

Unmanned Aerial Vehicles for 5G and Beyond: Optimization and Deep Learning

Moataz Shoukry

A Thesis

in

The Department

of

Information and Systems Engineering

**Presented in Partial Fulfillment of the Requirements for the Degree of
Doctor of Philosophy (Information and Systems Engineering) at
Concordia University
Montréal, Québec, Canada**

February 2021

© Moataz Shoukry, 2021

CONCORDIA UNIVERSITY

School of Graduate Studies

This is to certify that the thesis prepared

By: **Moataz Shoukry**

Entitled: **Unmanned Aerial Vehicles for 5G and Beyond: Optimization and Deep Learning**

and submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy (Information and Systems Engineering)

complies with the regulations of this University and meets the accepted standards with respect to originality and quality.

Signed by the Final Examining Committee:

_____	Chair
<i>Dr. Hovhannes Harutyunyan</i>	
_____	External Examiner
<i>Dr. Halim Yanikomeroglu</i>	
_____	External to Program
<i>Dr. Yousef R. Shayan</i>	
_____	Examiner
<i>Dr. Roch Glitho</i>	
_____	Examiner
<i>Dr. Nizar Bouguila</i>	
_____	Supervisor
<i>Dr. Chadi Assi</i>	
_____	Co-supervisor
<i>Dr. Ali Ghrayeb</i>	
_____	Co-supervisor
<i>Dr. Sanaa Sharafeddine</i>	

Approved by

Dr. Abdessamad Ben Hamza, Chair
Department of Information and Systems Engineering

February, 22, 2021

Dr. Mourad Debbabi, Interim Dean,
Gina Cody School of Engineering and Computer Science

Abstract

Unmanned Aerial Vehicles for 5G and Beyond: Optimization and Deep Learning

Moataz Shoukry, Ph.D.

Concordia University, 2021

Aerial platforms and, more precisely, Unmanned Aerial Vehicles (UAVs) or drones augmented with ubiquitous computing, processing and wireless communication technologies are expected to play an important role in next-generation cellular networks. The flexibility, autonomy, altitude adaptiveness, and controllable mobility of UAVs render them suitable to be part of the future wireless access. Nonetheless, combined terrestrial and UAV communication networks are capable of substantially improving network coverage and Quality of Service by leveraging line-of-sight communication as well as minimizing the delay and age-of-information for UAV to ground communication. Despite its numerous advantages, the deployment of UAVs faces different challenges with respect to wireless networks, ranging from radio resource management to UAVs' trajectory under energy limitation constraint and minimal knowledge of the environment. To this end, this dissertation aims to address the challenges in the efficient deployment of UAVs in future networks under various performance metrics. The key goal of this dissertation is to provide the analytical foundations for deployment, learning, in-depth analysis, and optimization of UAV-assisted wireless communication networks. Towards achieving this goal, this dissertation makes significant contributions to several areas of UAV-assisted wireless communication networks within the contexts of static environments as well as high mobility environments. For the deployment of UAVs in static environments such as Internet of Things (IoT) wireless networks, various tools such as optimization theory and machine learning frameworks are employed to enable UAV trajectory design under different scenarios and performance metrics. Results demonstrate the effectiveness of the proposed designs. In particular, UAVs adapt their mobility and altitude to enable reliable and energy efficient communication,

to maximize service for IoT applications, and to maintain the freshness of information. For the deployment of UAVs in high mobility environments such as vehicular networks, unique design challenges are considered and carefully handled to guarantee effective performance of the UAV. Particularly, the high mobility of the vehicles leads to distinct network conditions and changes the network topology. The challenge here is that designing an efficient deployment of UAVs while considering the complex and dynamic network conditions is not a trivial task. This challenge was addressed through comprehensive studies that led to effective, robust, and high-performance solutions. Different performance metrics such as coverage, age of information, throughput, and Quality of Service were evaluated and compared with other approaches. Results shed light on the trade-offs in the vehicular network such as throughput-latency when exploiting UAV mobility for service. The findings in this dissertation highlight key guidelines for the effective design of UAV assisted wireless communication networks. More insights on the efficient deployment of UAVs in static and high mobility environments are provided in order to assist and enhance communication in future networks while considering the unique features of UAVs such as their flight time, mobility, energy budget, and altitude.

Acknowledgments

First and foremost, all my gratitude goes to God the Almighty, Alhamdulillah. It is my utmost belief in Him and His boundless blessings that allowed me to finish this work.

During my Ph.D. journey, I have been very fortunate to have met many great individuals. I would like to thank my supervisor, Dr. Chadi Assi, for his guidance during the course of my research and for being my role model as a great engineer and teacher. Through numerous discussions and one-on-one meetings, I learned so much from him and I greatly appreciate all the advice and wisdom. I would also like to express my appreciation to my co-supervisors, Dr. Ali Ghrayeb and Dr. Sanaa Sharafeddine for their constant support and mentorship. Their knowledge and work ethics have always inspired me. I would also like to thank Dr. Dariush Ebrahimi, the wonderful research mentor that I had the chance to work with during the course of this thesis. My gratitude also goes to my colleague, Ahmed Al-Hilo. I thank him for his collaborations and his friendship, wishing him the best of luck in his future endeavors. I wish to express my appreciation to every reviewer and editor for their valuable comments and suggestions towards my publications. I would like also to thank the members of my Ph.D. advisory committee for their valuable comments which have helped me to substantially improve the quality of this dissertation. I also thank my brothers, my wife, and my daughters. I would finally like to express my deepest gratitude to my parents to whom this work is dedicated. Without their unconditional support, kind words, and sound advice, I would not be the person I am today.

"A goal without a plan is just a wish."

–Antoine de Saint-Exupéry–

Contents

List of Figures	xi
List of Tables	xiv
List of Abbreviations	xv
1 Introduction	1
1.1 New Communications Era with UAVs	1
1.2 Potential Use Cases of UAVs	4
1.2.1 On-Demand and Dynamic Aerial Networks	4
1.2.2 UAV-Assisted Communication in Disaster Scenarios	5
1.2.3 UAV-Assisted IoT Communication	6
1.2.4 Aerial Edge Caching in Vehicular Networks	6
1.3 Motivations and Contributions	7
1.3.1 Limitations of Existing Works	7
1.3.2 Thesis Contributions	9
1.4 List of Publications	15
1.4.1 Journal Publications	15
1.4.2 Conference Publications	16
2 Optimized UAV Trajectory Planning in IoT Networks for Maximum Service	17
2.1 Background, Related Works, and Contributions	17
2.2 System Model	21

2.3	Problem Formulation	24
2.4	Global Optimization Solution	25
2.4.1	Equivalent Formulation	26
2.4.2	Proposed BRB Solution	27
2.4.3	Convergence Analysis	29
2.5	Low-Complexity Sub-optimal Solution	29
2.5.1	SCA-Algorithm for Maximizing the Number of Served IoTDS	29
2.5.2	Complexity Analysis	33
2.6	Minimizing UAV Flight Distance	33
2.7	Greedy Location/Deadline-based Algorithms	35
2.8	Simulation Results and Discussion	36
2.9	Summary	47
3	UAVs as Active Relays in IoT Networks for Fresh Information	48
3.1	Background, Related Works, and Contributions	48
3.2	System Model	51
3.2.1	IoT-UAV-BS Channel Model	53
3.2.2	Definition of Age of Information	56
3.2.3	Optimization Problem Formulation	57
3.3	Proposed Solution	60
3.3.1	MDP Formulation	63
3.3.2	Proximal Policy Optimization Proposed Solution	65
3.3.3	PPO Analysis	69
3.4	Simulation Results and Discussion	70
3.5	Summary	77
3.6	Closed-Form and Upper Bound	78
4	UAV-mounted RIS as Passive Relays in IoT Networks for Fresh Information	81
4.1	Background, Related Works, and Contributions	81
4.2	System Model and Problem Formulation	83

4.3	Proposed Solution	88
4.3.1	MDP Formulation	89
4.3.2	Proposed Solution Description	90
4.4	Simulation Results and Discussion	92
4.5	Summary	95
5	Learning-based Trajectory Planning of Aerial Base Stations in Vehicular Networks	96
5.1	Background, Related Works, and Contributions	96
5.2	The Communication Scenario	102
5.3	Optimization Problem Formulation	105
5.4	The Proposed Deep Reinforcement Learning Approach	108
5.4.1	Deep Reinforcement Learning Background	108
5.4.2	Input From the Environment	110
5.4.3	Actions and Expected Rewards	111
5.4.4	Solution Algorithm	113
5.4.5	Complexity Analysis	115
5.5	Simulation Results and Discussion	116
5.6	Summary	121
6	AoI-Aware Data Collection in Vehicular Networks with Intelligent UAVs	126
6.1	Background, Related Works, and Contributions	126
6.2	System Model	131
6.2.1	The Communication Scenario	131
6.2.2	AoI Definition in Vehicular Networks	133
6.3	The Proposed Deep Reinforcement Learning Approach	136
6.3.1	Deep Reinforcement Learning Background	138
6.3.2	Input From the Environment	140
6.3.3	Actions and Expected Rewards	140
6.3.4	Solution Algorithm	141
6.3.5	Complexity Analysis	147

6.4	Simulation Results and Discussion	148
6.5	Summary	155
7	On-Demand Content Delivery in Vehicular Networks with Optimized Multi-UAV Trajectories	158
7.1	Background, Related Works, and Contributions	158
7.2	System Model	159
7.3	Problem Formulation	162
7.4	Proposed Solution	164
7.5	Simulation Results and Discussion	167
7.6	Summary	169
8	Conclusions and Future Research Directions	170
8.1	Conclusion	170
8.2	Future Work	172
8.2.1	Further Deployment of UAVs for 5G and Beyond Networks	172
8.2.2	Safety and Security with UAVs	173
	Bibliography	174

List of Figures

Fig. 1.1	UAV classification.	3
Fig. 1.2	Different use cases of UAVs.	5
Fig. 1.3	Graphical summary of addressed problems in future wireless networks.	10
Fig. 2.1	System model: timely data collection in a smart city environment using UAV. For illustration, an example of the time line representation for two IoTDS with two deadlines are presented.	20
Fig. 2.2	BRB-optimal algorithm to maximize the number of served IoTDS.	38
Fig. 2.3	UAV trajectory to maximize the number of served IoTDS.	39
Fig. 2.4	Achieved service amount per IoTDS.	40
Fig. 2.5	Performance of proposed SCA-algorithm.	40
Fig. 2.6	Percentage of served IoTDS for SCA-algorithm.	41
Fig. 2.7	Performance of proposed SCA-distance.	42
Fig. 2.8	Achieved service amount over different channel realizations for the Enhanced- algorithm.	43
Fig. 2.9	Total energy consumption of the UAV.	44
Fig. 2.10	Optimizing the UAV trajectory to maximize the number of served IoTDS. The values of the deadlines are represented next to each IoTDS.	45
Fig. 2.11	Percentage of served IoTDS for SCA-distance compared to alternative solutions.	46
Fig. 3.1	An illustration of our system model.	52
Fig. 3.2	The evolution of AoI.	55
Fig. 3.3	Achievable Rate versus UAV's Altitude for IoT-to-UAV Communication.	68

Fig. 3.4	Achievable Rate versus UAV's Altitude for UAV-to-BS Communication.	69
Fig. 3.5	Accumulated reward vs iterations.	73
Fig. 3.6	Accumulated reward vs iterations.	74
Fig. 3.7	UAV altitude vs time.	75
Fig. 3.8	The performance comparison of different policies for a sample of four IoTDs.	76
Fig. 3.9	Impact of number of IoTDs and comparisons.	77
Fig. 3.10	Average age per IoTD.	78
Fig. 4.1	ARIS-assisted IoT wireless networks	85
Fig. 4.2	Convergence.	91
Fig. 4.3	Impact of number of IoTDs.	92
Fig. 4.4	Average age per IoTD.	93
Fig. 4.5	Impact of number of RIS elements.	94
Fig. 5.1	A highway segment with no communication infrastructure where multiple UAVs covering vehicles.	102
Fig. 5.2	DRL-based proposed approach to obtain the control policy that governs the trajectories of the deployed UAVs.	114
Fig. 5.3	Accumulated reward over time.	117
Fig. 5.4	Accumulated reward over time for with prior knowledge.	118
Fig. 5.5	Performance vs time.	119
Fig. 5.6	Impact of r_{min}	120
Fig. 5.7	Impact of vehicular density.	121
Fig. 5.8	Performance evaluation and comparisons.	122
Fig. 5.9	Impact of energy saving.	122
Fig. 6.1	Intelligent transportation systems with multiple-UAVs used for collecting and processing status-update packets from vehicles.	129
Fig. 6.2	The evolution of AoI associated with vehicle i with the initial AoI of four time- slots, $A_i(1) = 4$	132
Fig. 6.3	DRL-based proposed approach to obtain the control policy.	145
Fig. 6.4	Impact of number of UAVs and comparisons.	148

Fig. 6.5	Impact of status-update size.	149
Fig. 6.6	Impact of vehicular density.	152
Fig. 6.7	Average age comparison with different policies.	153
Fig. 6.8	The performance comparison of different policies for a sample of three vehicles.	156
Fig. 6.9	Accumulated reward over time.	157
Fig. 7.1	A drive-thru scenario with multiple UAVs serving vehicles crossing a highway segment.	163
Fig. 7.2	UAVs trajectories.	165
Fig. 7.3	UAVs speeds.	166
Fig. 7.4	Resource allocation.	166
Fig. 7.5	Impact of S_i^{\min}	167

List of Tables

Table 1.1	List of advantages and challenges of UAV-assisted wireless networks.	7
Table 2.1	Simulation Parameters in UAV-assisted IoT Wireless Networks	37
Table 3.1	Table of Notations	58
Table 3.2	Simulation Parameters in UAV-Relay assisted IoT Networks	71
Table 3.3	List of Parameters for different environments.	74
Table 4.1	Simulation Parameters in UAV-RIS assisted IoT Networks	95
Table 5.1	Simulation Parameters in Intelligent Coverage Networks	125
Table 6.1	Optimization Problem Formulation	135
Table 6.2	Simulation Parameters in UAV-assisted Vehicular Networks	150

List of Abbreviations

3GPP	3rd Generation Partnership Project
5G	Fifth-Generation
6G	Sixth-Generation
A2G	Air to Ground
AC	Actor-Critic
AI	Artificial-Intelligence
AoI	Age of Information
AWGN	Additive White Gaussian Noise
B5G	Beyond Fifth-Generation
BRB	Branch, Reduce and Bound
BS	Base Station
CAPEX	Capital Expenditure
CSI	Channel State Information
DC	Difference of Convex
DDPG	Deep Deterministic Policy Gradient
DoF	Degree of Freedom
DP	Dynamic Programming
DQN	Deep Q-Network

DRL	Deep Reinforcement Learning
ESA	Expected Sum Age of Information
EWSA	Expected Weighted Sum Age of Information
FSO	Free Space Optics
FTGS	Fixed Trajectory with Greedy Scheduling
FTRS	Fixed Trajectory with Random Scheduling
GA	Genetic Algorithm
HAPs	High Altitude Platforms
HDGS	Heuristic Deployment with Greedy Scheduling
HGS	Hovering with Greedy Scheduling
HRS	Hovering with Random Scheduling
ICT	Information and Communications Technology
IoE	Internet of Everything
IoTds	Internet of Things Devices
IoT	Internet of Things
ITS	Intelligent Transportation System
LAPs	Low Altitude Platform
LiDARs	Light Detection and Ranging
LoS	Line of Sight
LP	Linear Programming
MDP	Markov Decision Process
MINLP	Mixed Integer Non Linear Program
MIQCP	Mixed Integer Quadratically Constrained Program
ML	Machine Learning

mmWave	Millimeter Wave
NLoS	Non Line of Sight
NN	Neural Network
NOMA	Non Orthogonal Multiple Access
OPEX	Operational Expenditure
P-DQN	Parametrized Deep Q-network
PPO	Proximal Policy Optimization
QoS	Quality of Service
RDRS	Random Deployment with Random Scheduling
ReLU	Rectified Linear Unit
RIS	Reconfigurable Intelligent Surface
RSUs	Road Side Units
RTGS	Random Trajectory with Greedy Scheduling
RTRS	Random Trajectory with Random Scheduling
SCA	Successive Convex Approximation
SGA	Stochastic Gradient Ascend
SNR	Signal to Noise Ratio
SON	Self Organized Networks
TDMA	Time Division Multiple Access
TPU	Tensor Processing Unit
TRPO	Trust Region Policy Optimization
UAVs	Unmanned Aerial Vehicles

Chapter 1

Introduction

1.1 New Communications Era with UAVs

Unlike earlier generations of cellular networks, future wireless networks under different labels such as beyond fifth-generation (B5G or 5G+) and sixth-generation (6G) are expected to seamlessly and ubiquitously connect everything and support very high data rates and diverse requirements on reliability and latency. They are also expected to support a myriad of services across different vertical industries such as augmented or virtual reality, autonomous driving, Internet of Space Things, remote healthcare, industrial automation, among others. Future wireless networks would therefore meet unprecedented demands for high quality wireless services, which impose challenges on the conventional terrestrial communication networks. In addition, with the inflation in communication traffic, the current cellular network infrastructure will experience a degradation in performance and will fail to provide quality service that meets the expectations of tomorrow's IoE. Cellular operators are therefore faced with the challenge of preparing their infrastructure to handle this exponential increase in data traffic and reduce the burden on the cellular network. To realize these goals, small cells are proposed to offload the traffic load and; thus, help in accommodating the massive traffic in future networks. Such technologies however usually require the deployment of new high cost fixed ground Base-Stations (BSs). While many other emerging technologies (such as Millimeter Wave (mmWave), Ultra densification, and Non-Orthogonal Multiple Access (NOMA)) are contemplated for networks of the future, each of such technologies has its own drawbacks and challenges.

Unmanned Aerial Vehicles (UAVs), also known as drones, are seen as an important feature of next generation wireless cellular systems [1]. Their unique capabilities such as maneuverability, flexibility, and adaptive altitude adjustment enable their diverse utilization, especially as aerial BSs to provide ubiquitous connectivity for the next generation of wireless networks¹. UAVs are widely used for military applications and are suited for the collection of real time information such as images and videos, which may be useful for strategic decision making. In recent years, UAVs have been used for many commercial and civil applications such as surveillance and traffic control, communication, disaster management, search and rescue operations, and land and border monitoring, among many others. UAVs can also be deployed in situations where a part of the communication infrastructure has been destroyed due to natural or man made disasters. UAVs can be deployed on demand whenever an unexpected surge of traffic demand occurs to offer high speed services such as dynamic coverage and edge computing.

For each application, the appropriate type of UAV has to be chosen to meet various requirements for the proposed application, the nature of the environment and federal regulations. Several factors need to be considered such as flying altitudes and the capabilities of the UAV such as energy resources, mobility, cost and coverage. In general, UAVs can be categorized, according to their altitude, into High Altitude Platforms (HAPs) and Low Altitude Platforms (LAPs). HAPs can fly at altitudes of up to 20 km with quasi stationary serving. On the other hand, LAPs can fly at altitudes of ten or hundred meters. They also have quick mobility and flexible deployment, making them more appropriate for time sensitive applications such as disaster situations or search and rescue operations. HAPs have longer endurance; hence, they can serve for longer periods of up to several months. LAPs can be categorized, based on their mobility, into fixed wing and rotary wing UAVs. Each of these has its unique advantages and disadvantages. Compared to rotary wing UAVs, fixed-wing UAVs such as small aircrafts are heavier, have higher velocity and cannot remain stationary. In contrast, rotary wing UAVs such as drones and quadcopter drones have a relatively low velocity and can hover and remain stationary over a given area. An overview on the different types of UAVs, their functions, and capabilities is presented in Fig. 1.1.

¹To further meet the requirements of connectivity of UAVs, the Third Generation Partnership Project (3GPP) recently considered the application of 5G networks in Release 17.

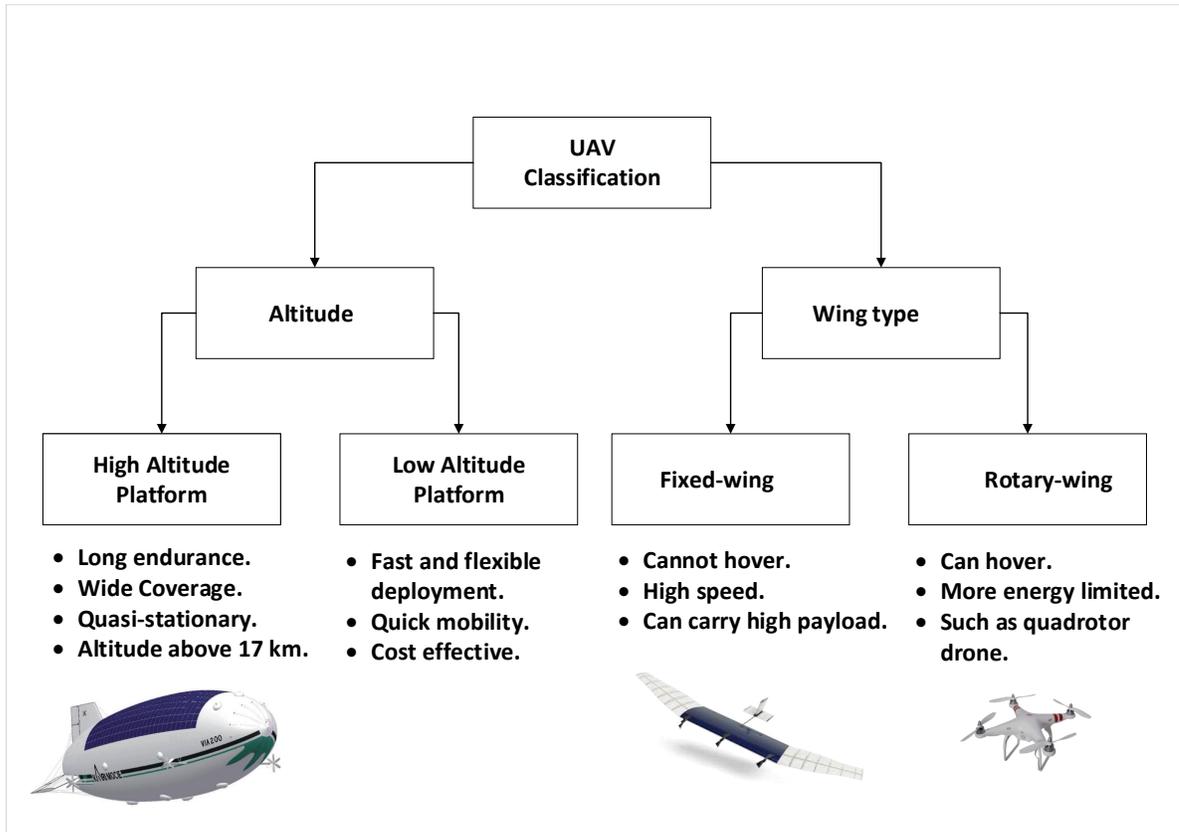


Fig. 1.1: UAV classification.

