbati, O. S., Chaib, N., Lakas, A., Lorenz, P., and Rachedi, A. (2019). Uav-assisted supporting services connectivity in urban vanets. *IEEE Transactions on Vehicular Technology*, 68(4):3944–3951.

[Panait and Luke, 2005] Panait, L. and Luke, S. (2005). Collaborative multi-agent learning: The state of the art. *Autonomous Agents and Multi-Agent Systems*, 11:387–434.

[Papoudakis et al., 2019] Papoudakis, G., Christianos, F., Rahman, A., and Albrecht, S. V. (2019). Dealing with non-stationarity in multi-agent deep reinforcement learning.

[Papoudakis et al., 2020] Papoudakis, G., Christianos, F., Schäfer, L., and Albrecht, S. V. (2020). Comparative evaluation of multi-agent deep reinforcement learning algorithms. *CoRR*, abs/2006.07869.

[Peng and Shen, 2020] Peng, H. and Shen, X. S. (2020). Ddpg-based resource management for mec/uav-assisted vehicular networks. In *2020 IEEE 92nd Vehicular Technology Conference (VTC2020-Fall)*, pages 1–6.

[Pesce and Montana, 2019] Pesce, E. and Montana, G. (2019). Improving coordination in multi-agent deep reinforcement learning through memory-driven communication. *CoRR*, abs/1901.03887.

[Phan et al., 2021] Phan, T., Belzner, L., Gabor, T., Sedlmeier, A., Ritz, F., and Linnhoff-Popien, C. (2021). Resilient multi-agent reinforcement learning with adversarial value decomposition. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(13):11308–11316.

[Poudel and Moh, 2019] Poudel, S. and Moh, S. (2019). Medium access control protocols for unmanned aerial vehicle-aided wireless sensor networks: A survey. *IEEE Access*, 7:65728–65744.

[Ranjan Kumar and Varakantham, 2020] Ranjan Kumar, R. and Varakantham, P. (2020). On Solving Collaborative MARL Problems with a Few Good Experiences. *arXiv e-prints*, page arXiv:2001.07993.

[Rashid et al., 2018] Rashid, T., Samvelyan, M., de Witt, C. S., Farquhar, G., Foerster, J. N., and Whiteson, S. (2018). QMIX: monotonic value function factorisation for deep multi-agent reinforcement learning. *CoRR*, abs/1803.11485.

[Raza et al., 2021] Raza, A., Bukhari, S. H. R., Aadil, F., and Iqbal, Z. (2021). An uav-assisted vanet architecture for intelligent transportation system in smart cities. *International Journal of Distributed Sensor Networks*, 17(7):15501477211031750.

[Roy, 2011] Roy, R. R. (2011). *Random Walk Mobility*, pages 35–63. Springer US, Boston, MA.

[Ruan et al., 2018] Ruan, L., Wang, J., Chen, J., Xu, Y., Yang, Y., Jiang, H., Zhang, Y., and Xu, Y. (2018). Energy-efficient multi-uav coverage deployment in uav networks: A game-theoretic framework. *China Communications*, 15(10):194–209.

[Samir et al., 2021] Samir, M., Ebrahimi, D., Assi, C., Sharafeddine, S., and Ghrayeb, A. (2021). Leveraging uavs for coverage in cell-free vehicular networks: A deep reinforcement learning approach. *IEEE Transactions on Mobile Computing*, 20(9):2835–2847.

[Sanchez-Aguero et al., 2020] Sanchez-Aguero, V., Valera, F., Vidal, I., Tipantuña, C., and Hesselbach, X. (2020). Energy-aware management in multi-uav deployments: Modelling and strategies. *Sensors*, 20(10):2791.

[Saxena et al., 2019] Saxena, V., Jaldén, J., and Klessig, H. (2019). Optimal uav base station trajectories using flow-level models for reinforcement learning. *IEEE Transactions on Cognitive Communications and Networking*, 5(4):1101–1112.

[Shakhatreh et al., 2017] Shakhatreh, H., Khreishah, A., Alsarhan, A., Khalil, I., Sawalmeh, A., and Othman, N. S. (2017). Efficient 3d placement of a uav using particle swarm optimization. In *2017 8th International Conference on Information and Communication Systems (ICICS)*, pages 258–263.