From the perspective of wireless communication and networking applications, UAVs can provide reliable and on demand flexibility for a variety of scenarios that will be discussed in this dissertation. On the one hand, drones can be used as flying edge servers to provide devices with low delay edge computing for tasks that involve relatively high computation in suburban and rural areas. On the other hand, UAVs can be used as aerial BSs that can deliver reliable, cost effective and on demand wireless communication to desired areas that are affected by a catastrophic disaster. Furthermore, the adjustable altitude of UAVs and their controllable trajectories enable them to effectively enhance communication by flying closer to the end users and establishing Line-of-Sight (LoS) links to significantly improve the Quality of Service (QoS).

The aforementioned advantages of UAVs demonstrate the high potential of their application in wireless networks. For instance, UAV-based aerial BSs can be deployed to provide effective communication coverage for ground users when traditional terrestrial infrastructure is partially or fully unavailable. Another important application of UAVs is in the Internet of Things (IoT) in which devices have limited computation resources and may not be able to meet the demands of applications such as gaming and augmented reality. In such scenarios, a UAV can be used for mobile edge

computing, where the computing resources in the network edge are employed to provide efficient and flexible computing services. UAVs can also be deployed to enhance wireless communication and provide QoS for a wide range of applications with heterogeneous requirements. In regions or countries where the construction of a complete cellular terrestrial infrastructure is economically infeasible due to high capital expenditure (CAPEX) and operational expenditure (OPEX), deploying UAVs is highly beneficial, as it eliminates the need for expensive towers and infrastructure deployment. Within the scope of these practical deployments, Qualcomm and AT&T are planning to employ LAP UAVs to enable wide scale communication in the upcoming 5G wireless networks [2].

Despite the numerous advantages and beneficial applications of UAVs, there are technical challenges that need to be tackled before the full potential of UAV communication can be realized. These challenges include three-dimensional (3D) deployment, trajectory design, communication resource allocation, performance analysis and air-to-ground channel modeling, among others. In this chapter, the potential use cases of UAV-based communication systems are presented. Then, the challenges associated with UAV communication are described followed by a review of limitations of existing literature on this subject. Finally the contributions of this dissertation are summarized and the resulting publications are listed.

1.2 Potential Use Cases of UAVs

1.2.1 On-Demand and Dynamic Aerial Networks

There are many regions that do not have access to wireless service or suffer from poor connectivity and QoS due to several limitations such as cost and geographical constraints such as the presence of a mountain or forest. Also, during major public events such as football matches, communication infrastructure experiences a substantial load surge in a very short period of time, that is, in the order of a few minutes. Therefore, the capacity and coverage of the existing cellular network infrastructure needs to be rapidly boosted in order to handle the high demand. In such scenarios, BSs mounted on flying UAVs (also known as aerial BSs, mobile BSs, or UAV-BSs) provide an effective solution to support the wireless communication networks. UAVs can be deployed at optimal altitudes at which they can provide maximum coverage or capacity for ground users.

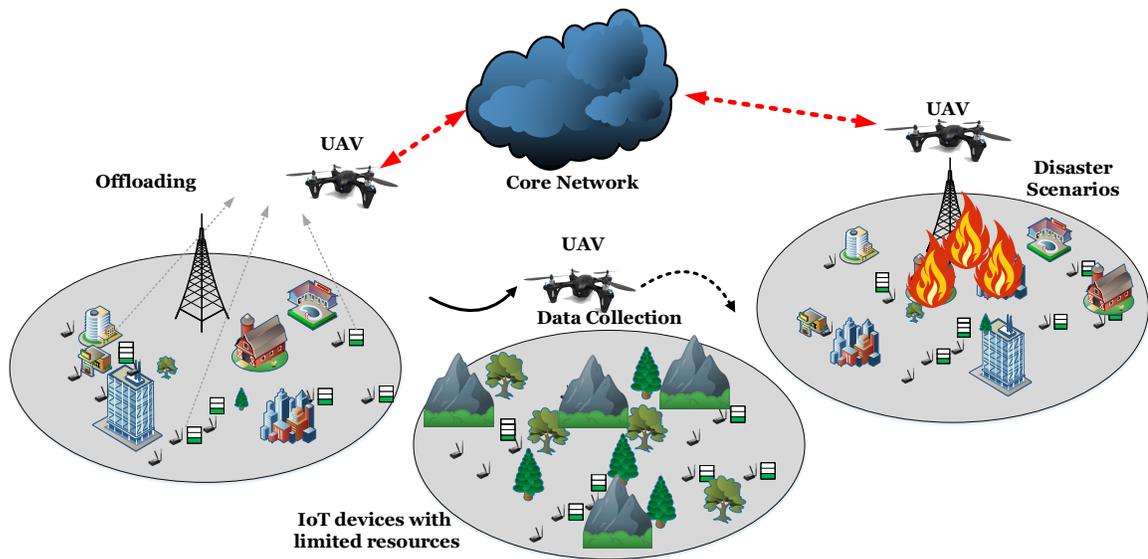


Fig. 1.2: Different use cases of UAVs.

1.2.2 UAV-Assisted Communication in Disaster Scenarios

Infrastructure-based communication networks tend to be susceptible to major damage arising from either natural catastrophic disasters (such as earthquakes, hurricanes, floods, lightning and tornadoes) or man-made disasters (such as wars, fire and explosion). Such events have the potential to damage or even destroy a country's communication infrastructure [3]. For instance, Hurricane Katrina, a major natural disaster that struck the Gulf Coast in 2005, disrupted the telecommunications infrastructure. More than 2000 cellular towers went out of service. More recently, Irma, a major hurricane that hit Florida in 2017, affected at least nine US states, ripping down power lines and cutting off connection in coastal communities. This event damaged about 20% of the communication sites in Puerto Rico and 55% in the US Virgin Islands [4] [5]. In such an event, people fleeing an affected zone to access a safe location would need to communicate with their family members or receive critical information (such as maps of unaffected highways and congestions) from rescue crews to guide their travel. To provide network connectivity during disasters, the use of aerial platforms (i.e., UAVs) is considered as a promising solution. Since UAVs can be quickly and efficiently deployed to provide support for cellular networks and enhance their Quality of Services.

1.2.3 UAV-Assisted IoT Communication

Governments and municipalities of major cities around the world pursue the vision of developing smart cities, which relies on information and communication technologies to gather information and to enable the efficient use of existing assets and resources. This would entail the use of a massive amount of network-connected devices such as wearables, smart home appliances, embedded sensors, traffic and street lights, connected vehicles and cameras within various sectors (such as health, transportation, energy and industrial), thus leading to the realm of IoE. Owing to the massive integration of IoT devices (IoTDs) into Information and Communications Technology (ICT) ecosystem, the sheer volume of data these devices generate, their diverse requirements in different sectors with respect to QoS (latency, reliability, higher rates and security), and their limited capabilities render current cellular systems unsuitable. UAVs can play a vital role in the context of smart cities with a dense deployment of sensors. UAVs can be used as a gathering entity for the acquired information from various IoTDs which have, for instance, limited communication capabilities. UAVs can also provide a computing hub at the edge to analyze the acquired data; therefore, low latency required by some IoT applications can be achieved. Owing to their high mobility, UAVs can move to enable LoS communication. They can come in close proximity to ground devices, leading to the achievement of higher throughput rates and energy conservation for less capable devices.

1.2.4 Aerial Edge Caching in Vehicular Networks

Edge caching is a promising technology to enhance end-users' QoS and to reduce backhaul multimedia traffic. However, caching at static ground BS or Road Side Units (RSUs) may not be effective in serving mobile users such as vehicles where frequent handovers occur. In this case, when a vehicle navigates to a new RSU, requested content may not be available and thus, the vehicle cannot be served properly. To better service vehicles in such cases, each requested content needs to be cached at multiple RSUs, but this is not efficient due to signaling overheads and additional storage usages. Hence, to enhance caching efficiency, UAV-assisted caching can provide higher QoS and multimedia data throughput through tracking vehicles' mobility and effectively deliver the required contents. In fact, UAVs can act as flying content providers and dynamically cache popular contents

Table 1.1: List of advantages and challenges of UAV-assisted wireless networks.

Advantages	Challenges
Coverage and capacity enhancement	Optimal deployment (placement and trajectory).
LoS connection.	Channel modeling.
Internet of Things support.	UAV Backhauling.
On-demand communications.	Flight time and energy constraints.
Dynamic deployment.	Resource management.
No significant infrastructure.	Safety and security.
Fully-controlled mobility.	Minimal knowledge of the environment.

in order to provide either paid streaming services (such as Netflix) or public streaming service (such as YouTube). This is a very plausible scenario since cellular networks (5G and beyond) may become overloaded and operators may opt to offload their networks by exploiting UAVs.

In summary, only a few potential use cases of UAVs in wireless communication networks have been highlighted, as shown in Fig. 1.2. In spite of these promising aspects of UAVs in wireless networks, there are several technical challenges that require significant efforts from researchers to propose efficient solutions for UAV deployment while considering the different constraints of UAVs and environments. Table 1.1 summarizes the advantages and challenges of UAV-based communications.

1.3 Motivations and Contributions

1.3.1 Limitations of Existing Works

In this subsection, the main limitations in existing studies on UAV communication are presented in order to define the research objectives of this thesis.

- (1) Despite the notable amount of research on trajectory planning of UAVs for data collection missions, a holistic view still cannot be provided on how to collect data from IoT devices under explicit flight time and restricted time constraints. For instance, during a natural disaster, crucial and specific data need to be collected in time for systematic evaluations of the current situation in the affected area. The timeliness of the transmitted data for these scenarios is

essential since outdated data may have no useful value. Therefore, the problem of timely data collection from IoT devices needs to be addressed, where the collected data has deadlines and needs to be collected before the data loses its meaning or becomes irrelevant. To the best of our knowledge, finding the most suitable trajectory for the UAV in order to maximize the number of served IoT devices, where each device has its own target data upload deadline, is a subject that remains unaddressed. Furthermore, the most relevant existing research on data collection using UAV assumes that A2G links are dominated by LoS channels and neglect small scale effects. Despite the dominance of LoS links, completely neglecting small scale fading is an oversimplification.

- (2) The most relevant existing research on UAV communication focuses on the design of the trajectory of UAVs to achieve high throughput, maximum coverage, or low latency. However, these performance metrics may not capture the freshness of information, which is required for real-time update applications. The freshness of collected information is quantified by a new performance metric, the age-of-information (AoI) or status age. To the best of our knowledge, very limited research exists on the analysis of AoI in the design of UAV trajectory for data sensing of real time tasks. There is therefore a need to develop novel solutions that leverage UAVs and machine learning to address challenges associated with AoI such as scheduling, UAV deployment, unreliability of communication, and uncertainty of networks.
- (3) There have been recent studies to address various challenges in the integration of UAVs with Reconfigurable Intelligent Surface (RIS) in order to assist wireless communication networks. However, the analysis, schemes and algorithms developed in these studies may not necessarily be optimal from the perspective of preserving freshness of information. This is because in these studies, the configuration of RIS and the deployment of UAVs were designed based on either maximizing the network spectral/energy efficiency or minimizing the latency or outage probability. To the best of our knowledge, no study has addressed the optimization of RIS configuration while considering the freshness of information.
- (4) If future networks decide to deploy UAVs in order to assist vehicular networks, UAVs' energy budget and the uncertainty of the newly arriving vehicles would present a critical challenge.

Some studies have investigated energy efficiency of UAVs, but they remain largely limited in scope since they do not analyze energy efficiency in vehicular networks. In particular, optimizing the performance of UAV assisted vehicular networks under the energy budget constraint of UAVs and in the absence of a complete knowledge of the environment has not been addressed, to the best of our knowledge.

- (5) UAVs empowered with Artificial Intelligence (AI) have been substantially studied from a robotics/control perspective. However, only a few studies consider UAVs to be wireless-equipped. These studies do not analyze the interrelationship between mobility and wireless QoS. Existing literature has also not studied the application of UAVs in vehicular networks while considering the continuous trajectory nature of UAVs such as the exploit of UAVs as aerial BS. In order to effectively use UAVs as flying BS, the continuous trajectory of UAVs needs to be optimized with respect to wireless performance metrics such as coverage and freshness of information. More specifically, studying the continuous trajectory of UAVs empowered with AI in the vehicular network is a subject that remains, to the best of our knowledge, unexplored.
- (6) While the performance of UAVs acting as mobile content providers has been studied in static environments, these results cannot readily extend to cases in highly dynamic environments, such as vehicular networks, as is envisioned for future networks. In such networks, vehicles usually move at a relatively high speed, causing the topology of the vehicular network to change occasionally. Hence, a novel design consideration must be considered. In particular, determining the minimum number of UAVs and their optimal trajectories, while considering the dynamic nature of the vehicular network, is a problem that remains, to the best of our knowledge, unaddressed.

1.3.2 Thesis Contributions

This dissertation focuses on UAVs' deployment in wireless communication networks. The main contribution of this dissertation is to propose various novel frameworks that fall within the vision of future wireless networks in realizing the realm of pervasive and intelligent high speed connectivity,

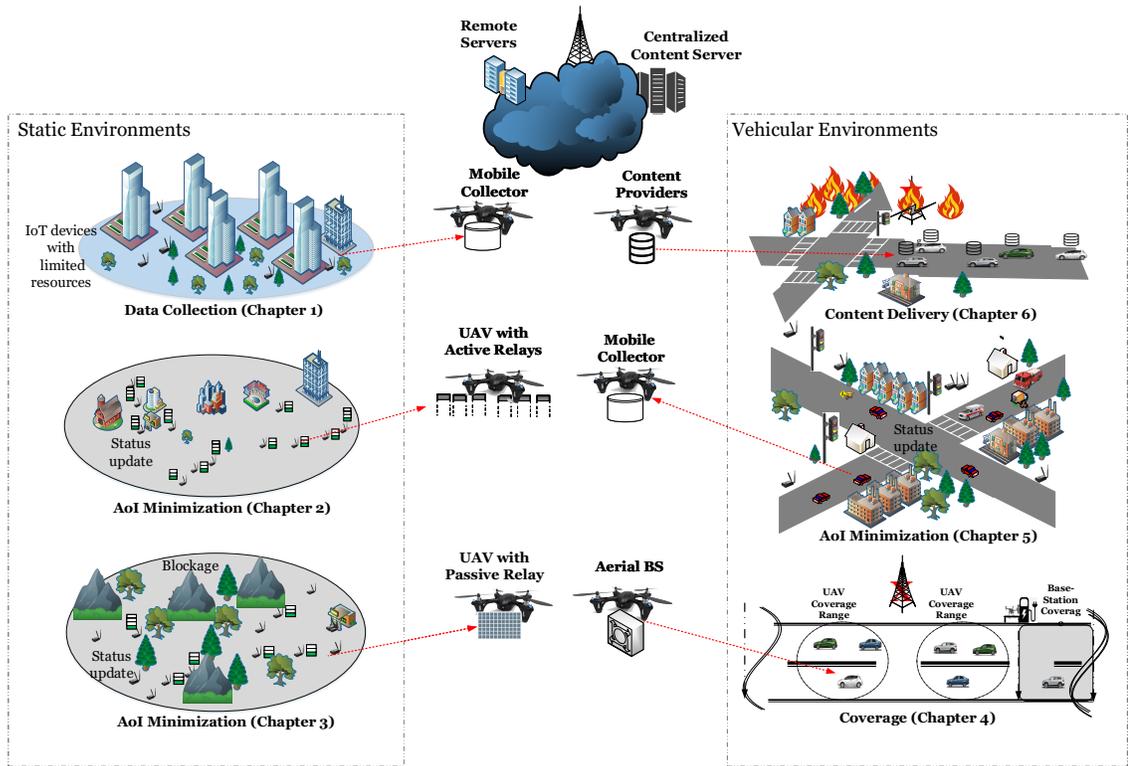


Fig. 1.3: Graphical summary of addressed problems in future wireless networks.

all through leveraging unmanned aerial vehicles. By using the proposed frameworks, the performance of UAV-based communication systems can be optimized under various metrics in terms of coverage, throughput and age-of-information. Optimization here also takes into account the unique features of UAVs such as their flight time, mobility, energy budget, and altitude. This dissertation weaves together notions from optimization theory, machine learning, traffic theory and probability. The use of such advanced mathematical tools enables the development of an in-depth analytical foundation and efficient algorithms in order to design, optimize, deploy and operate UAV-based communication. Moreover, this dissertation addresses different problems, which can be classified into two different contexts: 1) Efficient deployment of UAV-assisted static environments and 2) Efficient deployment of UAV-assisted high mobility environments. A graphical summary of the various addressed problems in this dissertation is presented in Fig. 1.3. In the sequel, each of the problems is introduced and the main findings are highlighted.

Optimized UAV Trajectory Planning in IoT Networks for Maximum Service

In Chapter 2, the optimal trajectory for a UAV is investigated in order to maximize the possible number of served IoTDs while ensuring a minimum amount of data upload per device in a realistic channel model. To achieve this, the trajectory of a UAV and the radio resource allocation are both optimized in order to maximize the number of served IoTDs, where each IoTD has its own target data upload deadline. This is modeled mathematically as a mixed-integer non-convex optimization problem. An optimal solution that follows a well-designed Branch, Reduce and Bound (BRB) algorithm is developed. However, given its complexity and lack of scalability, a low-complexity method (based on Successive Convex Approximation (SCA) method) is developed, where a trajectory that maximizes the number of served IoTDs is first identified. Then, the UAV is deployed along the designed trajectory and at each time slot, it collects accurate Channel State Information (CSI) knowledge in order to allocate radio resources that serve the IoT devices. In addition, more insight is provided on a method that further optimizes the trajectory (that is, finds the shortest trajectory) to serve the same number of IoTDs within their information deadlines. Finally, results are compared with two greedy methods as benchmarks based on distance and deadline metrics. Under variable deadlines and minimum service amounts, the proposed solution outperformed alternative solutions including static UAV placement and distance- and deadline-based greedy approaches in terms of the percentage of served IoTDs (average improvement of 10 % – 50 %).

UAVs as Active Relays in IoT Networks for Fresh Information

As discussed in Section 1.2, UAVs can play a significant role in IoT communication by relaying data from devices with limited transmission capabilities such as sensors to the nearest BS. In Chapter 3, UAVs as mobile relays from ground IoTDs are investigated, where low resource IoTDs periodically sample a stochastic process and need to upload more recent information to a BS. Numerous emerging applications rely on freshness of sensory data (i.e., status-updates) which is being monitored and generated by a plethora of IoTDs. Outdated updates may be inconsistent with the current status of the physical process being monitored and controlled, which may lead to erroneous decisions. To enable reliable uplink communications, UAVs with virtual queues are deployed as a

middle layer between IoT devices and the BS. In the absence of channel conditions, the optimal scheduling policy is investigated as well as the dynamic UAV altitude control that maintains a fresh status of information at the BS. The objective of this chapter is to minimize the Expected Weighted Sum Age of Information (EWSA) for IoT devices. First, the problem is formulated as an optimization problem that is however generally hard to solve. Second, a model free Deep Reinforcement Learning (DRL) with a central agent is proposed, where the deployed UAV obtains instantaneous CSI in real time along with any adjustment to its deployment altitude. Third, the online problem is formulated as a Markov Decision Process (MDP) and a highly stable state-of-the-art DRL algorithm, Proximal Policy Optimization (PPO), is leveraged to solve the formulated problem. Finally, extensive simulations are conducted to verify findings and comprehensive comparisons with other baseline approaches are provided to demonstrate the effectiveness of the proposed design.

UAV-mounted RIS as Passive Relays in IoT Networks for Fresh Information

Unlike Chapter 3, where UAVs act as active mobile relays, integrating RIS with UAVs proves to be beneficial in several ways. Data transmission from IoT devices to the BS through RIS empowered UAVs requires less intermediate delays compared to UAVs acting as mobile active relays. Also, power consumption, due to the processing of the relayed information at the UAVs, can be avoided, leading to an increase in flight endurance of the UAV. In Chapter 4, a new relaying system that integrates UAVs and RIS is proposed to maintain the freshness of information of remote IoT networks. The altitude of the UAV, transmission scheduling, and phase shift matrix of RIS elements are optimized to minimize the expected sum of AoI. To tackle this mixed-integer non-convex problem, PPO algorithm is proposed. Numerical results demonstrate that the proposed algorithm can significantly minimize AoI compared to other baselines such as random walk and heuristic greedy algorithms.

Learning-based Trajectory Planning of Aerial Base Stations in Vehicular Networks

In order to reap the benefits of UAV in wireless networking applications, it is necessary to study the use of UAVs in vehicular networks. In particular, the characteristics of a dynamic environment such as vehicular networks pose a unique design challenge for UAV assisted vehicular networks.

For instance, the high mobility of the vehicles leads to distinct network conditions and changes the network topology, both of which should be carefully handled in order to guarantee the performance of the UAV system. In Chapter 5, coverage analysis is provided for UAV assisted vehicular networks in the absence of a complete knowledge of the environment. This subject adds to the novelty of this dissertation since existing research in literature focuses mainly on the optimization of the trajectories of UAVs in static environments with a complete knowledge of the environment. The complex and dynamic network conditions heightens the difficulty to design an efficient approach to trajectory optimization. Due to varying traffic conditions and the uncertainty of the vehicular network, a machine learning approach is utilized to govern the required number of UAVs and their trajectories in order to serve existing and newly arriving vehicles. Efficient trajectory planning of UAVs requires addressing a number of key challenges such as the minimum number of UAVs and their trajectories needed to provide an effective coverage for a given highway segment that is not covered by ground BS or Road Side Units (RSUs). To address this challenge, decision making for trajectories is formulated as a Markov Decision Process where the system state space considers vehicular network dynamics. Then, DRL is leveraged to propose an approach to learn the optimal trajectories of the deployed UAVs. Actor-Critic algorithm is adopted to efficiently maximize vehicular coverage and learn the vehicular environment and its dynamics to handle the complex continuous action space. Simulations results are then presented to verify findings and demonstrate the effectiveness of the proposed design. During the mission time, the deployed UAVs adapt their velocities in order to cover the vehicles.

AoI-Aware Data Collection in Vehicular Networks with Intelligent UAVs

Traditional UAV assisted vehicular networks have adopted coverage, throughput, and latency as performance metrics. These metrics, however, are not adequate to reflect the freshness of the information, an attribute that has been recently identified as a critical requirement to enable services such as autonomous driving and accident prevention. In Chapter 6, data collection analysis for UAV assisted vehicular network is provided, wherein sensors (such as LiDARs and cameras) on vehicles generate time sensitive data streams and UAVs are used to collect and process this data while maintaining a minimum AoI. This chapter develops a new framework for optimizing the trajectories

of UAVs and finding scheduling policies to keep the information fresh under minimum throughput constraints. The formulated optimization problem is shown to be mixed-integer, non-convex, and generally hard to solve. Motivated by the success of machine learning techniques, particularly Deep Learning, in solving problems with low complexity, the trajectories and scheduling policies are reformulated as a Markov Decision Process. Then, Deep Deterministic Policy Gradient is leveraged to learn the trajectories of the deployed UAVs and to efficiently minimize the EWSA. Simulations results demonstrate the effectiveness of the proposed design and show that the deployed UAVs adapt their velocities during the data collection mission in order to minimize the AoI.

On-Demand Content Delivery in Vehicular Networks with Optimized Multi-UAV Trajectories

As discussed in Section 1.2, UAVs are recommended as promising solutions to provide fast network recovery when infrastructure is temporarily unavailable. In Chapter 7, several scenarios that explore the use of UAVs as mobile content providers are analyzed, including, but not limited to: i) the delivery of critical data to all vehicles (for example, warnings about speed curves) for dynamic path planning (for example, warning about collisions) where the infrastructure is destroyed or unavailable; ii) the use of UAVs, as flying content providers, to dynamically cache popular content in order to provide streaming services, as explained in subsection 1.1.4; iii) the use of UAVs in the context of Self-Organized-Networks (SON) where operators, who need to deliver certain popular content (for example, active streaming of ongoing events), do not have available infrastructure (due to high CAPEX) or are not interested in integrating additional infrastructure since it may only be used for a short period of time and thus does not justify the cost of deployment. Thus, in Chapter 7, the number of UAVs must be taken into account in the analysis of UAV-based communication systems in vehicular networks. In this case, a framework is needed to analyze and optimize the performance of UAV-based communication in vehicular networks based on the number of deployed UAVs. Therefore, in Chapter 7, a framework is developed to minimize the number of deployed UAVs to fully serve all vehicles. The trajectory of a UAV and the radio resource allocation are both optimized in order to minimize the number of deployed UAVs within a given time frame while guaranteeing the vehicles' requirements. The formulated problem is shown to be non-convex and generally hard to be solved. To solve it, successive convex approximation based method is employed

to approximate the original non-convex problem through a sequence of its convex approximates. Then, an efficient low complexity algorithm is developed to sequentially solve this convex approximated problem until convergence. Numerical results demonstrate the effectiveness of the proposed design and show that during the mission time and to fulfill the requirements of each vehicle, the UAVs adapt their velocities, as well as their directions, to the velocities of the current and incoming vehicles.

1.4 List of Publications

This dissertation has led to the following key publications:

1.4.1 Journal Publications

- (1) **Moataz Samir**, Dariush Ebrahimi , Sanaa Sharafeddine, Chadi Assi, and Ali Ghrayeb, "Age of Information Aware Trajectory Planning of UAVs in Intelligent Transportation Systems: A Deep Learning Approach", *IEEE Transactions on Vehicular Technology*, vol. 69, no. 11, pp. 12382-12395, Nov. 2020.
- (2) **Moataz Samir**, Sanaa Sharafeddine, Chadi Assi, and Ali Ghrayeb, "Online Altitude Control and Scheduling Policy for Minimizing AoI in UAV-assisted IoT Wireless Networks", *IEEE Transactions on Mobile Computing*, Accepted, Dec., 2020.
- (3) **Moataz Samir**, Dariush Ebrahimi , Sanaa Sharafeddine, Chadi Assi, and Ali Ghrayeb, "Leveraging UAVs for Coverage in Cell-Free Vehicular Networks: A Deep Reinforcement Learning Approach", *IEEE Transactions on Mobile Computing*, Accepted , April, 2020.
- (4) **Moataz Samir**, S. Sharafeddine, C. Assi, T. Nguyen and A. Ghrayeb, "UAV Trajectory Planning for Data Collection from Time-Constrained IoT Devices," in *IEEE Transactions on Wireless Communications*, vol. 19, no. 1, pp. 34-46, Jan. 2020.
- (5) **Moataz Samir**, S. Sharafeddine, C. Assi, T. Nguyen and A. Ghrayeb, "Trajectory Planning and Resource Allocation of Multiple UAVs for Data Delivery in Vehicular Networks," in *IEEE Networking Letters*, vol. 1, no. 3, pp. 107-110, Sept. 2019.

- (6) **Moataz Samir**, D. Ebrahimi, C. Assi, S. Sharafeddine and A. Ghayeb, "Trajectory Planning of Multiple Dronecells in Vehicular Networks: A Reinforcement Learning Approach," in IEEE Networking Letters, vol. 2, no. 1, pp. 14-18, March, 2020
- (7) **Moataz Samir**, Mohamed Elhattab, C. Assi, S. Sharafeddine and A. Ghayeb, "Optimizing Age of Information Through Aerial Reconfigurable Intelligent Surfaces: A Deep Reinforcement Learning Approach," in IEEE Transactions on Vehicular Technology, submitted, 2020.
- (8) Ahmed Al-Hilo, **Moataz Samir**, Chadi Assi, Sanaa Sharafeddine, Dariush Ebrahimi, "A Co-operative Approach for Content Caching and Delivery in UAV-Assisted Vehicular Networks", in IEEE Transactions on Intelligent Vehicles, submitted, 2020.
- (9) Ahmed Al-Hilo, **Moataz Samir**, Chadi Assi, Sanaa Sharafeddine, Dariush Ebrahimi, "UAV-assisted Content Delivery in Intelligent Transportation Systems - Joint Trajectory Planing and Cache Management," IEEE Intelligent Transportation Systems Transactions, Accepted, August, 2020.
- (10) Ali Muhammad, Ibrahim Sorkhoh, **Moataz Samir**, Dariush Ebrahimi and Chadi Assi, "Minimizing Age of Information in Multi-Access Edge Computing-assisted IoT Networks," IEEE Internet of Things Journal, submitted, Dec., 2020.

1.4.2 Conference Publications

- (1) **Moataz Samir**, M. Chraiti, C. Assi and A. Ghayeb, "Joint Optimization of UAV Trajectory and Radio Resource Allocation for Drive-Thru Vehicular Networks," 2019 IEEE Wireless Communications and Networking Conference (WCNC), Marrakesh, Morocco, 2019, pp. 1-6.
- (2) Ahmed Al-Hilo; **Moataz Samir**; Chadi Assi; Sanaa Sharafeddine; Dariush Ebrahimi, "Cooperative Content Delivery in UAV-RSU Assisted Vehicular Networks," ACM DroneCom 2020, Accepted, 2020.

Chapter 2

Optimized UAV Trajectory Planning in IoT Networks for Maximum Service

2.1 Background, Related Works, and Contributions

As previously discussed in Chapter 1, unmanned aerial vehicles (UAVs) have recently received much attention and been explored among the enabling and supporting technologies for 5G wireless systems and beyond. Indeed, UAVs can play a central role in the context of smart cities with dense deployment of sensors; UAVs can be used as a gathering entity of the collected information from various IoT devices (IoTDS) with for instance limited communication capabilities. UAVs can also provide a computing hub at the edge to run several data analytics on the collected data, therefore achieving the low latency required by several critical IoT applications. Owing to their mobility, UAVs can flexibly move to enable a Line-of-Sight (LoS) communication or come in close proximity to the ground devices, therefore, achieving higher throughput rates and conserving the energy of less capable devices. In summary, the benefits UAVs bring to current networks are enormous, and as such, they are considered among the contending enabling technologies for building networks of the future.

The work done in this chapter leads to an IEEE published journal [6]

In this chapter, we address the problem of data collection, where a deployable UAV (for instance in the context of offloading) can be dispatched to gather data collected by IoTDS in a smart city environment. In particular, *timely* data collection becomes very critical in scenarios that involve IoTDS with limited buffer sizes deployed for instance for continuous measurements, and thus data has to be extracted before it loses its value or being overwritten by newly incoming data. Other scenarios include situations (such as in emergency rescue operations, disaster monitoring and target tracking) where the accumulated data reveals current conditions of the respective field for alert and notification services and thus enjoys a restricted lifetime beyond which it loses its significance. For instance, during a natural disaster, specific vital data needs to be collected in time for systematic evaluations of the current situation in a given area. The timeliness of the transmitted data for these scenarios is essential, since outdated data may have no useful value. The existing literature has addressed various challenges in UAV communication systems. Optimizing the trajectory of the UAV is one of the important research challenges. In particular, optimizing the trajectory of the UAV depends on many factors. For instance, the work in [7], maximized the minimum rate among ground users by optimizing the trajectory and user scheduling for a single-UAV. In [8], the authors characterized the capacity region of a UAV-enabled two-user broadcast channel by optimizing the UAV trajectory jointly with transmit power or rate. The authors showed that for a sufficiently long flight duration, the optimal UAV trajectory with different multiple access schemes will achieve almost the same capacity. In [9], the authors characterized the capacity region of a UAV for multiple users by jointly optimizing the UAV trajectory and radio resource allocation for multiple access techniques. The authors showed that the capacity region achieved for multiple users by non-orthogonal multiple access significantly outperforms the rate regions by orthogonal multiple access, while frequency division multiple access achieves higher rate region than that by time-division multiple access. The work in [10] jointly optimized the trajectory, multi-user scheduling and power control for multiple UAVs to maximize the minimum rate of ground users. In [11], the authors optimized the UAV's trajectory to minimize the time to completely disseminate a common file to a number of distributed ground terminals. In [12], the UAV trajectory, bandwidth resources, and user partitioning between a ground Base-Station (BS) and UAV are optimized to maximize the minimum quality of service

(QoS) for ground users located at the cell edge. In [13], UAV trajectory and ground terminal transmit power are jointly optimized for both circular and straight trajectories to reveal a fundamental trade-off between the UAV propulsion energy consumption and ground terminal communication energy consumption. In [14], the authors maximized the throughput by optimizing the UAV trajectory jointly with the transmitted power for a mobile relay node mounted on a UAV, subject to the UAV mobility constraints. In [15], the UAV placement, radio resource allocation, and decoding order of the non-orthogonal multiple access transmission scheme are optimized to maximize the sum achievable rate of all users. In [16], the authors applied successive convex approximation (SCA) and the Lagrange duality to maximize the minimum average rate by optimizing the trajectory of UAV and spectrum allocation.

On the other hand, data collection using UAVs has been addressed in several prior work. For example, the authors in [17] proposed a data collection framework for UAV-assisted wireless system to maximize the system throughput. To increase the efficiency of data collection and increase the sensors' lifetime, the authors proposed a priority access and routing algorithm framework upon dividing the sensors into multiple groups, each associated with a certain priority. The authors of [18] optimized the UAV's trajectory and sensors' wake-up schedule to minimize the maximum energy consumption of all sensors to increase the network lifetime. The authors applied successive convex approximation, SCA method, to solve the optimization problem sub-optimally. Multiple UAVs are also considered in the same work while considering a fading channel. In [19], the authors deployed multiple UAVs for collecting data from ground IoT devices, where the total uplink transmit power of these IoT devices is minimized in a time-varying network by optimizing the UAV's trajectory and IoT power control. In [20], the authors proposed a greedy algorithm to optimize the trajectory of the UAV to minimize the mean square error for estimation parameters by sensor nodes. In [21], the authors proposed a solution for energy-efficient data collection by optimizing the trajectory of UAV jointly with optimized the selection of cluster head along with establishing forwarding trees between sensor-nodes and cluster head.

Recently, few works have been conducted to address the time-sensitive data collection. The authors in [22] proposed two UAV trajectory to minimize the maximum and average age-of-information. The authors adapt a dynamic programming (DP) method and genetic algorithm (GA) to obtain the

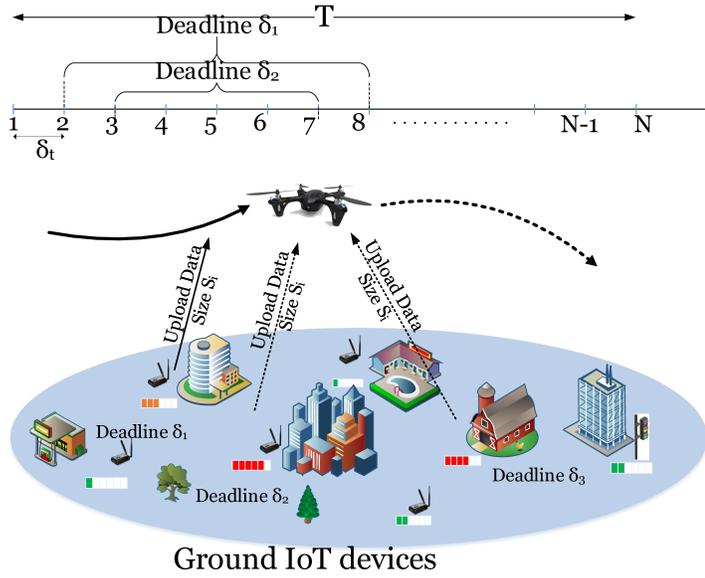


Fig. 2.1: System model: timely data collection in a smart city environment using UAV. For illustration, an example of the time line representation for two IoTDS with two deadlines are presented.

UAV trajectory. The authors in [23], optimized the UAV trajectory, service time allocation and the UAV energy to minimize the average peak age-of-information between a source and destination, where an iterative algorithm is proposed to solve the optimization problem. In this work, we are interested in further exploring the impact of the deadlines on the trajectory of UAV and the allocation of radio resources. Specifically, we aim to jointly optimize the trajectory of a UAV and the radio resource allocation when imposing a deadline on data packets that need to be collected before expiry.

Compared to the surveyed related work, here we address the problem of timely data collection from IoTDS where the collected data has deadlines and needs to be collected before the data loses its meaning or becomes irrelevant. Moreover, we adopt a Rician channel model, which encompasses a wider range of channel models, and hence makes the proposed solution more realistic. Our objective is to find the most suitable trajectory for a UAV to collect data from the maximum possible number of devices while ensuring a minimum amount of data uploaded per device. This turns out to be a challenging problem, which we model mathematically as a non-convex optimization problem. We develop an optimal solution following a well designed branch, reduce and bound (BRB) algorithm.

However, given its complexity and lack of scalability, we develop a low complexity method (based on SCA method) where we first find *a trajectory* that maximizes the number of served devices. Then, the UAV is deployed along the designed trajectory and at each time-slot collects accurate channel state information (CSI) knowledge to allocate radio resources to serve the IoTDS. Next, we elaborate a method that further optimizes the trajectory (i.e., find the shortest) for serving the same number of IoTDS within their information deadlines. Finally, we compare our results with two greedy methods as benchmarks based on distance and deadline metrics.

The rest of this chapter is organized as follows. Section 2.2 presents the system model followed by the problem formulation in section 2.3. We propose an optimal solution for data collection in section 2.4. A sub-optimal solution along with enhanced algorithm are proposed in section 2.5. Section 2.6 proposes enhanced algorithm for minimizing UAV flight distance. Simulation results are presented in section 2.8. Finally, conclusions are drawn in section 2.9, and future research directions are highlighted.

2.2 System Model

We consider a smart city environment comprising a set \mathcal{M} of M IoTDS with limited capabilities distributed over a given area and continuously collecting time-sensitive data. This data is assumed to carry useful information as long as it is uploaded within a given target deadline, beyond which it loses its significance and becomes irrelevant. The system model is depicted in Fig. 2.1, where a UAV is dispatched on a regular basis to serve as many IoTDS as possible by completely collecting information from each device i before its expiry deadline δ_i . The locations of the IoTDS together with their corresponding data sizes, the data generation time, τ_i , and target deadlines are assumed to be known by the UAV, through a central controller, prior to the launch of the UAV for every data collection mission. The mission duration, referred to as flight time, is fixed to T and divided into N equal time-slots, indexed by $n = 1, \dots, N$, each of length δ_t . Technically, δ_t is sufficiently small such that we can assume the location change of the UAV within δ_t is negligible, compared to the distances from all IoTDS to the UAV.

The UAV is assumed to fly at a fixed altitude H in meters above ground level, e.g., that is

imposed by the regulatory authority for safety considerations; the UAV's location in time-slot n is given by (x^n, y^n, H) . Orthogonal transmission is employed in the uplink to allow multiple IoTDS to simultaneously upload their data to the UAV. Given the location $(x_i, y_i, 0)$ of each IoTDS $i \in \mathcal{M}$ at ground level and the current UAV location (x^n, y^n, H) in time-slot n , the distance d_i^n between the UAV and the IoTDS is calculated as follows:

$$d_i^n = \sqrt{(x_i - x^n)^2 + (y_i - y^n)^2 + H^2}, n = 1, 2, \dots, N. \quad (2.1)$$

In practice, the speed of a UAV is limited to a maximum value v_{max} in m/s and, thus, its travel distance in one time-slot is constrained as follows:

$$(x^{n+1} - x^n)^2 + (y^{n+1} - y^n)^2 \leq (v_{max} \delta_t)^2, n = 1, \dots, N - 1. \quad (2.2)$$

We also assume the channel between the IoTDSs and the UAV follows a Rician fading channel model with a factor K , where the channel coefficient h_i^n can be written as

$$h_i^n = \widehat{h}_i^n \Delta_i^n, \quad (2.3)$$

where \widehat{h}_i^n and Δ_i^n respectively represent the small-scale fading and path-loss coefficients. In particular, we can write the path-loss coefficient as $\Delta_i^n = \gamma_0 (d_i^n)^{-\alpha}$, where γ_0 is the average channel power gain at a reference distance $d_0 = 1m$, α is the path-loss exponent that usually has a value greater than 2 for Rician fading channel. The small scale fading \widehat{h}_i^n is composed of LoS component \bar{h}_i^n , where $|\bar{h}_i^n| = 1$, and a random Non-line-of-sight (NLoS) component \widetilde{h}_i^n , where $\widetilde{h}_i^n \sim \mathcal{CN}(0, 1)$. The small scale fading \widehat{h}_i^n is given by

$$\widehat{h}_i^n = \left(\sqrt{\frac{K}{K+1}} \bar{h}_i^n + \sqrt{\frac{1}{K+1}} \widetilde{h}_i^n \right). \quad (2.4)$$

Each IoTDS i is assumed to transmit with constant power P leading to a received power at the UAV $P_i^n = |h_i^n|^2 P$ in time-slot n . The signal-to-noise ratio (SNR) of each IoTDS is $\Upsilon_{i,n} = P |\widehat{h}_i^n|^2 \Delta_i^n / \sigma^2$, where σ^2 is the thermal noise power which is linearly proportional to the allocated

bandwidth [24]. Thus, the achievable rate for each IoTD i in time-slot n is given by

$$r_i^n(b_i^n, x^n, y^n) = b_i^n \log_2(1 + \Upsilon_{i,n}), \quad (2.5)$$

where b_i^n is the fraction of spectrum allocated to IoTD i in time-slot n and it is equivalent to a number of resource blocks. In practice, for large numbers of resources, b_i^n is approximately continuous between 0 and 1. Thus, the allocation of the radio resources should meet the below constraints

$$\sum_{i \in \mathcal{M}} b_i^n \leq 1, \forall n, \quad (2.6)$$

$$0 \leq b_i^n \leq 1, \forall n, i \in \mathcal{M}. \quad (2.7)$$

We should note that our model assumes a frequency non selective, or flat, channel which, unlike frequency selective channel, only the fraction of radio spectrum allocated to each IoTD is of interest, rather than which fraction of the radio spectrum. Hence, our allocation constraint decides on the amount of resource blocks that need to be allocated to achieve the service amount for each served device.

We define the service amount as the amount of data that one IoTD delivers to the UAV within a given deadline during a data collection mission. The service amount concept has been proposed in multiple previous works especially in scenarios with mobility [25–27], where the achievable rate is time-variant and does not exhibit the service quality of the corresponding transmitting device. Similarly, in our system model, the achievable rate of one IoTD is not only based on the device itself but varies according to the data deadlines of the other IoTDs to be served. Consequently, we utilize the service amount concept to represent the service quality of each IoTD. The service amount $S_i(b_i^n, x^n, y^n)$ provided by each IoTD i over flight time T can be computed based on the summation of the achievable rates throughout the information lifetime, where the rate of a given device is set to 0 as soon as its data deadline passes. The service amount $S_i(b_i^n, x^n, y^n)$, computed in bits/Hz, can be written as

$$S_i(b_i^n, x^n, y^n) = \delta_t \sum_{n=1}^N s_i^n, \forall i \in \mathcal{M}, \quad (2.8)$$

where

$$s_i^n = \begin{cases} r_i^n(b_i^n, x^n, y^n), & \text{if } \tau_i \leq n \leq \delta_i, \\ 0, & \text{otherwise.} \end{cases} \quad (2.9)$$

2.3 Problem Formulation

The objective of this work is to optimize the UAV trajectory and allocation of resources to maximize the total number of served IoTDs within a flight mission duration T based on a given set of target time constraints. To serve device i , its data S_i should be completely collected by the UAV throughout the lifetime. To mathematically formulate the problem, we define a binary variable $\kappa_i \in \{0, 1\}$, $\forall i \in \mathcal{M}$, that is asserted if the UAV can successfully serve device i with a minimum service amount S_i^{\min} ; otherwise, it is set to 0, where S_i^{\min} is defined as the minimum amount of information (bits/Hz) that need to be uploaded by device i . Let us denote $\mathbf{X} = \{x_n, \forall n\}$, $\mathbf{Y} = \{y_n, \forall n\}$, $\mathbf{K} = \{\kappa_i, i \in \mathcal{M}\}$ and $\mathbf{B} = \{b_i^n, i \in \mathcal{M}, n\}$. The formulated optimization problem is given in (2.10) with the objective to maximize the number of served IoTDs.

$$(\mathcal{P}_1) : \max_{\mathbf{X}, \mathbf{Y}, \mathbf{B}, \mathbf{K}} \sum_{i \in \mathcal{M}} \kappa_i \quad (2.10a)$$

$$\text{s.t. } S_i(b_i^n, x^n, y^n) \geq \kappa_i S_i^{\min}, \forall n, i \in \mathcal{M}, \quad (2.10b)$$

$$\kappa_i \in \{0, 1\}, i \in \mathcal{M}, \quad (2.10c)$$

$$0 \leq b_i^n \leq \kappa_i, \forall n, i \in \mathcal{M}, \quad (2.10d)$$

$$(2.2), (2.6), \quad (2.10e)$$

$$[x^0 \ y^0] = [x_s \ y_s], \quad (2.10f)$$

$$[x^N \ y^N] = [x_e \ y_e]. \quad (2.10g)$$

Constraint (2.10b) guarantees that each served IoTD uploads the minimum amount of data S_i^{\min} . Constraint (2.10d) prevents the UAV from wasting radio resources on IoTDs that cannot be served within their deadline. As a result, the share resources b_i^n in Eq. 2.7 is upper bounded by κ_i that is set to 0 if device i is not selected to be served. Constraints (2.10f) and (2.10g) indicate the

initial position of the UAV's trajectory located at $[x_s \ y_s]$ and the final position at $[x_e \ y_e]$. In fact, the operator may decide on those positions based on multiple factors such as the location of their managed property, legislation and/or UAV's charging stations.