[Shakhatreh et al., 2019] Shakhatreh, H., Sawalmeh, A. H., Al-Fuqaha, A., Dou, Z., Almaita, E., Khalil, I., Othman, N. S., Khreishah, A., and Guizani, M. (2019). Unmanned aerial vehicles (uavs): A survey on civil applications and key research challenges. *IEEE Access*, 7:48572–48634.

[Sherman et al., 2021] Sherman, M., Shao, S., Sun, X., and Zheng, J. (2021). Uav assisted cellular networks with renewable energy charging infrastructure: A reinforcement learning approach. In *MILCOM 2021 - 2021 IEEE Military Communications Conference (MILCOM)*, pages 495–502.

[Shi et al., 2022] Shi, D., Tong, J., Liu, Y., and Fan, W. (2022). Knowledge reuse of multi-agent reinforcement learning in collaborative tasks. *Entropy (Basel, Switzerland)*, 24(4):895–943.

[Simoes et al., 2020] Simoes, D., Lau, N., and Reis, L. P. (2020). Multi agent deep learning with collaborative communication. *Journal of Artificial Intelligence and Soft Computing Research*, 10(3):189–207.

[Solmaz and Turgut, 2019] Solmaz, G. and Turgut, D. (2019). A survey of human mobility models. *IEEE Access*, 7:125711–125731.

[Stone et al., 2010] Stone, P., Kaminka, G. A., Kraus, S., and Rosenschein, J. S. (2010). Ad hoc autonomous agent teams: Collaboration without pre-coordination. In *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence*, AAAI'10, page 1504–1509. AAAI Press.

[Sukhbaatar et al., 2016] Sukhbaatar, S., Szlam, A., and Fergus, R. (2016). Learning multi-agent communication with backpropagation. *CoRR*, abs/1605.07736.

[Sun et al., 2023] Sun, R., Zhao, D., Ding, L., Zhang, J., and Ma, H. (2023). Uav-net+: Effective and energy-efficient uav network deployment for extending cell tower coverage with dynamic demands. *IEEE Transactions on Vehicular Technology*, 72(1):973–985.

[Sutton and Barto, 2018] Sutton, R. S. and Barto, A. G. (2018). *Reinforcement Learning: An Introduction*. The MIT Press, second edition.

[Számadó, 2010] Számadó, S. (2010). Pre-hunt communication provides context for the evolution of early human language. *Biological Theory*, 5(4):366–382.

[Tan and Guan, 2022] Tan, J. and Guan, W. (2022). Resource allocation of fog radio access network based on deep reinforcement learning. *Engineering Reports*, 4(5):e12497.

[Tan, 1993] Tan, M. (1993). Multi-agent reinforcement learning: Independent vs. collaborative agents. In *In Proceedings of the Tenth International Conference on Machine Learning*, pages 330–337. Morgan Kaufmann.

[Taylor, 2015] Taylor, A. (2015). *Parallel Transfer Learning: Accelerating Reinforcement Learning in Multi-Agent Systems.* PhD thesis, School of Computer Science & Statistics, Trinity College (Dublin, Ireland).

[Tenorio-Gonzalez et al., 2010] Tenorio-Gonzalez, A. C., Morales, E. F., and Villaseñor-Pineda, L. (2010). Dynamic reward shaping: Training a robot by voice. In Kuri-Morales, A. and Simari, G. R., editors, *Advances in Artificial Intelligence – IBERAMIA 2010*, pages 483–492, Berlin, Heidelberg. Springer Berlin Heidelberg.

[Terry et al., 2020] Terry, J. K., Grammel, N., Son, S., and Black, B. (2020). Parameter sharing for heterogeneous agents in multi-agent reinforcement learning. *CoRR*, abs/2005.13625.

[van Hasselt, 2012] van Hasselt, H. (2012). *Reinforcement Learning in Continuous State and Action Spaces*, pages 207–251. Springer Berlin Heidelberg, Berlin, Heidelberg.

[Vinogradov et al., 2020] Vinogradov, E., Minucci, F., and Pollin, S. (2020). Wireless communication for safe uavs: From long-range deconfliction to short-range collision avoidance. *IEEE Vehicular Technology Magazine*, 15(2):88–95.

[Wang et al., 2021] Wang, L., Wang, K., Pan, C., Xu, W., Aslam, N., and Hanzo, L. (2021). Multi-agent deep reinforcement learning-based trajectory planning for multi-uav assisted mobile edge computing. *IEEE Transactions on Cognitive Communications and Networking*, 7(1):73–84.