Clearly, the solution of (\mathcal{P}_1) , which yields a trajectory for the UAV during a time frame T , relies on the knowledge of the instantaneous channel at each time-slot during the flight. Given that by the time a trajectory is designed, there is no possible way of obtaining the channel conditions in future slots; hence, we overcome this obstacle by assuming a path loss model for the channel and solve (\mathcal{P}_1) to obtain a trajectory which maximizes the number of IoTs which can be served. Then, we utilize the obtained trajectory to fly the UAV; but, at each time-slot during its flight, the UAV obtains correct instantaneous CSI, and then assigns resources for the IoTs to meet their service rate, at that slot. The process is repeated throughout the trajectory of the UAV. More details will be presented in the sequel.

We also observe that (\mathcal{P}_1) is a mixed integer non-linear program (MINLP), which is generally hard to be solved, due to the existence of the binary variable κ_i in constraint (2.10c) and non-convex constraint (2.10b) [10], even if the binary variable κ_i is relaxed to take any value between 0 and 1. The relaxed version of (\mathcal{P}_1) is, nevertheless, non-convex due to the trajectory variables x^n and y^n in constraint (2.10b). To the best of our knowledge, there is no solver for (\mathcal{P}_1) .

2.4 Global Optimization Solution

In this section, we present a solution to optimally solve the problem (\mathcal{P}_1) using a customized branch, reduce and bound (BRB) algorithm [28]. Although the optimization problem is monotonically increasing with respect to κ_i , it is yet hard to be solved by the BRB algorithm due to the non-convex constraint (2.10b), with respect to their variables. In what follows, we transform (\mathcal{P}_1) into another equivalent and monotonically increasing optimization form, based on which a BRB algorithm is customized to solve our optimization problem optimally.

2.4.1 Equivalent Formulation

Consider the following optimization problem:

$$(\mathcal{P}1_O) : \max_{\mathbf{X}, \mathbf{Y}, \mathbf{B}, \mathbf{U}} \sum_{i \in \mathcal{M}} \mathbb{1} \left[\max \left\{ \delta_i \sum_{n=\tau_i}^{\delta_i} b_i^n u_i^n - S_i^{\min}, 0 \right\} \right] \quad (2.11a)$$

$$\text{s.t. } u_i^n \leq \log_2 \left(1 + \Upsilon_{i,n}(x^n, y^n) \right), \forall i \in \mathcal{M}, n = \tau_i, \dots, \delta_i, \quad (2.11b)$$

$$(2.2), (2.6), (2.7), (2.10f), (2.10g), \quad (2.11c)$$

where $\mathbb{1}[x]$ is the indicator function that equals unity if $x > 0$, and zero otherwise. To propose the optimal solution for our optimization, we propose the following lemma. *Lemma 1*: By introducing the slack variable $\mathbf{U} = \{u_i^n \geq 0, \forall n\}$, (\mathcal{P}_1) is equivalent to the monotonic formulation $(\mathcal{P}1_O)$, i.e., (\mathcal{P}_1) and $(\mathcal{P}1_O)$ have the same objective and solution set.

Proof of the Equivalence

To prove that (\mathcal{P}_1) and $(\mathcal{P}1_O)$ are equivalent, we must prove that any feasible solution of (\mathcal{P}_1) is also a feasible solution of $(\mathcal{P}1_O)$. Conversely, from any feasible solution of $(\mathcal{P}1_O)$, we can always find a feasible solution of (\mathcal{P}_1) . Assume that $\check{\mathbf{X}}, \check{\mathbf{Y}}, \check{\mathbf{B}}, \check{\mathbf{U}}$ is a feasible solution set of $(\mathcal{P}1_O)$. It is easy to remark that since $\check{\mathbf{X}}, \check{\mathbf{Y}}, \check{\mathbf{B}}, \check{\mathbf{U}}$ satisfy all constraints (2.10c)-(2.10g) of (\mathcal{P}_1) . Now, we need to prove $\check{\mathbf{X}}, \check{\mathbf{Y}}, \check{\mathbf{B}}, \check{\mathbf{U}}$ also satisfy constraint (2.10b). We can prove it as follows:

$$\begin{aligned} \text{Since: } \quad & \check{u}_i^n \leq \log_2 \left(1 + \Upsilon_{i,n}(\check{x}^n, \check{y}^n) \right), \\ \Rightarrow \quad & \check{b}_i^n \check{u}_i^n \leq \check{b}_i^n \log_2 \left(1 + \Upsilon_{i,n}(\check{x}^n, \check{y}^n) \right), \\ \Rightarrow \quad & \delta_i \sum_{n=\tau_i}^{\delta_i} \check{b}_i^n \check{u}_i^n \leq \delta_i \sum_{n=\tau_i}^{\delta_i} \check{b}_i^n \log_2 \left(1 + \Upsilon_{i,n}(\check{x}^n, \check{y}^n) \right). \end{aligned} \quad (2.12)$$

Let us denote $\delta_i \sum_{n=\tau_i}^{\delta_i} \check{b}_i^n \log_2 \left(1 + \Upsilon_{i,n}(\check{x}^n, \check{y}^n) \right) = S_i(\check{b}_i^n, \check{x}^n, \check{y}^n)$, then we obtain $S_i(\check{b}_i^n, \check{x}^n, \check{y}^n) \geq \delta_i \sum_{n=\tau_i}^{\delta_i} \check{b}_i^n \check{u}_i^n$. At this point $\delta_i \sum_{n=\tau_i}^{\delta_i} \check{b}_i^n \check{u}_i^n$ can take either one of two values, for instance, $\delta_i \sum_{n=\tau_i}^{\delta_i} \check{b}_i^n \check{u}_i^n \geq S_i^{\min}$ or $\delta_i \sum_{n=\tau_i}^{\delta_i} \check{b}_i^n \check{u}_i^n < S_i^{\min}$. Based on this, we can determine $\check{\mathbf{K}}$ as a function of $\check{\mathbf{X}}, \check{\mathbf{Y}}, \check{\mathbf{B}}, \check{\mathbf{U}}$ to satisfy

constraint (2.10b). In particular

$$\begin{aligned}
\text{If: } \delta_t \sum_{n=\tau_i}^{\delta_i} \check{b}_i^n \check{u}_i^n \geq S_i^{\min}, & \quad \Rightarrow \quad \kappa^{\check{}}_i = 1, \\
\text{If: } \delta_t \sum_{n=\tau_i}^{\delta_i} \check{b}_i^n \check{u}_i^n < S_i^{\min}, & \quad \Rightarrow \quad \kappa^{\check{}}_i = 0.
\end{aligned} \tag{2.13}$$

In conclusion, we can determine $\check{X}, \check{Y}, \check{B}, \check{K}$ from $\check{X}, \check{Y}, \check{B}, \check{U}$ to satisfy constraint (2.10b). On the other hand, assume that $\check{X}, \check{Y}, \check{B}, \check{K}$ is a feasible set of solution of (\mathcal{P}_1) . It is easy to remark that since $\check{X}, \check{Y}, \check{B}, \check{K}$ satisfy all constraints (2.2), (2.6), (2.7), (2.10f), (2.10g) of $(\mathcal{P}1_O)$. Now, we need to prove $\check{X}, \check{Y}, \check{B}, \check{K}$ also satisfy constraint (2.11b). Since we got $\delta_t \sum_{n=\tau_i}^{\delta_i} \check{b}_i^n \log_2 \left(1 + \Upsilon_{i,n}(\check{x}^n, \check{y}^n) \right) \geq \kappa^{\check{}}_i S_i^{\min}$. Let us denote $\check{u}_i^n = \log_2 \left(1 + \Upsilon_{i,n}(\check{x}^n, \check{y}^n) \right)$, then this notation makes constraint (2.11b) satisfied. Let us assume $\check{X}, \check{Y}, \check{B}, \check{K}$ is the optimum solution of (\mathcal{P}_1) . We have to prove that $\check{X}, \check{Y}, \check{B}, \check{K}$ is the optimum solution of $(\mathcal{P}1_O)$. This can be shown by contradiction. Assuming $\check{X}, \check{Y}, \check{B}, \check{U}$ is not the optimum solution of $(\mathcal{P}1_O)$, this means that there exists another solution denoted by $\bar{X}, \bar{Y}, \bar{B}, \bar{K}$ which results in a larger objective function of $(\mathcal{P}1_O)$. This means that there exists an index i_1 which makes $\delta_t \sum_{n=\tau_{i_1}}^{\delta_{i_1}} \bar{b}_{i_1}^n \bar{u}_{i_1}^n - S_{i_1}^{\min} \geq 0$, while it is $\delta_t \sum_{n=\tau_{i_1}}^{\delta_{i_1}} \check{b}_{i_1}^n \check{u}_{i_1}^n - S_{i_1}^{\min} < 0$. Now, from i_1 we can determine $\kappa^-_{i_1} = 1$, while $\kappa''_{i_1} = 0$, thus, $\sum_{i \in \mathcal{M}} \kappa^-_i > \sum_{i \in \mathcal{M}} \kappa''_i$. This means that there is at least one more served IoT, which contradicts the assumption of optimality. This completes this part of the proof. Similarly, we can prove any optimum solution of $(\mathcal{P}1_O)$ is also optimum of (\mathcal{P}_1) . This completes the proof.

2.4.2 Proposed BRB Solution

It can be seen that, when the variables b_i^n and $u_i^n, i \in \mathcal{M}, \forall \delta_i$ are fixed, our optimization problem $(\mathcal{P}1_O)$ becomes a feasibility checking for a convex monotonic optimization problem [29]. Consequently, the BRB method can be applied to optimally solve the problem. In the BRB algorithm, a set of \mathcal{N} non-overlapping hyper-rectangles that cover the optimization problem $(\mathcal{P}1_O)$ is maintained, where one of the hyper-rectangles includes the optimal solution.

We define the hyper-rectangle $\mathcal{A} = [\underline{\mathbf{A}}, \bar{\mathbf{A}}]$ that contains all feasible solutions for our optimization problem, where $\underline{\mathbf{A}}$ and $\bar{\mathbf{A}}$ are the lower bound and the upper bound vector that hold the

Algorithm 1: BRB-Optimal: Proposed to Optimally Solve the Data Collection Problem.

- 1 **Inputs:** The hyper-rectangle \mathcal{A} , the error tolerance ε_o , the minimum service amount S_i^{min} and the deadline δ_i .
 - 2 **Initialization:** Apply the reduction procedure to our initial hyper-rectangle \mathcal{A} to obtain the new reduction hyper-rectangle $\text{red}(\mathcal{A})$.
 - 3 Update the hyper-rectangle box $B_1 = \text{red}(\mathcal{A})$ and set iteration number $m = 1$.
 - 4 Branch B_1 into two smaller hyper-rectangle boxes $B_1^{(j)}, \forall j = 1 : 2$.
 - 5 Update the set of hyper-rectangles boxes $D_1 = \{B_1^{(j)}\}$ and the lower bound $\psi_m = LB(B_1)$.
 - 6 **while** $\left(\max_{B_m^{(j)} \in D_m} (UB(B_m^{(j)})) - \psi_m \geq \varepsilon_o \right)$ **do**
 - 7 Select the hyper-rectangle box that has the maximum upper bound
 $B_m = \arg \max \{UB(B_m^{(j)}) | B_m \in D_m\}$ for branching.
 - 8 Branch B_m into two smaller hyper-rectangles $B_m^{(1)}, B_m^{(2)}$ using bisection method along
with the longest edge of B_m . \triangleright //Branching operation//
 - 9 Compute the lower bound for both hyper-rectangles and apply the reduction procedure
to the feasible hyper-rectangles to obtain $\text{red}(B_m^{(1)})$ and $\text{red}(B_m^{(2)})$. \triangleright //Reduction
operation//
 - 10 Compute the maximum lower bound for both reduced hyper-rectangles $\text{red}(B_m^{(1)})$ and
 $\text{red}(B_m^{(2)})$. \triangleright //Bounding operation//
 - 11 Update $\psi_{m+1} = \max(LB(\text{red}(B_m^{(1)})), LB(\text{red}(B_m^{(2)})), \psi_m)$.
 - 12 Remove the hyper-rectangle box that do not contain the optimal solution and update the
set of hyper-rectangles boxes
 $D_{m+1} = D_m \setminus \{B_i | \psi_{m+1} > UB(\text{red}(B_i))\}, \forall i = 1, \dots, \text{cardinal}(D_m)$.
 - 13 $m = m + 1$.
 - 14 **Output** The global optimal solution for maximizing the number of served IoTDS.
-

lowest and the highest values for the variables u_i^n and b_i^n , respectively. In fact u_i^n is bounded $0 \leq u_i^n \leq \log_2(1 + \Upsilon_{i,n}^{max}), i \in \mathcal{M}, \forall \delta_i$, where $\Upsilon_{i,n}^{max}$ is the maximum signal-to-noise ratio when the UAV is hovering right above the IoTDS i at time-slot n and is computed as $\Upsilon_{i,n}^{max} = \frac{P\gamma_0|\hat{h}_i^n|^2}{\sigma^2 H^2}$.

In principle, three operations are conducted for each iteration in the BRB method to improve the lower and upper bounds; namely, Branching, Reduction and Bounding. First, the branching operation is applied to divide the selected hyper-rectangle \mathcal{A} that contains the largest upper bound into two equal smaller hyper-rectangles using one of the partition methods (such as bisection method), and checks the feasibility of each hyper-rectangle through one of the optimization solvers (such as SDPT3) can solve it. Second, the reduction operation is applied to the hyper-rectangles to remove the parts that cannot satisfy the feasible solution to find a smaller hyper-rectangle. Third, a bounding operation is performed to search for the optimal solution by updating the upper and lower bounds.

The algorithm proceeds until the difference between the lower and upper bounds is smaller than a predefined accuracy ε_o . These operations are illustrated in more details in [28]. Algorithm 1, referred to as BRB-optimal, summarizes the BRB solution of (2.11) to determine the optimal UAV trajectory that allows the maximum number of IoTDS to be served.

2.4.3 Convergence Analysis

Algorithm 1 is guaranteed to compute the global optimal solution for maximizing the number of served IoTDS for (\mathcal{P}_1) and its convergence can be proved based on [28], which can be explained as follows. The BRB operations iteratively update and improve the lower and upper bounds of the objective (2.11a). Specifically, in each iteration the lower bound is non-decreasing by updating the step 10, while the upper bound is non-increasing by reduction and bound operations. Due to the monotonic property, after a number of iterations, the gap between the upper and lower bounds of the box that contains an optimal solution is less than or equal to a predefined accuracy level ε_o .

2.5 Low-Complexity Sub-optimal Solution

Since our trajectory optimization is time-sensitive as it depends on the deadlines of the data, the BRB method does not lend itself as an efficient approach and requires a long time to achieve optimal solution especially with a large number of IoTDS. In general, The BRB method is used to generate optimal solutions for relatively small-scale scenarios and also serves as a benchmark for other approaches (as will be shown in our numerical evaluation in the sequel). Motivated by this, we aim to solve the problem for practical network scenarios with a larger number of IoTDS, a low-complexity algorithm is presented to maximize the number of served IoTDS in the next section.

2.5.1 SCA-Algorithm for Maximizing the Number of Served IoTDS

In this section, we attempt to solve (\mathcal{P}_1) based on convex approximation methods and multiple equivalent transformations to generate a more efficient but sub-optimal solution. To solve our optimization, the non-convex constraint (2.10b) is approximated into another equivalent convex equation form and SCA method is applied to solve it iteratively.

As mentioned earlier, (\mathcal{P}_1) is integer non-convex program and difficult to solve. Here, the main difficulty of solving (\mathcal{P}_1) is to deal with the binary variable κ_i which appears in the objective function of (\mathcal{P}_1) . Moreover, even if we relax the binary variable κ_i to make it continuous between '0' and '1', the relaxed version of (\mathcal{P}_1) is still non-convex. The non-convexity of (\mathcal{P}_1) is due to the existence of the non-convex non-concave service amount S_i a function of the UAV's trajectory, which appears in the constraint (2.10b). To tackle the problem, we introduce slack variables $\mathbf{G} = \{g_i^n \geq 0, \forall n, i \in \mathcal{M}\}$ and $\mathbf{C} = \{c_i^n \geq 0, \forall n, i \in \mathcal{M}\}$.

Then, we relax the binary variable κ_i in constraint (2.10c) between 0 and 1. Next, we employ an approximate to function $\log_2(1 + \Upsilon_{i,n})$ by convex approximation with respect to $(x_i^n - x^n)^2 + (y_i^n - y^n)^2$ [30], where at the r^{th} iteration, the following inequality can be obtained:

$$\begin{aligned} \log_2(1 + \Upsilon_{i,n}) &\geq -A_i^{r,n} \left((x_i - x^n)^2 + (y_i - y^n)^2 - (x_i - x^{r,n})^2 - (y_i - y^{r,n})^2 \right) + B_i^{r,n} \\ &\triangleq \zeta_i^{n,r}(x^n, y^n), \end{aligned} \quad (2.14)$$

where

$$\begin{aligned} A_i^{r,n} &= \frac{\alpha(P\gamma_0|\widehat{h}_i^n|^2/\sigma^2)\log_2 e}{2\left((H^2 + (x_i - x^{r,n})^2 + (y_i - y^{r,n})^2)^{\alpha/2} + (P\gamma_0|\widehat{h}_i^n|^2/\sigma^2)\right)} \\ &\quad \cdot \frac{1}{\left(H^2 + (x_i - x^{r,n})^2 + (y_i - y^{r,n})^2\right)}, \forall n, i \in \mathcal{M}, \end{aligned} \quad (2.15)$$

$$B_i^{r,n} = \log_2 \left(1 + \frac{P\gamma_0|\widehat{h}_i^n|^2}{\sigma^2\left(H^2 + (x_i - x^{r,n})^2 + (y_i - y^{r,n})^2\right)^{\alpha/2}} \right), \forall n, i \in \mathcal{M}. \quad (2.16)$$

To this end, we can reformulate (\mathcal{P}_1) as:

$$(\mathcal{P}_{1L}) : \max_{\substack{\mathbf{X}, \mathbf{Y}, \mathbf{B}, \\ \mathbf{K}, \mathbf{G}, \mathbf{C}}} \sum_{i \in \mathcal{M}} \kappa_i \quad (2.17a)$$

$$\text{s.t. } \delta_i \sum_{n=\tau_i}^{\delta_i} c_i^n \geq \kappa_i S_i^{\min}, i \in \mathcal{M}, \quad (2.17b)$$

$$c_i^n \leq b_i^n g_i^n, i \in \mathcal{M}, n = \tau_i, \dots, \delta_i, \quad (2.17c)$$

$$g_i^n \leq \zeta_i^{n,r}(x^n, y^n), i \in \mathcal{M}, n = \tau_i, \dots, \delta_i, \quad (2.17d)$$

$$0 \leq \kappa_i \leq 1, i \in \mathcal{M}, \quad (2.17e)$$

$$0 \leq b_i^n \leq \kappa_i, \forall n, i \in \mathcal{M}, \quad (2.17f)$$

$$(2.2), (2.6), (2.10f), (2.10g). \quad (2.17g)$$

Examining constraint (2.17c), the non-convexity factor $b_i^n g_i^n$ is on the greater side of the inequality. To deal with this constraint, we simply replace the right side of constraint (2.17c) by an equivalent Difference of Convex (DC) function $b_i^n g_i^n = \frac{(b_i^n + g_i^n)^2 - (b_i^n - g_i^n)^2}{4}$, and linearize the concave term $\frac{(b_i^n + g_i^n)^2}{4}$ of the constraint at iteration r . Hence, the constraint (2.17c) is approximated as

$$-\frac{(b_i^{r,n} + g_i^{r,n})^2}{4} - \frac{(b_i^{r,n} - g_i^{r,n})(b_i^n - b_i^{r,n} + g_i^n - g_i^{r,n})}{2} + \frac{(b_i^n - g_i^n)^2}{4} + c_i^n \leq 0. \quad (2.18)$$

Using the above approximation, $(\mathcal{P}1_L)$ transforms into a convex problem, and can be optimally solved by updating parameter $\zeta_i^{n,r}(x^n, y^n)$ iteratively. Algorithm 2, summarizes the SCA-based sub-optimal solution to find the maximum number of served IoTs during a data collection mission. The solution of $(\mathcal{P}1)$ results in a trajectory that maximizes the number of served IoTs during the flight time period. Now, $(\mathcal{P}1)$ assumes a path loss model (no fading) for the channel; to deal with the unknown CSI, we present the following approach. The UAV uses the obtained designed trajectory, and during its deployment it obtains accurate knowledge of the CSI at each time-slot and attempts to serve the devices within its coverage along its trajectory. Namely, using the solution of $(P1_L)$ with path loss model, let \mathcal{M}'_n be the set of IoTs served by the UAV at time-slot n and let $s_{i,pl}^n$ be the service rate each IoT i received at time-slot n using $(P1_L)$. Now, during the operation phase of the UAV, we would ideally like each IoT $i \in \mathcal{M}'_n$ to receive at least a rate equals $s_{i,pl}^n, \forall n$. However, given that the channel has a fading component now, it is likely that some devices may not receive their required rate at some time-slot. Let $\mathcal{M}''_n \subseteq \mathcal{M}'_n$ be the set of devices that, at each slot n , are unable to receive a rate $s_i^n \geq s_{i,pl}^n$. Then, at each slot n , we solve a problem of resource allocation to maximize the number of served IoTs as follows:

$$(\mathcal{P}_E) : \max_{\mathbf{B}, \mathbf{K}} \sum_{i \in \mathcal{M}'_n} \kappa_i \quad (2.19a)$$

Algorithm 2: Sub-optimal: Proposed SCA for Solving ($\mathcal{P}1_L$) and ($\mathcal{P}2_L$).

- 1 **Inputs:** The error tolerance ε , the minimum service amount S_i^{min} , and the deadlines δ_i .
 - 2 **Initialization:** Set the initial trajectory $x^{r,n} y^{r,n}, \forall n$ the resource allocation $b_i^{r,n}, \forall n, \forall i$ and iteration number $r = 1$.
 - 3 **while** ($Obj(r-1) - Obj(r) \geq \varepsilon$) **do**
 - 4 For *SCA-algorithm problem* ($\mathcal{P}1_L$): solve the convex problem (2.17) to obtain the trajectory $x^{r+1,n} y^{r+1,n}, \forall n$ and $b_i^{r+1,n}, \forall n, \forall i \in \mathcal{M}$.
 - 5 For *SCA-distance problem* ($\mathcal{P}2_L$): solve the convex problem (2.22) with the updated subset \mathcal{M}' devices to obtain the trajectory $x^{r+1,n} y^{r+1,n}, \forall n$ and $b_i^{r+1,n}, \forall n, \forall i \in \mathcal{M}'$.
 - 6 Update the UAV's trajectory $x^{r,n} y^{r,n}, \forall n$,
 - 7 Update the resource allocation $b_i^{r,n}, \forall i$,
 - 8 Update $r = r + 1$.
 - 9 **Output:**
 - 10 For *SCA-algorithm problem* ($\mathcal{P}1_L$), the output is the sub-optimal solution for maximizing the number of served IoTDs \mathcal{M}' .
 - 11 For *SCA-distance problem* ($\mathcal{P}2_L$), the output is the sub-optimal solution for minimizing the flight distance.
-

$$\text{s.t. } \delta_i s_i^n(b_i^n) \geq \kappa_i (s_{i,pl}^n + \beta_i \Theta_i^{n-1} - \lambda_i^{n-1}), \forall i \in \mathcal{M}'_n, \quad (2.19b)$$

$$0 \leq \kappa_i \leq 1, i \in \mathcal{M}'_n, \quad (2.19c)$$

$$\sum_{i \in \mathcal{M}'_n} b_i^n \leq 1, \quad (2.19d)$$

$$0 \leq b_i^n \leq 1, \forall n, i \in \mathcal{M}'_n, \quad (2.19e)$$

where β_i is a binary value that takes a value of 1 if data of IoTD i is within its deadline and 0 otherwise. If $\mathcal{M}'' = \emptyset$, then all IoTDs at slot n would receive their minimum service rate. On the one hand, if at least one device i obtains $s_i^n < s_{i,pl}^n$, then we compute the service amount deficit ($\Theta_i^n = s_{i,pl}^n - s_i^n$) for this device and attempt to allocate a surplus service amount in subsequent time-slots. On the other hand, for admitted IoTD i (i.e. $s_i^n > s_{i,pl}^n$), we compute the service rate surplus ($\lambda_i^n = s_i^n - s_{i,pl}^n$) for this device and subtract it from future slots. Indeed, to compensate, the UAV needs in subsequent time-slots to allocate more radio resources for IoTD in deficit to meet their service amount target. It is obvious that \mathcal{P}_E is a convex problem and several optimization solvers can solve it optimally. It is also clear that the UAV will exploit its knowledge of accurate CSI at each slot to resolve \mathcal{P}_E .

2.5.2 Complexity Analysis

In this section, the complexity analysis is discussed. For the optimal algorithm based on the BRB method, which is Algorithm 1, the BRB algorithm requires an extremely long time to achieve the optimal solution especially with a large number of IoTDs. This is due to the fact that BRB is an exhaustive search approach and depends on the problem size, i.e., the number of IoTDs, the service amount and the allocated resources. Furthermore, at each iteration, a feasibility optimization problem is solved. For the SCA-algorithm, the overall complexity of $(\mathcal{P}1_L)$ depends on the solver that is employed to solve $(\mathcal{P}1_L)$. In particular, $(\mathcal{P}1_L)$ is a convex problem and, thus, several interior-point solvers can be employed to solve it. Therefore, we employ the number of Newton steps, denoted by C_s , as a metric to measure its complexity. In fact, the Newton steps depends on the problem size and the number of recursive iterations till convergence from a given initial point. Based on [31] [32], the worst-case C_s to reach a local solution in $(\mathcal{P}1_L)$ can be expressed as:

$$C_s \sim \sqrt{\text{problem size}}, \quad (2.20)$$

where the problem size is the total number of variables of the optimization problem. First we remark that, in the worst case, $(\mathcal{P}1_L)$ must iteratively solve and update the variables. Precisely, there are $3MN + 2N + M$ variables in $(\mathcal{P}1_L)$. Thus, in each iteration, the complexity of solving $(\mathcal{P}1_L)$ is approximately $\sqrt{3MN + 2N + M}$, which induces an overall complexity of $I\sqrt{3MN + 2N + M}$ in the worst-case, where I is a finite number of iterations that depends on the value of error tolerance ϵ .

2.6 Minimizing UAV Flight Distance

In practice, it is essential to minimize the UAV flight distance while satisfying all other problem constraints. Two operation modes are typically considered for the UAV [33]: hovering mode in which the UAV hangs in one spot to collect data from IoTDs and forward flight mode in which the UAV moves from one location to another. Short flight distance infers an efficient trajectory that saves time and forward flight energy (propulsion). As a matter of fact, solving $(\mathcal{P}1)$ allows the UAV to go back and forth to concurrently collect data from distant IoTDs. Doing so incurs

additional propulsion energy consumption that can be saved by minimizing the UAV flight distance. In attempt to conserve propulsion energy, we use the output generated by the solution of (\mathcal{P}_1) that includes the maximum number of IoTDs \mathcal{M}' that may be served by one UAV while meeting their data deadlines, and minimize the distance traveled by the UAV to satisfy the same number of devices with optimized radio resource allocation. Given the initial and the final locations of the UAV trajectory, we formulate the optimization problem with the objective to minimize the flight distance as the below:

$$(\mathcal{P}2) : \min_{\mathbf{X}, \mathbf{Y}, \mathbf{B}} \sum_{n=0}^{N-1} d((x^{n+1}, y^{n+1}), (x^n, y^n)) \quad (2.21a)$$

$$\text{s.t. } S_i(b_i^n, x^n, y^n) \geq S_i^{\min}, i \in \mathcal{M}', n, \quad (2.21b)$$

$$0 \leq b_i^n \leq 1, \forall n, i \in \mathcal{M}', \quad (2.21c)$$

$$(2.2), (2.6), (2.10f), (2.10g), \quad (2.21d)$$

where $d(.,.)$ is the distance between two way-points. The characteristics of $\mathcal{P}2$ deserve more elaboration. In $\mathcal{P}1$, we aim at maximizing the number of served IoTDs. This means that in some scenario, the UAV wastes time and energy while serving the maximum number of served IoTDs. On the contrary, $\mathcal{P}2$ guarantees that the UAV must minimize the traveling distance when the maximum number of IoTDs is achieved.

This problem is essentially equivalent to a well-known problem called Traveling Salesman Problem (TSP), which is known to be NP-hard. One straightforward approach for solving $\mathcal{P}2$ is to find the nearest device (known as the greedy or nearest neighbor algorithm) under deadline constraint. However, serving devices once at a time is an inefficient approach. Therefore, we propose an efficient sub-optimal solution to $\mathcal{P}2$ based on SCA algorithm. Similar to $\mathcal{P}1$, $\mathcal{P}2$ is non-convex problem because the non-concave non-convex function S_i in constraint (2.21b). By introducing convex approximation in constraint (2.17d) and the slack variables $\mathbf{W} = w_i^n \geq 0, \forall n, i \in \mathcal{M}'$ and

$\mathbf{Z} = \mathbf{Z}_i^n \geq 0, \forall n, i \in \mathcal{M}'$, $\mathcal{P}2$ can be solved by iteratively solving the following approximated convex problem formulated at the $r + 1$ iteration index as

$$\mathcal{P}2_L : \min_{\substack{\mathbf{X}, \mathbf{Y}, \mathbf{B}, \\ \mathbf{W}, \mathbf{Z}}} \sum_{n=0}^{N-1} d\left((x^{n+1}, y^{n+1}), (x^n, y^n)\right), \quad (2.22a)$$

$$\text{s.t. } \delta_i \sum_{n=\tau_i}^{\delta_i} z_i^n \geq S_i^{\min}, i \in \mathcal{M}', \quad (2.22b)$$

$$z_i^n \leq b_i^n w_i^n, i \in \mathcal{M}', n = \tau_i, \dots, \delta_i, \quad (2.22c)$$

$$w_i^n \leq \zeta_i^{n,r}(x^n, y^n), i \in \mathcal{M}', n = \tau_i, \dots, \delta_i, \quad (2.22d)$$

$$0 \leq b_i^n \leq 1, \forall n, i \in \mathcal{M}', \quad (2.22e)$$

$$(2.2), (2.6), (2.10f), (2.10g). \quad (2.22f)$$

Algorithm 2, presents SCA-Distance to sub-optimally minimize flight distance and determine resource allocation among served IoTDS.

2.7 Greedy Location/Deadline-based Algorithms

In this section, we summarize two greedy approaches as benchmarks to solve our trajectory optimization problem. Two approaches have been devised to find the trajectory of the UAV. The first approach is based on the minimum distance, where the UAV flies and hovers above the closest IoTDS and allocate all resources to the IoTDS if and only if the UAV speed, flying time, and minimum service constraints are satisfied. It is worth mentioning that by knowing the locations of the IoTDSs the above constraints could be checked without applying an optimization checking at the intermediate steps. The UAV keeps repeating the process either until no more IoTDSs can be served or mission time is over. The second approach decides the trajectory of the UAV by allocating all the resources to serve the IoTDS with the shortest deadline if and only if the above constraints are satisfied. The two approaches are described in Algorithm 3.

Algorithm 3: Greedy Approaches for Data Collection.

- 1 **Inputs:** The UAV initial location x^0 and y^0 , the minimum service amount S_i^{min} , the deadline δ_i , the maximum speed of the UAV v_{max} , and the location of all IoTDs x_i, y_i ;
 - 2 **Initialization:**
 - 3 For *Distance-based approach*: Sort all IoTDs based on the distance to the current location of the UAV, $d_{i,U}$, where the closest IoTD is at the top of the list, and set the updated time $N' = N$.
 - 4 For *Deadline-based approach*: Sort all IoTDs based on their deadline, where the most urgent device (minimum deadline) is at the top of the list, and set the updated time $N' = N$.
 - 5 **for** $i \in \mathcal{M}$ **do**
 - 6 For *Distance-based approach*: Select the closest unmarked IoTD to the current location of the UAV.
 - 7 For *Deadline-based approach*: Select the most urgent unmarked IoTD.
 - 8 **while** $(S_i(b_i^n, x^n, y^n) \geq S_i^{min} \text{ and } N' > 0 \text{ and } \frac{d_{i,U}}{N'} \leq v_{max})$ **do**
 - 9 Find the minimum time to serve the IoTD and update the flying time N' .
 - 10 Update the UAV location with the current location of the IoTD.
 - 11 Mark the IoTD, and updated time N' .
 - 12 **Output:**
 - 13 For *Distance-based approach*: The sub-optimal solution for maximizing the number of served IoTDs based on distance.
 - 14 For *Deadline-based approach*: The sub-optimal solution for maximizing the number of served IoTDs based on deadline.
-

2.8 Simulation Results and Discussion

In this section, we evaluate the performance of the proposed algorithms numerically. The main input parameters that are used in this simulation are listed in Table 2.1. We assume a geographical area of size $0.8 \times 0.8\text{km}^2$ in which 1 UAV is dispatched to collect data from IoTDs. We assume that the required minimum service amount for all IoTDs is identical, and all IoTDs can communicate with the UAV within the given area. The data generation, deadlines and locations of the IoTDs are generated based on a normal distribution, these deadlines and locations' samples are then used to identify the UAV trajectory and maximize the number of served IoTDs. For sake of illustration, we assume the flight duration in the simulations is sampled every 1secs, unless mentioned otherwise. We also compare with two greedy ones.

Table 2.1: Simulation Parameters in UAV-assisted IoT Wireless Networks

Parameter	Value
IoT transmission power, P	0.1mW
UAV altitude, H	100m
Channel power gain, γ_0	-50dB
Noise power, σ^2	-110dBm
UAV max speed, v_{max}	50m/s
Path-loss exponent, α	2.7
The error tolerance ϵ	10^{-3}

Optimal Solution:

We start by first studying the performance of our designed BRB. To show its convergence towards the optimal solution, we consider a small scenario with a 3 IoTs and a short flying duration ($N = 15$ time-slots sampled every 5secs). As shown in Fig. 2.2, BRB requires a high number of iterations to converge, and the optimal solution falls between the upper and lower bounds. BRB spends a large amount of time to close the gap between the upper and lower bounds. This clearly demonstrates that when considering a larger number of IoTs, the BRB method will take much more time to find the optimal amount of service for all devices, which may not be an effective and practical solution to meet the IoT deadlines. Here, the maximum number of served IoTs is computed by

$$\left\lfloor \frac{\sum_{\forall i \in \mathcal{M}} S_i}{S_i^{min}} \right\rfloor, \text{ where } \lfloor \cdot \rfloor \text{ denote the floor function.}$$

Sub-Optimal Proposed Solution:

The CVX toolbox and numerical convex optimization solver SDPT3 are used to solve our optimization sub-optimally. We set the UAV's initial and final locations at $[0 \ 400]$ and $[800 \ 400]$, respectively.

We start by solving $(\mathcal{P}1_L)$ where we assume a path loss model for the air to ground (A2G) channel. Fig. 2.3(a), depicts the UAV trajectory for collecting data from IoTs (for a network of 15 devices) over a period $N = 90$ time-slots and a minimum requirement $S_i^{min} = 25\text{bits/Hz}$. The values of the deadlines (in time-slots) are depicted next to each IoT in the Fig. 2.3; the maximum deadline is equal to 90 time-slots. We observe that the total number of served IoTs, through this trajectory, within this time period is equal to "12", i.e., 80% of the total number of devices. To see

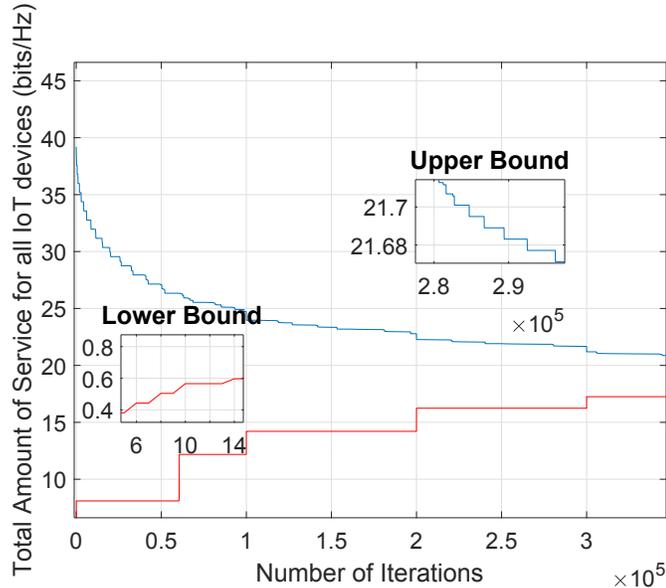


Fig. 2.2: BRB-optimal algorithm to maximize the number of served IoTDs.

the impact of the A2G channel on the performance, we next assume that at the time of trajectory design, the operator gained accurate knowledge of the CSI for all subsequent time-slots (somehow unrealistic, but serve the purpose of the study). We assume a realistic Rician channel model with $K = 3$; Fig. 2.3(b) shows the trajectory the UAV will take to serve the maximum number of IoTDs. Clearly, the trajectory is different from that of Fig. 2.3(a); here, the UAV flies closer to each device in order to compensate for the fading on the channel and serve the device with the required service amount. This explains the difference in the obtained trajectory, however, we notice as well that the UAV serves exactly the same IoTDs as in the simple A2G channel. This shows minor effect on the number of served devices.

To better understand the impact of the channel, we next take the trajectory obtained with the path loss model, and fly the UAV on that trajectory, but this time at each slot, use a Rician channel and vary the value of K . The reason for doing this is to assess the impact of the channel as we operate the UAV. We look at each of the 12 IoTDs and measure their achieved service amount. Here the service amount is computed by replacing the channel in Equation 2.8, using the values for b , x , and y after solving (\mathcal{P}_{1L}) . The results are depicted in Fig. 2.4(a). We observe that for smaller values of K , not all IoTDs (of the 12) would receive the minimum required service amount (e.g., devices 4, 7, and 10 received slightly below the minimum), however as the value of K increases, the impact

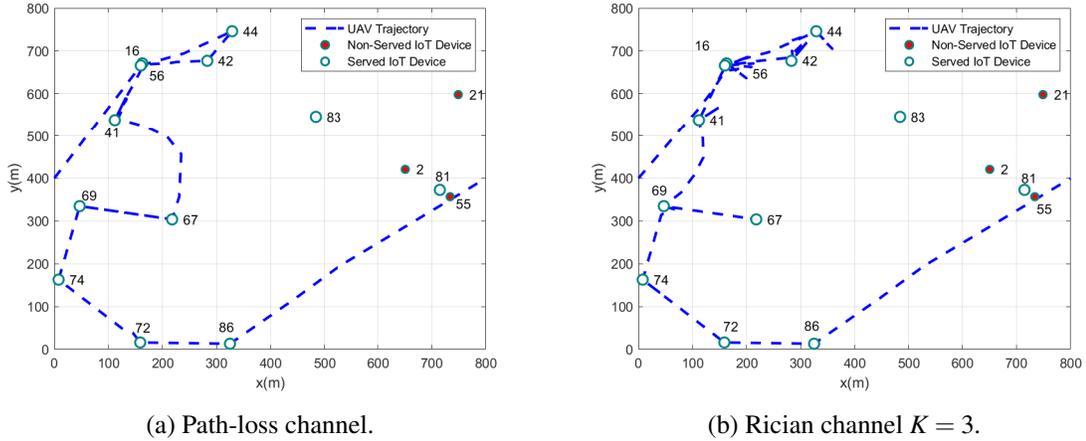
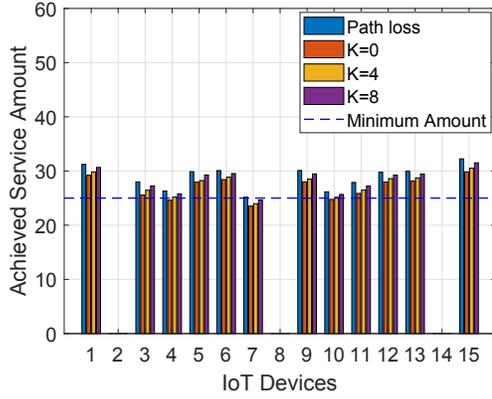


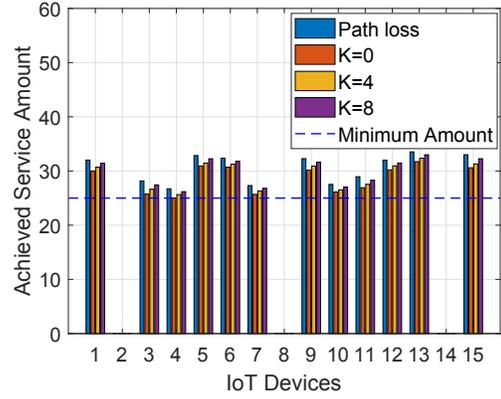
Fig. 2.3: UAV trajectory to maximize the number of served IoT devices.

of the fading diminishes, the LoS becomes the dominant and all devices get served. To overcome this issue, we turn our attention to evaluate Enhanced-algorithm in problem \mathcal{P}_E . Here, again the UAV will fly using the trajectory of (\mathcal{P}_{1L}) (i.e., using the simple A2G channel), but as the UAV operates, at each time-slot, it collects accurate CSI from the location, and allocates radio resources to serve the IoTDs served by the path loss trajectory with their service amounts. The results are depicted in Fig. 2.4(b), where we show the service amount attained by each IoTD, for the path loss channel, the Rician ($K = 3$) and Rayleigh ($K = 0$) channel with knowledge of CSI at each slot. Surprisingly, \mathcal{P}_E is always able to allocate resources so that devices attain their minimum amount; this is possible through keeping track of the surplus and deficit in service amount for each device along the trajectory, so any IoTd in deficit will be compensated in subsequent slots if possible.

We present in Fig. 2.5(a) the per IoTD allocation of resources. It is observed that at each slot, the UAV allocates radio resources unequally among the devices, depending on the deadlines and their locations. We also observe that while serving the devices, the UAV may allocate the resources in multiple, not contiguous, time-slots (such as the first IoTD) or in a one time-slot (such as fifth IoTD). Further, the UAV may not be able to meet the deadlines of all devices. Although the UAV was able to meet the deadline of the farthest ones, our proposed solution puts more effort to fulfill the requirements of the nearest devices within the deadlines instead of wasting time to fulfill the requirements of farther ones. To better understand our results, as shown in Fig. 2.5(b), in the first time-slot the UAV increased its speed to reach the first subset of devices then decelerates to allow

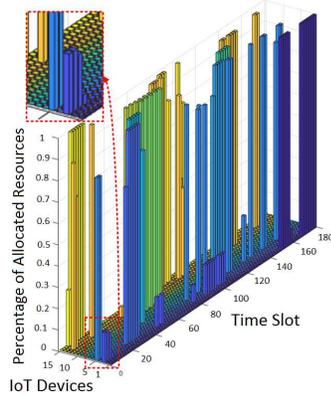


(a) SCA-algorithm over Rician channel.

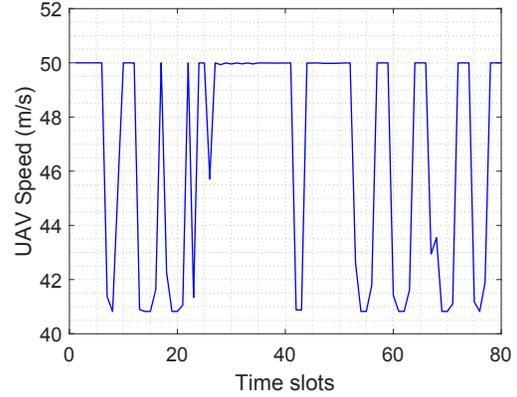


(b) Enhanced-algorithm with CSI knowledge.