[Warrier et al., 2022] Warrier, A., Al-Rubaye, S., Panagiotakopoulos, D., Inalhan, G., and Tsourdos, A. (2022). Interference mitigation for 5g-connected uav using deep q-learning framework. In *2022 IEEE/AIAA 41st Digital Avionics Systems Conference (DASC)*, pages 1–8.

[Watkins and Dayan, 1992] Watkins, C. J. C. H. and Dayan, P. (1992). Q-learning. *Machine Learning*, 8(3):279–292.

[Wong et al., 2022] Wong, A., Bäck, T., Kononova, A. V., and Plaat, A. (2022). Deep multiagent reinforcement learning: challenges and directions. *Artificial Intelligence Review*.

[Wu et al., 2021] Wu, H., Li, H., Zhang, J., Wang, Z., and Zhang, J. (2021). Generating individual intrinsic reward for collaborative multiagent reinforcement learning. *International Journal of Advanced Robotic Systems*, 18(5):17298814211044946.

[Xu et al., 2011] Xu, A., Viriyasuthee, C., and Rekleitis, I. (2011). Optimal complete terrain coverage using an unmanned aerial vehicle. In *2011 IEEE International Conference on Robotics and Automation*, pages 2513–2519.

[Xue et al., 2019] Xue, Z., Wang, J., Ding, G., Zhou, H., and Wu, Q. (2019). Maximization of data dissemination in uav-supported internet of things. *IEEE Wireless Communications Letters*, 8(1):185–188.

[Yan et al., 2019] Yan, C., Fu, L., Zhang, J., and Wang, J. (2019). A comprehensive survey on uav communication channel modeling. *IEEE Access*, 7:107769–107792.

[Yuan et al., 2021] Yuan, T., Rothenberg, C. E., Obraczka, K., Barakat, C., and Turletti, T. (2021). Harnessing uavs for fair 5g bandwidth allocation in vehicular communication via deep reinforcement learning. *IEEE Transactions on Network and Service Management*, 18(4):4063–4074.

[Zeng et al., 2019] Zeng, Y., Xu, J., and Zhang, R. (2019). Energy minimization for wireless communication with rotary-wing uav. *IEEE Transactions on Wireless Communications*, 18(4):2329–2345.

[Zeng and Zhang, 2017] Zeng, Y. and Zhang, R. (2017). Energy-efficient uav communication with trajectory optimization. *IEEE Transactions on Wireless Communications*, 16(6):3747–3760.

[Zhang et al., 2021a] Zhang, K., Yang, Z., and Başar, T. (2021a). *Multi-Agent Reinforcement Learning: A Selective Overview of Theories and Algorithms*, pages 321–384. Springer International Publishing, Cham.

[Zhang et al., 2018] Zhang, K., Yang, Z., Liu, H., Zhang, T., and Basar, T. (2018). Fully decentralized multi-agent reinforcement learning with networked agents. *CoRR*, abs/1802.08757.

[Zhang et al., 2021b] Zhang, M., Fu, S., and Fan, Q. (2021b). Joint 3d deployment and power allocation for uav-bs: A deep reinforcement learning approach. *IEEE Wireless Communications Letters*, 10(10):2309–2312.

[Zhang et al., 2020a] Zhang, N., Liu, J., Xie, L., and Tong, P. (2020a). A deep reinforcement learning approach to energy-harvesting uav-aided data collection. In *2020 International Conference on Wireless Communications and Signal Processing (WCSP)*, pages 93–98.

[Zhang et al., 2022] Zhang, Q., Ferdowsi, A., Saad, W., and Bennis, M. (2022). Distributed conditional generative adversarial networks (gans) for data-driven millimeter wave communications in uav networks. *IEEE Transactions on Wireless Communications*, 21(3):1438–1452.

[Zhang et al., 2020b] Zhang, T., Xu, H., Wang, X., Wu, Y., Keutzer, K., Gonzalez, J. E., and Tian, Y. (2020b). Multi-agent collaboration via reward attribution decomposition.

[Zhao et al., 2022] Zhao, X., Yi, P., and Li, L. (2022). Distributed policy evaluation via inexact admm in multi-agent reinforcement learning. *Control Theory and Technology*, 18(4):362–378.

[Zhu et al., 2022] Zhu, C., Dastani, M., and Wang, S. (2022). A survey of multi-agent reinforcement learning with communication.

[Zou et al., 2019] Zou, H., Ren, T., Yan, D., Su, H., and Zhu, J. (2019). Reward shaping via meta-learning.