Fig. 2.4: Achieved service amount per IoTD.



(a) Resource allocation for each IoTD.



(b) UAV speed versus time-slots.

Fig. 2.5: Performance of proposed SCA-algorithm.

enough time for their data upload before their deadlines expire. It then accelerates again to reach another subset of devices to collect their data.

The percentage of served IoTDs is another performance metric we study. Fig. 2.6(a) depicts this metric versus the service amount, and for different deadlines (in time-slots and for a network of 20 devices). Clearly, as we increase the minimum service amount S_i^{min} per IoTD, the UAV will spend more time and radio resources for collecting the data from one device before flying to another device to collect its data. Furthermore, with less strict deadlines, the UAV will have extra time and enough resources to serve more devices compared to tighter deadlines. Fig. 2.6(b) depicts the percentage of served IoTDs versus the network size (maximum number of IoTDs located in the same area);

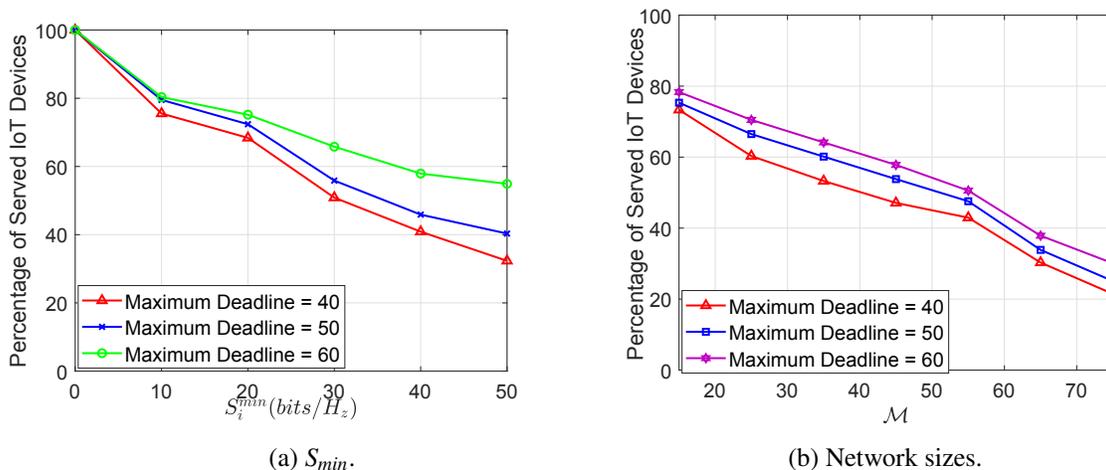


Fig. 2.6: Percentage of served IoTs for SCA-algorithm.

when the required minimum service amount $S_i^{min} = 20\text{bits/Hz}$, as can be seen, the percentage of served IoTs decreases by increasing the number of IoTs within the same area. Since the radio resources and flying time are limited, whenever more devices are considered within the same area, then less radio resources are being allocated for each IoT. In turn, by increasing the deadlines, the percentage of served IoTs will increase as expected since the UAV will have extra time to allocate more resources.

Next, we study the performance of $(\mathcal{P}2_L)$, whose objective is to find an efficient trajectory for serving the IoTs. By inspecting Figs. 2.3(a) and 2.3(b), we observe a long trajectory with many detours taken by the UAV to be able to collect information from the largest number of devices. $(\mathcal{P}2_L)$ attempts to find a more efficient trajectory and Fig. 2.7(a) depicts the obtained trajectory (assuming a path loss channel) for serving the same devices. This trajectory is indeed more efficient since the UAV avoids flying back and forth to serve the same device at different time-slots.

It should be noted that owing to the flexibility of the UAV (rotary-wing UAV), the UAV is able to hover in one place while collecting data from multiple devices, achieving the same number of served IoTs with a minimum movement. As shown in Fig. 2.7(b), the UAV increases its speed to serve a subset of devices before decreases its speed for a certain time for collecting data, then it increases its speed again to serve another subset of devices.

Now, when the trajectory is optimized, we fly the UAV and study whether the UAV is able

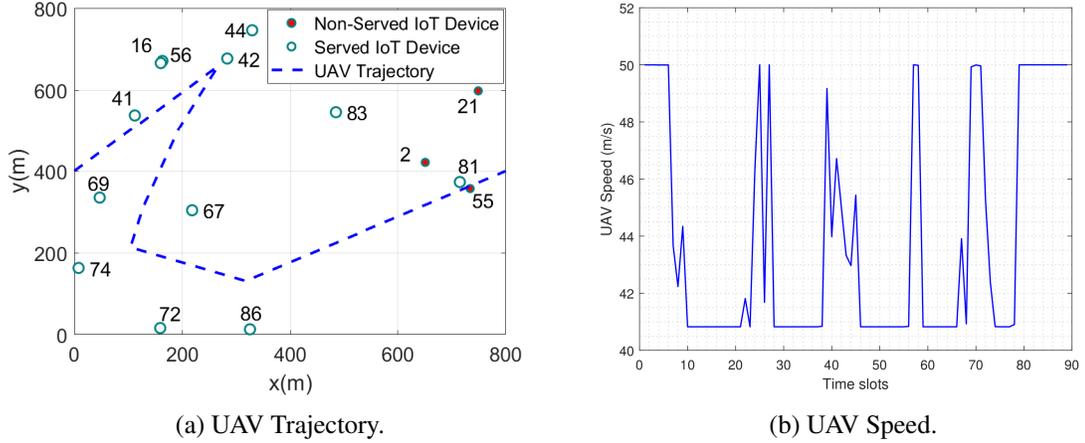


Fig. 2.7: Performance of proposed SCA-distance.

to allocate resources at each slot for the corresponding devices, taking into account accurate CSI knowledge from that location. Here, during the flight of the UAV, the UAV collects information about the channel status, and re-optimizes the allocation of resources, keeping track of the deficit and surplus for each IoTD along the path. We show in Fig. 2.8 the results for different channel realizations. When the fading component is high ($K = 1$), we observe, in some of our simulations, the UAV fails to serve all IoTDs, however when $K = 6$, and the path loss becomes dominant, all devices are served. Further inspecting Fig. 2.7(a), we observe that the UAV may hover far away from some devices and when the condition of the channel is degraded, the UAV does not have enough spectral resources to meet the requirement of the devices. Hence, we conclude a trade-off between the enhanced trajectory and the achieved performance in terms of number of served devices.

One can observe that, although we serve the maximum number of IoTDs over the period N , it is obvious that the UAV trajectory is not an efficient trajectory. In order to enhance the trajectory, we follow SCA-distance algorithm proposed in Algorithm 2 to minimize the flight distance. This achieves the same number of served IoTDs with much better trajectory. It can be seen in Fig. 2.7(a), that the same number of IoTDs with the same deadlines can be served with an enhanced trajectory without having to fly back and forth.

Next, we compare the energy of both trajectories for SCA-distance and SCA-algorithm for the

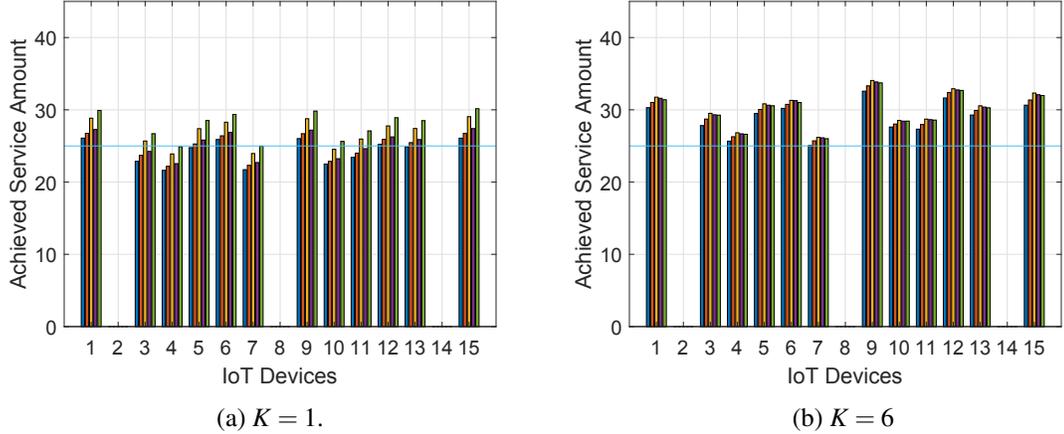


Fig. 2.8: Achieved service amount over different channel realizations for the Enhanced-algorithm.

same configuration. We use the same energy model and the corresponding typical parameters mentioned in [33]. As shown in Fig. 2.9, it can be observed that the proposed SCA-distance shows a lower energy consumption compared to SCA-algorithm, because the former allows the UAV to go back and forth to concurrently collect data from distant IoTDs. Doing so incurs additional propulsion energy consumption that can be saved by minimizing the UAV flight distance as shown for SCA-distance.

Comparison with Greedy Solutions:

Here, we evaluate the performance of two greedy methods for determining the UAV trajectory, and compare them to our proposed solution. The two greedy methods work as follows. A trajectory is computed either based on the shortest distance, i.e., the IoTD closest to the current location is selected to be served, or based on the data deadline, i.e., the UAV flies from its location to serve the IoTD with the strictest latency. Once a trajectory is decided, the UAV follows the designed path and at each time-slot, it allocates the radio resource to maximize the service rate of each device. We compare the performance of these methods with our SCA-algorithm and SCA-distance methods. The results are presented in Fig. 2.10 .

In this evaluation, we fix the flying time ($N = 90$ time-slots), the locations of IoTDs and the deadlines; we consider a network of $\mathcal{M} = 20$ devices, a minimum service amount $S_i^{min} = 60$ bits/Hz, and a maximum deadline of 90 time-slots. Our proposed algorithms are validated in free trajectory

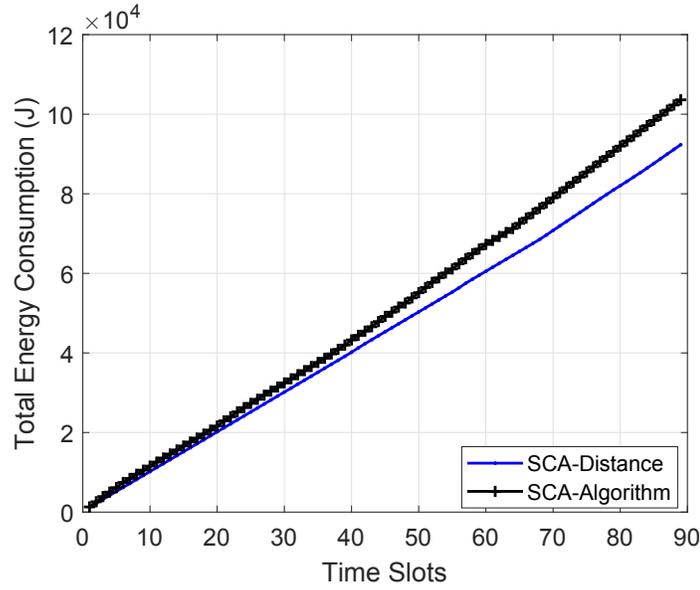


Fig. 2.9: Total energy consumption of the UAV.

(i.e. the initial location is determined at $[0 \ 400]$ while the final location is not set) for a fair comparison. The trajectory of sub-optimal solution is shown in Fig. 2.10(a), where the UAV exploits its mobility as well as the efficient allocation of radio resources to adapt its trajectory to fly closer to a subset of devices to meet their deadlines. It can be seen while considering these parameters, the UAV is able to serve "15" devices out of "20" (%75). The same percentage can be achieved with the enhanced proposed trajectory, SCA-distance, as shown in Fig. 2.10(b), where the trajectory is further optimized to serve the same number of IoTDs.

In contrast, both greedy approaches optimize the trajectory of the UAV differently while allocating the whole resources to one IoTD at a time, as explained in Algorithm 3 in Appendix 2.7. It can be observed from Fig. 2.10(c), while considering the maximum speed of the UAV, devices locations and the expected traveling time, the UAV adapts its trajectory to serve the closest IoTD regardless of its deadline. It can also be observed that the UAV misses the most urgent device while maximizing the number of served ones. The UAV is able to serve only 11 IoTDs (55%), and this is due to the fact that along its trajectory, the UAV allocates its radio resources to serve only one device at a time. In Fig. 2.10(d), while considering the most urgent deadline, the maximum speed and the expected traveling time, the UAV adjusts its trajectory to serve the most urgent IoTD regardless of its location. Following this greedy method, the UAV was only able to collect data from 45% of the

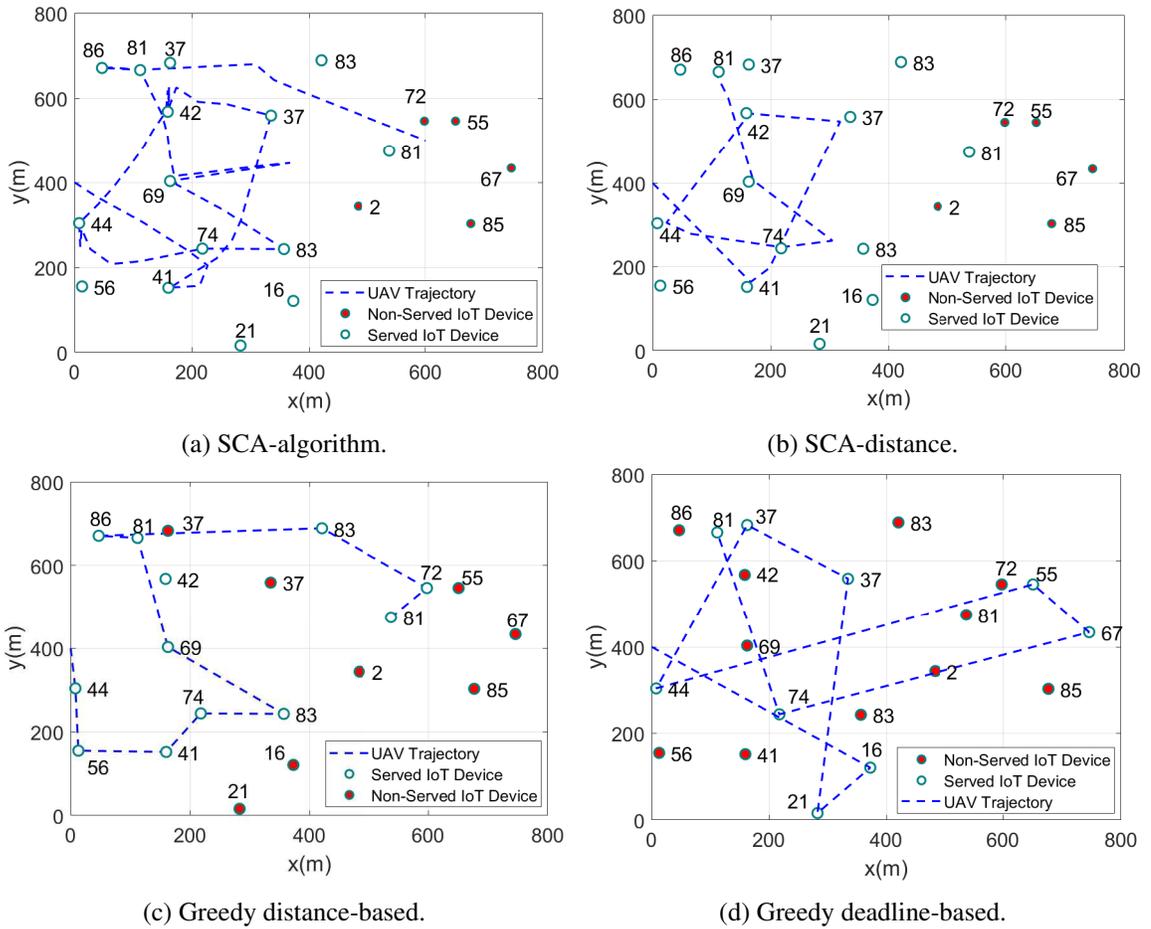


Fig. 2.10: Optimizing the UAV trajectory to maximize the number of served IoTs for alternative solutions.

IoTds, and this is due to the fact that the UAV wastes more time in flying to reach the device with the strictest deadline and hence ends up with little time to collect data.

In Fig. 2.11(a), we compare the effect of deadlines on the performance of the proposed solutions with the greedy as well as a static benchmark scheme over a period of $N = 90$ time-slots and minimum service amount $S_i^{min} = 20$ bits/Hz. For the static UAV scheme, the UAV is located in the middle of the given area (i.e. located at $[400 \ 400]$). The comparison is investigated in different environments, the dense environment (i.e. $\mathcal{M} = 40$ devices) and sparse environment (i.e. $\mathcal{M} = 10$ devices). It can be observed that while increasing the maximum deadline the SCA-algorithm achieves higher performance compared to the other approaches. The UAV then is able to optimize both its trajectory and radio resources to fly closer to multiple IoTds to serve them simultaneously to maximize

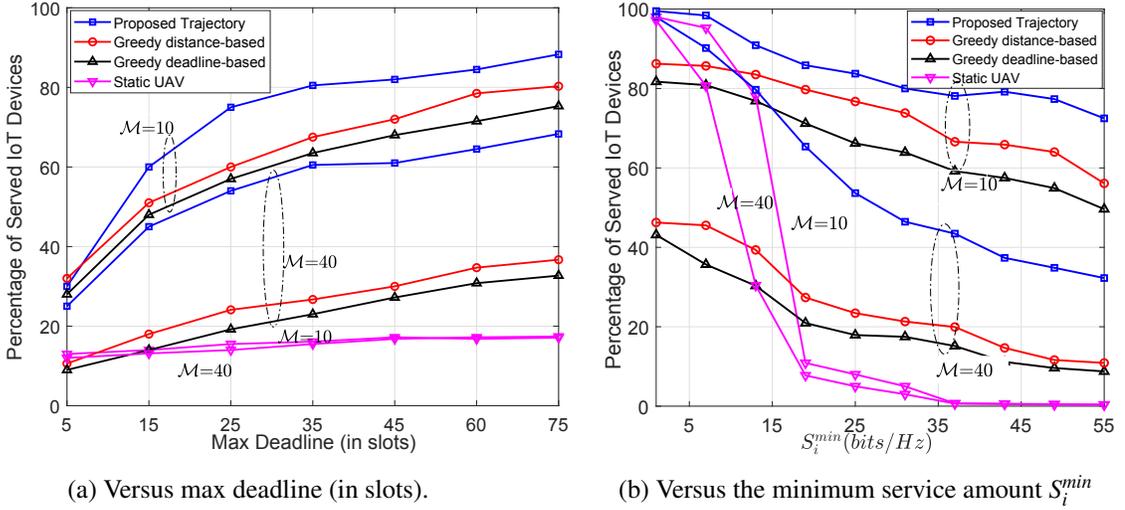


Fig. 2.11: Percentage of served IoTDs for SCA-distance compared to alternative solutions.

the served IoTDs. We also observe that when the deadline is very tight, the static achieves best performance since the UAV does not waste any time flying, but rather spends all its time serving the IoTDs. However, as the deadline starts increasing, the percentage of served devices starts to increase, especially in the proposed method as well as the greedy methods, both for sparse and dense networks. Our proposed method indeed achieves superior performance against the other methods because concurrently the trajectory as well as the allocation of radio resources are optimized to serve the maximum number of devices. In contrast, the greedy methods follow each a trajectory that is somehow oblivious to the objective of serving the largest number of devices. In both greedy methods, as explained above, the UAV flies either to the closest device or to the device with the strictest deadline, and allocates all resources to serve that device. Along the process, some time gets lost due to the abundance of radio resources which could be used to serve more devices. It is also shown in Fig. 2.11(a) that the distance-based greedy approach achieves better performance than the deadline-based since the latter makes the UAV waste more time to fly closer to the IoTD with strictest deadline.

Next, we study the impact of the minimum service amount S_i^{\min} on the performance of the different possible solutions with maximum deadline 90 time-slots and over a period of $N = 90$ time-slots. As shown in Fig. 2.11(b), with the lower service amount, optimizing the radio resources is sufficient to maximize the number of served IoTDs as illustrated for the static and SCA-algorithm.

We also observe that, with increasing the minimum service amount, it is obvious that the static UAV will not be able to collect the required data by only optimizing the radio resources. Optimizing the UAV trajectory becomes more crucial for achieving better communication channels to increase the transmission rate with larger minimum service amount. For example, in the sparse environment (i.e. $\mathcal{M} = 10$ devices) to achieve the minimum service amount of 20bits/Hz, the percentage of served devices for our proposed solution is almost 88% of the total number of devices with optimizing the resources and the UAV trajectory compared to. On the other hand, by increasing the number of devices (i.e. increase the density), optimizing the resources becomes significant to serve more devices. By comparing the greedy approaches with the proposed algorithm with increasing the minimum service amount, our proposed solution achieves higher performance since the trajectory and radio resources are jointly optimized.

2.9 Summary

This chapter studied the time constrained data collection from IoTDS. Since IoTDS have different QoS requirements within certain deadlines, the UAV trajectory and radio resource allocation are optimized to collect a differentiated amount of data from IoTDS. We formulated our optimization problem to maximize the number of served IoTDS while guaranteeing the minimal amount of data uploaded from each served device within the given deadline. Although our problem is non-convex, we solved it optimally by BRB algorithm. By convexifying our problem we provided a low complexity solution to solve our problem efficiently, then we extended the solution to generate an enhanced trajectory in order to minimize the distance traveled by the UAV while serving the IoTDS. Under variable deadlines and minimum service amounts, our proposed solution outperformed alternative solutions including distance- and deadline-based greedy approaches, and static UAV placement in terms of the percentage of served IoTDS (average improvement of 10% - 50%).

Chapter 3

UAVs as Active Relays in IoT Networks for Fresh Information

3.1 Background, Related Works, and Contributions

Emerging applications of smart cities require efficient, timely and reliable real-time updates to enable remote monitoring and control of physical processes and networked control systems. Taking wildfires as an example, it is extremely important to detect the fire and its real-time status update in a timely way so as to notify residents and authorities about its location. The wildfire crisis in California in 2018 killed thousands of animals, destroyed thousands of homes, forced hundreds of thousands of residents to evacuate, and killed twenty-five people [35]. Adequate timely response could have been possible if fresh real-time monitoring had been available. Compared to traditional data networks, fresh real-time monitoring has unique features. The first is the *Markovian feature* at which the existing status-update can be completely replaced by the newly arrived status-update information. The second is that the real-time status updates require more frequent updates with minimal timeliness. Timeliness is different from the conventional performance metrics as studied in Chapter 2, where timeliness is counted from the time the information is generated/sampled at the

The work done in this chapter leads to an IEEE published journal [34]

sensor until its reception for processing at the destination. Timeliness of fresh information therefore consists of three delays: the delay until data is being sampled/generated, the delay until the transmission of sampled data is scheduled, and their communication delays through the network. The freshness of status update information is quantified by a new performance metric, the *Age-of-Information* (AoI), in which a lower AoI implies fresher collected information. The collected information with high AoI may be inconsistent with the present status, which may lead to losing its meaning. The AoI is defined as the time elapsed since the most recent successful transmission of the valid status update data [36]. AoI is introduced to evaluate the freshness of information from the destination's perspective, where it characterizes latency and inter-delivery time intervals. Conventional performance metrics lack the ability to capture the freshness of the collected information since they (such as the latency) do not account for the time elapsed since the information was first generated at the Internet of Things Devices (IoTDs). As a result, conventional performance metrics may not deem suitable for real-time status-update applications. For more details on AoI and its applicability, the reader is referred to [36].

UAVs can be deployed to designated areas in order to provide affordable network connectivity to low-resource Internet of Things (IoT) devices by relaying data to the nearest Base-Station (BS). UAVs can also dynamically adjust their altitude to establish better communication links to IoTDs and improve network performance. In fact, UAVs as mobile relays introduce a new challenging task that should be carefully addressed. In particular, both performance metrics, that is, latency and inter-delivery time, should be optimized in the communication from IoTDs to UAVs and then from UAVs to the BS. To the best of our knowledge, the impact of UAVs as mobile relays with unreliable transmission condition on the AoI in a stochastic environment has not been explored.

In this chapter, we consider a UAV-assisted wireless IoT network, where UAVs act as mobile relays to the BS (or remote server) for a number of IoTDs with limited transmission capabilities¹. IoTDs sample a stochastic process and their sampled data need to be uploaded to the UAVs over unreliable channels, which in turn relay sampled data to the BS that processes these packets. UAVs are assumed to be equipped with virtual queues to re-transmit undelivered sampled data, thus improving

¹Due to IoTDs' energy constraints and environmental obstacles (i.e., blockages), IoTDs are commonly unable to communicate over a long distance. In other words, having strong direct communication links is difficult to achieve. Therefore, UAVs with virtual queues act as mobile active relays between IoTDs and the BS is proposed.

the transmission efficiency. Intuitively, the altitude of a UAV affects the propagation characteristics of the channel between IoTDs and the UAV and between the UAV and the BS; thus, the altitude of a UAV affects the AoI. For example, when a UAV flies at a higher altitude, the probability to establish a Line-of-Sight (LoS) link is higher with the IoTDs as well as with the BS. At higher altitudes, the long distance path loss is higher and thus, the received signal power is relatively small. The converse is true, that is, when a UAV flies at lower altitudes.

The wireless channel quality depends, to a large extent, on the position of the UAV since the surrounding environment at different positions varies (height or density of buildings). Therefore, to ensure the freshness of the sampled data, we jointly study dynamic UAV altitude control and scheduling policy from IoTDs to the UAV and from the UAV to the BS. The main objective of the stochastic scheduling and altitude control problem is to minimize the Expected Weighted Sum AoI (EWSA) of sampled data, which is dependent on the wireless channel conditions and coupled with the altitude of the UAV, to ensure effective communication. Thus, the deployed UAV must decide on the best streams to be relayed. To the best of our knowledge, our work is the first to study the Age-of-Information in relay networks under unreliable transmission conditions.

Recently, several works have been proposed to address the deployment of one or more UAVs for maintaining the freshness of the collected information (captured by AoI). Specifically, authors in [37–43] proposed machine learning (ML) approaches to design the UAV’s trajectory while considering the freshness of the collected information. In [37,38], the authors proposed DRL based on a compound-action actor-critic algorithm to design the trajectories of a swarm of UAVs that minimize the AoI while considering the cooperative sensing and transmission among the UAVs. In [39], the authors leveraged DRL based on deep Q-network (DQN) algorithm to optimize the UAV’s trajectory and transmission scheduling that minimizes the Weighted Sum-AoI. In [40], the authors exploited RL based on a Q-learning algorithm to optimize a UAV trajectory for data collection mission to minimize the expired data packets. In [41], a DRL based on deep Q-network (DQN) algorithm is adopted to design the trajectory of a single UAV to minimize the long-term AoI of multiple ground nodes. In [42], the authors optimized the UAV’s trajectory using deep Q-network algorithm to minimize the average AoI while preserving the packet loss ratio as low as possible. In [43], a deep Q-network algorithm is used to find the trajectory of a UAV that minimizes the weighted sum AoI

of the ground nodes while considering the energy consumption of the UAV. Another direction of research has focused on various tools such as dynamic programming and iterative optimization algorithm to design the flight trajectory of the UAV along with other communication parameters (e.g., energy, scheduling, collection time, etc.) [22, 23, 44–49].

In this work, different from the aforementioned works, we study the Age-of-Information in relay networks under unreliable transmission conditions in the absence of the knowledge of channel state information. In this chapter, we propose a novel model in UAV relay-assisted IoT networks which takes into account the channel reliability between IoTDs and the UAVs and that between UAVs and BS to improve the freshness of information. In addition, a concrete analytical characterization of AoI for UAV-assisted IoT networks under unreliable transmission conditions is derived when UAVs with virtual queues act as mobile active relays between IoTDs and the BS. In addition, an optimization problem is formulated to find the optimal altitude and scheduling policy that minimizes the Expected Weighted Sum AoI, and then the optimization problem is shown to be difficult to solve. Furthermore, we formulate the IoT-UAV-BS status update problem as a Markov Decision Process (MDP) and develop Deep Reinforcement Learning (DRL) to learn environment dynamics in order to handle the altitude and scheduling policy of UAVs. In particular, we leverage the Proximal Policy Optimization (PPO) algorithm, which is a highly stable state-of-the-art model-free DRL, to find the best policy that efficiently minimizes EWSA. Then, the performance of the proposed PPO algorithm is compared with different baseline policies and the impact of different design parameters is analyzed. Besides, the proposed algorithm is evaluated through extensive simulations.

3.2 System Model

Consider a geographical area, where a number of IoTDs with limited capabilities is distributed over a given area and continuously sample time-sensitive information (that is, time-stamped, status-update packets). One-hop transmission is assumed not effective because transmission capabilities of IoTDs are limited, hence, multiple UAVs are deployed for relaying transmissions to the BS.

Given the distribution of IoTDs, multiple UAVs are deployed, each to cover one cluster of IoTDs. The horizontal coordinates of each UAV (x_U, y_U) are assumed to be placed at the center of

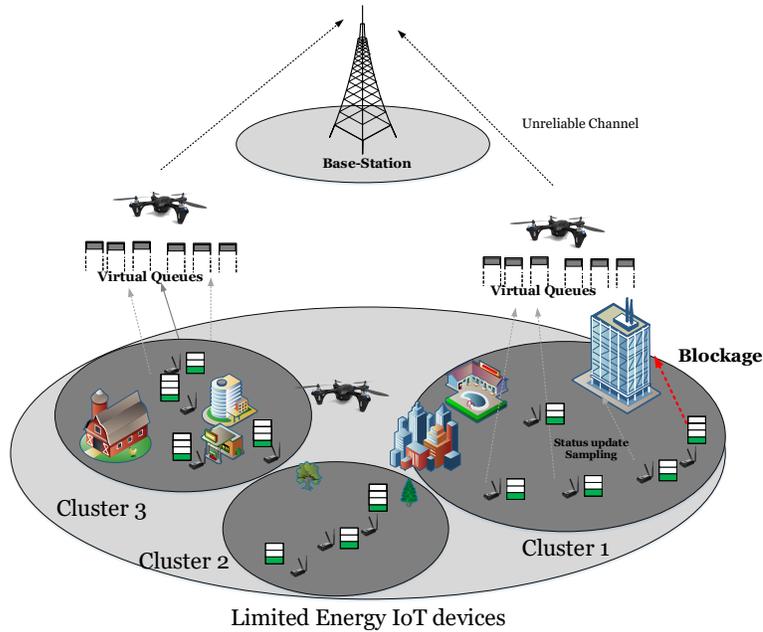


Fig. 3.1: An illustration of our system model.

the area. For simplicity, we assume each cluster consists of a set \mathcal{M} of M IoTDs². Let the locations of IoTDs be $(x_i, y_i, 0), \forall i \in \mathcal{M}$ at ground level. Each IoTD is relayed by the closest UAV to the BS that is located at (x_s, y_s, H_S) , where H_S denotes the height of the BS.

Each UAV is assumed to be equipped with M' virtual queues with $M' > M$, where the UAV only stores the latest received packet for each IoT. The UAV then schedules or retransmits to the BS if the transmission fails due to the unreliability of the channel. We consider the system over multiple time frames. Each of these frames is further divided into equal segments, that is, N time-slots of length δ_t , which is normalized to unity. At the beginning of every time-slot n , the deployed UAV either remains idle or schedules an IoTD $i \in 1, 2, \dots, M$ to transmit its status-update packet over an unreliable wireless communication channel. The deployed UAV then relays the status-update packets over another unreliable wireless communication channel to the BS. The deployed UAV is assumed to operate in a half-duplex mode. Thus, the UAV can either transmit to the BS or receive status-update packets from IoTDs at a time. To achieve a reliable communication, dominant interference should be avoided. Thus, IoTDs in adjacent clusters use different spectrum and therefore,

²In this chapter, for tractability, we work only with one cluster. However, the same approach is valid to be applied for all clusters.

the inter-cell-interference can be considered as noise. Orthogonal transmission is exploited to avoid interference among IoTDs in each cluster.

The distance from the IoTDs to the UAV, $d_{i \rightarrow U}^n$, and that from the UAV and BS, $d_{U \rightarrow S}^n$, in time-slot n , are calculated as follows:

$$d_{i \rightarrow U}^n = \sqrt{(x_i - x_U)^2 + (y_i - y_U)^2 + (H_U^n)^2}, \quad (3.1)$$

and

$$d_{U \rightarrow S}^n = \sqrt{(x_S - x_U)^2 + (y_S - y_U)^2 + (H_S - H_U^n)^2}, \quad (3.2)$$

where H_U^n is the altitude of the UAV in time-slot n .

3.2.1 IoT-UAV-BS Channel Model

Depending on whether there is a LoS link between an IoTD and the UAV, and that between the UAV and the BS the received signal power is different. The probability of having a LoS depends on the actual environment and the distance between the IoTD and UAV and between the UAV and BS. The probability of establishing a LoS link between IoT-to-UAV is given by [50]

$$\mathbb{P}_{i \rightarrow U} = \frac{1}{1 + C_2 e^{-C_1(\theta_{i,U}^n - C_2)}}. \quad (3.3)$$

Similarly, between UAV-to-BS

$$\mathbb{P}_{U \rightarrow S} = \frac{1}{1 + C_4 e^{-C_3(\theta_{U,S}^n - C_4)}}, \quad (3.4)$$

where $\theta_{i,U}^n$ and $\theta_{U,S}^n$ are the elevation angle of IoT-to-UAV and UAV-to-BS, respectively. C_1 , C_2 , C_3 and C_4 are environment-dependent variables, which are varying from one topology to another, i.e., communication surrounding such as the building blockage and density. $\theta_{i,U}^n$ and $\theta_{U,S}^n$ are determined by

$$\theta_{i,U}^n = \arctan \frac{H_U^n}{\sqrt{(x_i - x_U)^2 + (y_i - y_U)^2}}, \quad (3.5)$$

and

$$\theta_{U,S}^n = \arctan \frac{\sqrt{(H_S - H_U^n)^2}}{\sqrt{(x_S - x_U)^2 + (y_S - y_U)^2}}. \quad (3.6)$$

Thus, the path-loss of IoT-to-UAV and UAV-to-BS, respectively, follows

$$\Delta_{i \rightarrow U}^n = 20 \log \left(\frac{4\pi f_c (d_{i \rightarrow U}^n)}{c} \right) + C_5 \mathbb{P}_{i \rightarrow U} + C_6 (1 - \mathbb{P}_{i \rightarrow U}), \quad (3.7)$$

and

$$\Delta_{U \rightarrow S}^n = 20 \log \left(\frac{4\pi f_c (d_{U \rightarrow S}^n)}{c} \right) + C_7 \mathbb{P}_{U \rightarrow S} + C_8 (1 - \mathbb{P}_{U \rightarrow S}), \quad (3.8)$$

where C_5 , C_6 , C_7 and C_8 are attenuation factors that depend on the environment. f_c denotes the carrier frequency (MHz), and c denotes the speed of light (m/s).

All IoTDs and the UAV are assumed to transmit with power P_I and P_U , respectively. The signal-to-noise ratio (SNR) at the UAV and at the BS in time-slot n can be respectively expressed as

$$\Upsilon_{i \rightarrow U}^n = \frac{P_I 10^{-\frac{\Delta_{i \rightarrow U}^n}{10}}}{N_o}, \quad (3.9)$$

and

$$\Upsilon_{U \rightarrow S}^n = \frac{P_U 10^{-\frac{\Delta_{U \rightarrow S}^n}{10}}}{N_o}. \quad (3.10)$$

Let S_i^n and S_U^n be the achievable rate (in bps/Hz) that is delivered to the deployed UAV and BS, respectively. Given the available channel bandwidth W (in Hz), the achievable rate, S_i^n and S_U^n , can be expressed as the follows:

$$S_i^n(H_U^n) = W \log (1 + \Upsilon_{i \rightarrow U}^n), \quad (3.11)$$

and

$$S_U^n(H_U^n) = W \log (1 + \Upsilon_{U \rightarrow S}^n). \quad (3.12)$$

In this scenario, depending on Channel State Information from IoTDs to the UAV (CSIU), and that from the UAV to the BS (CSIB), only a part of the status-update packet can be successfully recovered/decoded, which is random. In order to achieve a reliable transmission, S_i^n and S_U^n should be strictly greater than or equal to S_{th} , where S_{th} is the minimum achievable rate to ensure reliable

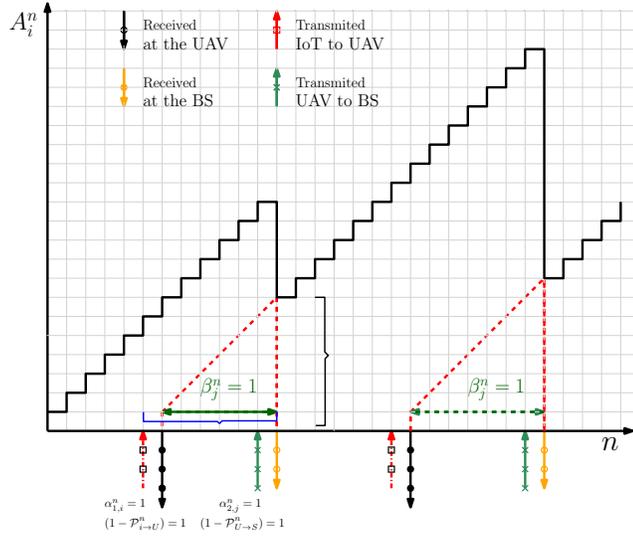


Fig. 3.2: The evolution of AoI.

decoding.

Recall that the deployed UAV is equipped with a single antenna and operates in a half-duplex mode; hence, the service time can be divided into two processes: 1) *Uplink process*: where the deployed UAV is successfully able to reliably decode status-update packets of IoT i when $S_i^n(\Delta_{i \rightarrow U}^n) \geq S_{th}$, and a transmission failure occurs otherwise, and 2) *Downlink process*: where the BS is successfully able to reliably decode status-update packets from the deployed UAV when $S_U^n(\Delta_{U \rightarrow S}^n) \geq S_{th}$ and a transmission failure occurs otherwise.

Let $\alpha_{1,i}^n$ be a binary variable, which indicates that IoT i is scheduled in time-slot n to transmit its status-update, and 0 otherwise. A successful transmission with reliable decoding occurs to the deployed UAV when $\alpha_{1,i}^n \cdot \mathbb{1}[S_i^n(\Delta_{i \rightarrow U}^n) \geq S_{th}] = 1$.¹ Similarly, let $\alpha_{2,j}^n$ be a binary variable, which indicates that the packet on the virtual queue j is scheduled in time-slot n to be transmitted to the BS, and 0 otherwise. A successful reliable transmission occurs to the BS when $\alpha_{2,j}^n \cdot \mathbb{1}[S_U^n(\Delta_{U \rightarrow S}^n) \geq S_{th}] = 1$. With Time Division Multiple Access (TDMA), one packet at most is scheduled for transmission from the IoT to the deployed UAV or from the UAV to the BS at any given time-slot. Thus, in each time-slot, each UAV only schedules at most one IoT to transmit its status-update. Therefore,

¹ $\mathbb{1}[z]$ is the indicator function that equals unity if z is true.

the transmission scheduling should meet the constraint below:

$$\sum_{i=1}^M \alpha_{1,i}^n + \sum_{j=1}^{M'} \alpha_{2,j}^n \leq 1, \quad \forall n. \quad (3.13)$$

3.2.2 Definition of Age of Information

A single packet queuing discipline is assumed to be employed at both the IoTDs and the deployed UAV such that the older status-update packet is dropped and replaced with the newly arrived sample. A *per time-slot sampling* policy is considered for sampling the information, where each IoTD samples the status-update information at the beginning of each time-slot. Let b_i^n denotes the time elapsed at the UAV's virtual queue, Q_i , associated with IoTD i in time-slot n . Thus, the evolution of b_i^n can be written as

$$b_i^{n+1} = \begin{cases} 1, & \text{if } \alpha_{1,i}^n \cdot \mathbb{1}[S_i^n(H_U^n) \geq S_{th}] = 1, \\ b_i^n + 1, & \text{if } (\beta_j^n = 1) \wedge (\alpha_{1,i}^n \cdot \mathbb{1}[S_i^n(H_U^n) \geq S_{th}] = 0), \\ 0, & \text{otherwise (i.e., } \beta_j^n = 0), \quad // \text{empty buffer} \end{cases} \quad (3.14)$$

where β_j^n is a binary variable that is equal to 1 if the selected stream from virtual queue j has a non-empty queue, and 0 otherwise. Intuitively, the value of β_j^n changes to 0 only when the Head-of-Line status-update packet is successfully delivered to the BS and there is no newly arrival arrived on the same virtual queue. Thus, β_j^n can be written as

$$\beta_j^{n+1} = \begin{cases} 1, & \text{if } \alpha_{1,i}^n \cdot \mathbb{1}[S_i^n(H_U^n) \geq S_{th}] = 1, \\ 0, & \text{if } \beta_j^n \cdot \alpha_{2,j}^n \cdot \mathbb{1}[S_U^n(H_U^n) \geq S_{th}] = 1, \\ \beta_j^n, & \text{otherwise.} \end{cases} \quad (3.15)$$

Accordingly, the evolution of A_i^n of IoTD i can be written³

³For more tractable analysis, all virtual queues are initially assumed to be empty, that is, $\beta_j^0 = 0, \forall j$ and the initial values of AoI is neglected, that is, $A_i^0 = 0, \forall i$.

$$A_i^{n+1} = \begin{cases} b_i^n + 1, & \text{if } \beta_j^n \cdot \alpha_{2,j}^n \cdot \mathbb{1}[S_U^n(H_U^n) \geq S_{th}] = 1, \\ A_i^n + 1, & \text{otherwise.} \end{cases} \quad (3.16)$$

To better understand the definition of AoI, we provide an example in Fig. 3.2. The figure illustrates the evolution of AoI associated with one IoTD. The solid line represents the AoI of IoTD i and the dashed line denotes the elapsed time of the status update on the virtual queue, Q_i , of the UAV. As shown, the elapsed time, b_i^n , on the virtual queue of the UAV starts when a new status update is successfully received at the UAV. The elapsed time, b_i^n , is reset once the status update is successfully received at the BS and remains at zero before a new status update successfully arrives. AoI increases linearly at every time-slot between two successfully received updates at the BS and jumps downward to the elapsed time, b_i^n , when the status-update is received successfully. It is evident that the AoI of one IoTD is completely determined by the scheduling policy, the altitude of the UAV and Channel State Information. Thus, to obtain the AoI within the relay mission time, we use the EWSA $\frac{1}{N} \mathbb{E} \left[\sum_{n=1}^N \sum_{i=1}^M \xi_i A_i^n | A_i^0 = 0 \right]$, where ξ_i is a positive weight that denotes the relative importance of the application associated with IoTD i .

3.2.3 Optimization Problem Formulation

This chapter aims at optimizing communication scheduling and UAV altitude in order to minimize the Expected Weighted Sum AoI. For ease of notation, let us denote $\mathbf{L} = \{H_U^n, \forall n\}$ and $\mathbf{S} = \{\alpha_{1,i}^n, \alpha_{2,j}^n, \forall i, j, n\}$. Thus, our optimization problem is formulated as:

$$\begin{aligned} (\mathcal{O} \mathcal{P}): \quad & \min_{\mathbf{L}, \mathbf{S}} \frac{1}{N} \mathbb{E} \left[\sum_{n=1}^N \sum_{i=1}^M \xi_i A_i^n | A_i^0 = 0 \right] \\ \text{s.t. } \quad & \mathcal{C}1 : \alpha_{1,i}^n \in \{0, 1\}, \forall i, n, \\ & \mathcal{C}2 : \alpha_{2,j}^n \in \{0, 1\}, \forall j, n, \\ & \mathcal{C}3 : \beta_j^n \in \{0, 1\}, \forall j, n, \beta_j^0 = 0, \forall j, \\ & \mathcal{C}4 : \sum_{i=1}^M \alpha_{1,i}^n + \sum_{j=1}^{M'} \alpha_{2,j}^n \leq 1, \quad \forall n, \\ & \mathcal{C}5 : H_{min} \leq H_U^n \leq H_{max}, \forall n, \end{aligned}$$

Table 3.1: Table of Notations

Parameters	Description
\mathcal{M}	Set of IoTDs.
(x_U, y_U)	UAV horizontal coordinates.
$(x_i, y_i, 0)$	Position of IoTD i .
H_S	BS's altitude.
N	Total number of time slots.
δ_t	Time slot duration.
$C_1 - C_8$	Environment-dependent variables.
f_c	Carrier frequency.
c	Speed of light.
W	Channel bandwidth.
N_o	Noise power.
P_I	IoTD transmission power.
P_U	UAV's transmission power.
s_n, a_n, r_n	State, action, reward at time-slot n .
S_{th}	Minimum rate for reliable decoding.
Variables	Description
H_U^n	UAV's altitude at time-slot n .
$\alpha_{1,i}^n$	Indicates if IoTD i is scheduled for transmission at time-slot n .
$\alpha_{2,j}^n$	Indicates if virtual queue j is scheduled for transmission at time-slot n .
z_i^n	Indicates if reliable transmission is achieved between IoTD i and UAV at time-slot n .
g^n	Indicates if reliable transmission is achieved between UAV and BS at time-slot n .

$$\mathcal{C}6 : (3.14),$$

$$\mathcal{C}7 : (3.15),$$

$$\mathcal{C}8 : (3.16),$$

$$\mathcal{C}9 : |H_U^{n+1} - H_U^n| \leq V_{\max} \delta_t, n = 1, \dots, N - 1.$$

Constraint $\mathcal{C}5$ denotes the UAV altitude constraint, with H_{max} and H_{min} denoting the maximum⁴ and minimum altitude, respectively. Table 3.1 provides a summary of the variables and parameters used in the formulation. Finally, $\mathcal{C}9$ limits the traveled vertical distance by the UAV in one time slot based on its maximum speed.

Problem ($\mathcal{O} \mathcal{P}$) is hard to solve as a result of the intractability of the objective function, Eq.

⁴The maximum UAV altitude is limited to around 120 m in North America and most developing countries.

3.14, Eq. 3.15 and Eq. 3.16, for which an exact derivation cannot be achieved. In order to overcome this difficulty, we formulate a closed-form expression for an upper bound of b_i^{n+1} and A_i^{n+1} , see Appendix A for details, and reformulate ($\mathcal{O}\mathcal{P}$) as

$$\begin{aligned}
(\mathcal{O}\mathcal{P}.1): \quad & \min_{\mathbf{L}, \mathbf{S}, \mathbf{Z}, \mathbf{G}} \frac{1}{N} \mathbb{E} \left[\sum_{n=1}^N \sum_{i=1}^M \xi_i A_{i,ub}^n | A_{i,ub}^0 = 0 \right] \\
\text{s.t. } \quad & \mathcal{C}1: \alpha_{1,i}^n \in \{0, 1\}, \forall i, n, \\
& \mathcal{C}2: \alpha_{2,j}^n \in \{0, 1\}, \forall j, n, \\
& \mathcal{C}3: \beta_{j,ub}^n \in \{0, 1\}, \forall j, n, \beta_{j,ub}^0 = 0, \forall j, \\
& \mathcal{C}4: \sum_{i=1}^M \alpha_{1,i}^n + \sum_{j=1}^{M'} \alpha_{2,j}^n \leq 1, \quad \forall n, \\
& \mathcal{C}5: H_{min} \leq H_U^n \leq H_{max}, \quad \forall n, \\
& \mathcal{C}6.1: z_i^n \in \{0, 1\}, \forall i, n, \\
& \mathcal{C}6.2: z_i^n > \frac{S_i^n(H_U^n) - S_{th}}{\Lambda}, \forall n, i, \\
& \mathcal{C}6.3: z_i^n \leq 1 + \frac{S_i^n(H_U^n) - S_{th}}{\Lambda}, \forall n, i, \\
& \mathcal{C}6.4: b_{i,ub}^{n+1} \geq (b_{i,ub}^n + 1) \cdot \beta_{j,ub}^n \cdot (1 - \alpha_{1,i}^n \cdot z_i^n), \forall i, n, \\
& \mathcal{C}6.5: b_{i,ub}^{n+1} \geq \alpha_{1,i}^n \cdot z_i^n, \forall i, n, \\
& \mathcal{C}7.1: g^n \in \{0, 1\}, \forall n, \\
& \mathcal{C}7.2: g^n > \frac{S_U^n(H_U^n) - S_{th}}{\Lambda}, \forall n, \\
& \mathcal{C}7.3: g^n \leq 1 + \frac{S_U^n(H_U^n) - S_{th}}{\Lambda}, \forall n, \\
& \mathcal{C}7.4: \beta_{j,ub}^{n+1} \geq \beta_{j,ub}^n \cdot (1 - \alpha_{2,j}^n \cdot g^n), \forall j, n, \\
& \mathcal{C}7.5: \beta_{j,ub}^{n+1} \geq \alpha_{1,i}^n \cdot z_i^n, \forall i, j, n, \\
& \mathcal{C}8.1: A_{i,ub}^{n+1} \geq (b_{i,ub}^n + 1) \cdot (\beta_{j,ub}^n \cdot \alpha_{2,j}^n \cdot g^n), \forall i, j, n, \\
& \mathcal{C}8.2: A_{i,ub}^{n+1} \geq (A_{i,ub}^n + 1) \cdot (1 - (\beta_{j,ub}^n \cdot \alpha_{2,j}^n \cdot g^n)), \forall i, j, n, \\
& \mathcal{C}9: |H_U^{n+1} - H_U^n| \leq V_{max} \delta_t, n = 1, \dots, N-1.
\end{aligned}$$

Constraints $\mathcal{C}6.1$ – $\mathcal{C}6.5$ replace constraint $\mathcal{C}6$ where $\mathcal{C}6$ captures the time elapsed at the UAV’s virtual queue under the reliability condition. Constraints $\mathcal{C}7.1$ – $\mathcal{C}7.5$ replace constraint $\mathcal{C}7$. $\mathcal{C}7$ ensures that the device’s data is successfully received by the UAV iff the transmission is reliable. Constraints $\mathcal{C}8.1$ – $\mathcal{C}8.2$ replace constraint $\mathcal{C}8$.

Due to the randomness of the environment, $(\mathcal{O}\mathcal{P}.1)$ is a constrained stochastic optimization problem over the UAV operating time N . To the best of our knowledge, offline solutions are generally impractical since it is mathematically difficult to track the AoI over unknown channel condition. In fact, in the actual deployment of UAVs, obtaining complete information on channel quality (LoS/NLoS channels) depends on the location of the UAV and the surrounding environment (height/density of buildings). It also requires an excessive measurement, which is not easy to perform in practice. In addition, the LoS/NLoS links and the received signal strength may alter frequently as horizontal coordinates of the deployed UAV change with clustering. It is important to note that deploying the UAV with the free-space channel model is practically inaccurate and may result in considerable degradation of performance due to the blockage that exists in the actual environment.

We also observe that $(\mathcal{O}\mathcal{P}.1)$ is a mixed integer non-linear program (MINLP), which is generally hard to solve, due to the existence of the binary variables $\alpha_{1,i}^n$, $\alpha_{2,j}^n$, $\beta_{j,ub}^n$, z_i^n and g^n . In addition, dynamic programming method might be infeasible for such large scale problems. Hence, our problem is reformulated as MDP and online DRL is exploited to find the best policy to control the altitude of the UAV and the schedule between the IoTD to the UAV and between the UAV and the BS. The proposed online DRL approach does not rely on a prior knowledge of the channel conditions, where the environment variables are unknown, i.e., $(C_1 : C_8)$. In the next sections, the proposed solution methodology for solving our formulated problem is described.

3.3 Proposed Solution

Here, an AI-agent is deployed on the UAV which interacts with the BS and IoTDs in a sequence of actions. The AI-agent observes the rewards and steadily learns the optimal altitude and scheduling policy. At each time-slot n , the AI agent decides an action for the deployed UAV. To design a

framework for adjusting the altitude of the UAV and finding a scheduling policy is a non-trivial challenge. This is because the considered scenario is a hybrid discrete-continuous action space problem and the altitude and scheduling are also closely coupled with each other in the considered problem. These increase the difficulty of solving the problem. To tackle the first challenge, we approximate the hybrid space by discretizing the altitude of the UAV into discrete values and solving the problem with an efficient online DRL algorithm that uses Proximal Policy Optimization. [51].

The AI-agent may have to execute two actions simultaneously. For instance, the deployed UAV might need to adjust its altitude and at the same time, schedule an IoT. Most of the existing RL approaches however treat each action (altitude adjustment and scheduling) independently and thus, learn each action separately. Combining the two actions into one single action could be one way to deal with this challenge. However, for an environment with M IoTs that need to be scheduled as well as a UAV, with Q discrete values, whose altitude needs to be adjusted, a total of $(Q \times M)$ possible actions need to be considered. This could increase the difficulty of learning for the AI-agent. To maintain a small size of the action space, the altitude and scheduling actions are concatenated into a single action space and the AI-agent is allowed to execute one action per time slot. In other words, at a given time, the AI-agent will either adjust the altitude of the UAV or schedule a transmission. The AI-agent will learn to allocate more time slots for scheduling since it is not necessary to frequently adjust the altitude at every time slot.

In the following, a brief review is presented on Proximal Policy Optimization (PPO), a learning technique, that is suitable for online controlling of autonomous machines.

Background on Proximal Policy Optimization

In this chapter, we focus on policy-based RL algorithms as they have become prevalent and have shown significant improvements compared to state-of-the-art algorithms. The focus of policy-based algorithms is to build an estimator of the policy gradient and exploit a stochastic gradient ascend (SGA) in order to achieve the maximum rewards. The gradient of the objective function is defined as

$$\nabla J(\theta) = \mathbb{E}_{(s_t, a_t) \sim \pi_\theta} [\nabla \log \pi_\theta(a_t | s_t) A(s_t, a_t)], \quad (3.19)$$

where \mathbb{E} is the expected value. π_θ is the probability of policy θ selecting action a_n at given state s_n . $A(s_n, a_n)$ is the advantage estimate in time-slot n that is used to mitigate the high variance of the gradient. The advantage estimate $A(s_n, a_n)$ is given by

$$A^\pi(s_n, a_n) = R_\pi(s_n, a_n) - V_\pi(s_n), \quad (3.20)$$

where $V_\pi(s_n)$ is the state-value function of state s_n under policy π . R_π is the future discounted cumulative rewards.

In fact, there are two major problems associated with DRL. The first is *update instability* since the DRL algorithms are sensitive to step size parameter for the policy optimization. Choosing a step size that is too small makes learning (convergence) very slow while a step size that is too large drastically reduces the performance of the policy. The second is the *data inefficiency*, where the new policy is evaluated based on a completely new training data; thus, DRL requires a large amount of data to learn.

Trust Region Policy Optimization (TRPO) algorithm [52] overcomes above problems by limiting the update range of the policy. It exploits the Kullback Leibler divergence between the current and old policy distributions. TRPO proposes to optimize a surrogate objective function⁵ by applying the Kullback Leibler divergence constraint that can provide local improvements to the current policy at each iteration. The surrogate objective function is defined as

$$J(\theta) = \mathbb{E}_{(s_t, a_t) \sim \pi_{\theta_{old}}} \left[\frac{\pi_\theta(a_n | s_n)}{\pi_{\theta_{old}}(a_n | s_n)} A(s_n, a_n) \right]. \quad (3.21)$$

Proximal Policy Optimization (PPO) is proposed to overcome the high complexity of TRPO [51]. PPO replaces the hard constraint of TRPO by setting a boundary for the update $\frac{\pi_\theta(a_n | s_n)}{\pi_{\theta_{old}}(a_n | s_n)}$ within a target range. In order to achieve that, the surrogate advantage objective is clipped. The

⁵Surrogate objective function is not the same as the main objective function but it leads to the same result by applying importance sampling technique.

PPO-clip objective function can be written as

$$L^{CLIP}(\theta) = \mathbb{E}_n \left[\min \left(\frac{\pi_\theta(a_n|s_n)}{\pi_{\theta_{old}}(a_n|s_n)} A_{\pi_{\theta_{old}}}(s_n, a_n), \text{clip} \left(\frac{\pi_\theta(a_n|s_n)}{\pi_{\theta_{old}}(a_n|s_n)}, 1 + \varepsilon, 1 - \varepsilon \right) A_{\pi_{\theta_{old}}}(s_n, a_n) \right) \right], \quad (3.22)$$

where ε is the clip fraction used to control the clip range. In practice, PPO usually is implemented in Actor-Critic framework, where more objective functions are added to the surrogate objective. The overall objective function is given by

$$L^{total}(\theta) = \mathbb{E}_n [L^{CLIP}(\theta) - K_1 L_n^{VF}(\theta) + K_2 S(\pi|s_n)], \quad (3.23)$$

where K_1 and K_2 are loss coefficients. L_n^{VF} and $S(\pi|s_n)$ denote the square error-loss for Critic network, $(V_\theta(s_n) - V_n^{targ})^2$, and entropy bonus respectively. The entropy bonus encourages the AI-agent to execute actions more unpredictably for exploration. Thus, the update of the objective is restricted by target region. Because of these advantages, we developed our solution approach based on PPO. For more information on Proximal Policy Optimization, the reader is referred to [51]. The next subsection presents the system state representation as well as the rewards and penalties associated with the agent's actions.

3.3.1 MDP Formulation

We first formulate the scheduling and altitude decision problem as an MDP, and design a PPO algorithm, in order to find the best policy that governs the altitude of the deployed UAV and the scheduling decision within unknown environment. MDP is usually represented by 4-tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R})$, in which

- A set of states \mathcal{S} which includes all possible states s_n at any time-slot n , where $s_n \in \mathcal{S}$.
- A set of actions \mathcal{A} which includes all feasible actions a_n at any time-slot n , where $a_n \in \mathcal{A}$.
- A transition distribution \mathcal{P} , where $P(s_{n+1}|s_n, a_n)$, $s_n, s_{n+1} \in \mathcal{S}$, $a_n \in \mathcal{A}$ is the probability that leads to the new state, s_{n+1} , after executing an action a_n at the state s_n .

- A reward distribution \mathcal{R} where, $P(r_n|s_n, a_n)$, $s_n \in \mathcal{S}$, $r_n \in \mathcal{R}$, $a_n \in \mathcal{A}$ is a measurable function which gives an immediate reward after an action a_n is chosen in a state s_n at time-slot n

Under the MDP framework, we will elaborate the state, action, and reward functions sequentially as follow

1) **State** \mathcal{S} : The state at time slot n is defined as $s_n = (A^n, \beta^n, b_i^n, S_i^n, S_U^n)$, where:

- $A^n = (A_1^n, \dots, A_i^n, \dots, A_M^n)$: a vector of size M containing the AoI of all the IoTDs at time-slot n .
- $\beta^n = (\beta_1^n, \dots, \beta_j^n, \dots, \beta_M^n)$: a vector of size M containing the status of the virtual queue $\forall i \in M$, at time-slot n .
- $b^n = (b_1^n, \dots, b_i^n, \dots, b_M^n)$: a vector of size M containing the time elapsed at the UAV's virtual queue associated with all IoTDs, at time-slot n .
- $S^n = (S_1^n, \dots, S_i^n, \dots, S_M^n)$: a vector of size M containing the achievable rate $\forall i \in M$, at time-slot n .
- S_U^n : the status-update size that could be delivered to the BS, at time-slot n

2) **Action** \mathcal{A} : At each step-slot n , the deployed UAV executes an action a^n , The current composite action a^n is denoted by

$$a_n = (\xi^n, \psi^n, \kappa^n), \quad (3.24)$$

where ξ^n , ψ^n and κ^n are defined and interpreted as follows:

- Defined row vector $\xi^n = [\alpha_{1,1}^n, \dots, \alpha_{1,i}^n, \dots, \alpha_{1,M}^n]$, where $\alpha_{1,i}^n$ represents the schedule control from IoTD i to the UAV, and each element $\alpha_{1,i}^n \in 0, 1$ where $\alpha_{1,i}^n = 0$ means the IoTD is not scheduled to transmit its status-update at time-slot n , and $\alpha_{1,i}^n = 1$ means it is scheduled to transmit its status-update.
- Defined row vector $\psi^n = [\alpha_{2,1}^n, \dots, \alpha_{2,i}^n, \dots, \alpha_{2,M}^n]$, where $\alpha_{1,i}^n$ represents the schedule control from UAV's virtual queue j to the BS, and each element $\alpha_{2,j}^n \in 0, 1$ where $\alpha_{2,j}^n = 0$ means the

status-update on virtual queue j is not scheduled to transmit its status-update at time-slot n , and $\alpha_{2,j}^n = 1$ means it is scheduled to transmit to the BS.

- Defined row vector $\kappa^n \in (K_{Up}, K_{Down})$, where K_{Up} represents the upward flight, and $K_{Up} \in 0, 1$, where $K_{Up} = 1$ means the UAV adjusted its altitude 10m upward at time-slot n , and $K_{Up} = 0$ means no action towards upward flight. $K_{Down} = 1$ means the UAV adjusted its altitude 10m downward at time-slot n , and $K_{Down} = 0$ means no action towards downward flight. In other words, when $K_{Up} = K_{Down} = 0$ that means the UAV is hovering.

Hence, the deployed UAV adjusts its altitude and decides which IoT to transmit its status-update or schedule the transmission from UAV's virtual queue to the BS.

3) **Reward \mathcal{R}** : The immediate reward r_n is the sum of the following normalized quantities:

- Penalty from the IoT network when the UAV collects status-updates from IoTs with high AoI: the value of this penalty is proportional to the summation of all AoI. As a result, the AI-agent learns to minimize this penalty by optimizing scheduling decisions between IoT-to-UAV and UAV-to-BS and altitude control of the UAV in order to relay the old status-update (i.e., highest AoI) from IoTs.
- Penalty incurred on network if the deployed UAV flies outside the given altitude constraint h_{min} and h_{max} : the AI-agent learns how to adjust the altitude of the UAV to be within altitude range.

Generally, an MDP problem can be solved using Dynamic Programming (DP) algorithms. However, since the UAV is deployed with no prior knowledge on the environment, then DP or LP algorithms cannot be leveraged to solve the formulated MDP problem. Thus, to solve the formulated MDP problem in absence of the state transition probabilities \mathcal{P} of the Markov model, an online model-free DRL algorithm is employed in the next subsection.

3.3.2 Proximal Policy Optimization Proposed Solution

Our PPO-based solution to altitude control and scheduling policy is described in this section. As previously mentioned, our main objective is to find the online control policy that governs the

altitude and scheduling policy of the UAV at each time-slot to minimize the EWSA. To solve the control problem in the absence of the channel conditions and state transition probabilities of our MDP model, we leverage the online DRL algorithm where the AI-agent interacts with the wireless network environment and learns the control policy online. To solve our problem, we adapt the PPO, which is a stability algorithm (in term of Actor training) with outstanding performance. To obtain the online control policy, the proposed PPO algorithm, presented in Algorithm 1, is applied. The proposed algorithm can be divided into three parts as follows:

The initialization phase is the first part. After defining the input and output of the algorithm (Lines 1-2), the proposed algorithm randomly initializes the deep neural networks (NNs) parameter θ , where Deep NNs have the same structure. An actor and a critic network with random weights (Lines 3-4) are initialized. Training is conducted over multiple iterations at which the proposed algorithm alternates between sampling phase (lines 6-14) and optimization/exploration phase (lines 18-22). Each iteration corresponds to several episodes (actors) and each episode corresponds to multiple trajectories (i.e., sequence of states, actions and rewards).

The sampling phase is the second part (lines 6-17). In this phase, the AoI, time elapsed, queue status, and the achievable rate between IoT-to-UAV and UAV-to-BS, $(A_i^n, \beta_i^n, b_i^n, S_i^n, S_U^n)$ are observed by the UAV (lines 8-9). $\mathcal{C}3$, $\mathcal{C}6$, $\mathcal{C}7$ and $\mathcal{C}8$ of $(\mathcal{O}\mathcal{P})$ are guaranteed in Line 9 where the states of the MDP are defined. In addition, $\mathcal{C}9$ is also guaranteed by the definition of the action space κ^n in the MDP. The UAV then decides the best control policy, according to sampled action a_l from the policy $\pi_{\theta_{old}}$ (line 10). The sampled action, a_l , represents the current altitude of the UAV and scheduling status for each IoT and virtual queue. $\mathcal{C}1$, $\mathcal{C}2$ and $\mathcal{C}4$ of $(\mathcal{O}\mathcal{P})$ are guaranteed in Line 11 where the deployed UAV decides which IoT to transmit its status-update or schedule the transmission from UAV's virtual queue to the BS. In this step the algorithm assigned a binary variable "1" to the selected IoT or virtual queue and assigned "0" for other IoTs and virtual queues. During the sampling phase, the online algorithm guides the AI-agent to avoid actions that violate the altitude constraint (i.e., flies outside the allowable altitude) by applying a specific penalty to the reward (Lines 12-14), where, a penalty p is deducted from the overall reward, and the corresponding altitude action of the UAV is cancelled. Thus, the UAV altitude constraint $\mathcal{C}5$ is guaranteed in these Lines.

Algorithm 4: Proposed PPO Solution to obtain dynamic UAV altitude control and scheduling policy.

```

1 Input: Clip threshold  $\varepsilon$ , discount factor, learning rate, Adam optimizer parameters and
   wireless communication parameters;
2 Output: The altitude control of UAV and scheduling policy.
3 Initialize the Neural Networks parameter  $\theta$  randomly.
4 Initialize the sampling policy  $\pi_{\theta_{old}}$  with  $\theta_{old} \leftarrow \theta$ .
5 for Iteration = 0,1,.. do
6   for l = 0,1,.. L do
7     ▷ //Sampling with  $\pi_{\theta_{old}}$ //
8     for i = 0,1,.. M do
9        $\lfloor$  Observe:  $(A_i^l, \beta_i^l, b_i^l, S_i^l, S_U^l)$ ,
10      Sample: sample action  $a_l \sim \pi_{\theta_{old}}$ .
11      Execute: execute the action  $a_l$  that specifies the altitude of the UAV, the scheduling
           IoT-to-UAV and the scheduling UAV-to-BS.
12      if UAV flies outside the allowable altitude range  $h_{min}$  and  $h_{max}$  then
13         $\lfloor$   $r_l = r_l - P$ .
14         $\lfloor$  Cancel the movement of UAV and update  $s_{l+1}$ .
15      Evaluate: obtain the weighted sum AoI,  $r_l$ , and new state  $s_{l+1}$ .
16      Cache: store the trajectory  $(s_l, a_l, r_l, s_{l+1})$  in policy training  $f_l$ .
17      Compute advantage estimate  $\forall L$ .
18   for epoch = 0,1,.. do
19     ▷ //Optimizing  $\pi_{\theta}$  and Exploring//
20     Compute the PPO-Clip objective function (3.22).
21     Fit the value network via stochastic gradient descent with ADAM  $(V_{\theta}(s_n) - V_n^{targ})^2$ ,
22     Optimize the the overall objective function (3.23), and update the policy via SGA
           with ADAM, i.e.,  $\theta \leftarrow \arg \max_{\theta} L^{total}(\theta)$ .
23   Synchronize the sampling policy with  $\theta_{old} \leftarrow \theta$ .
24   Drop the stored data.

```

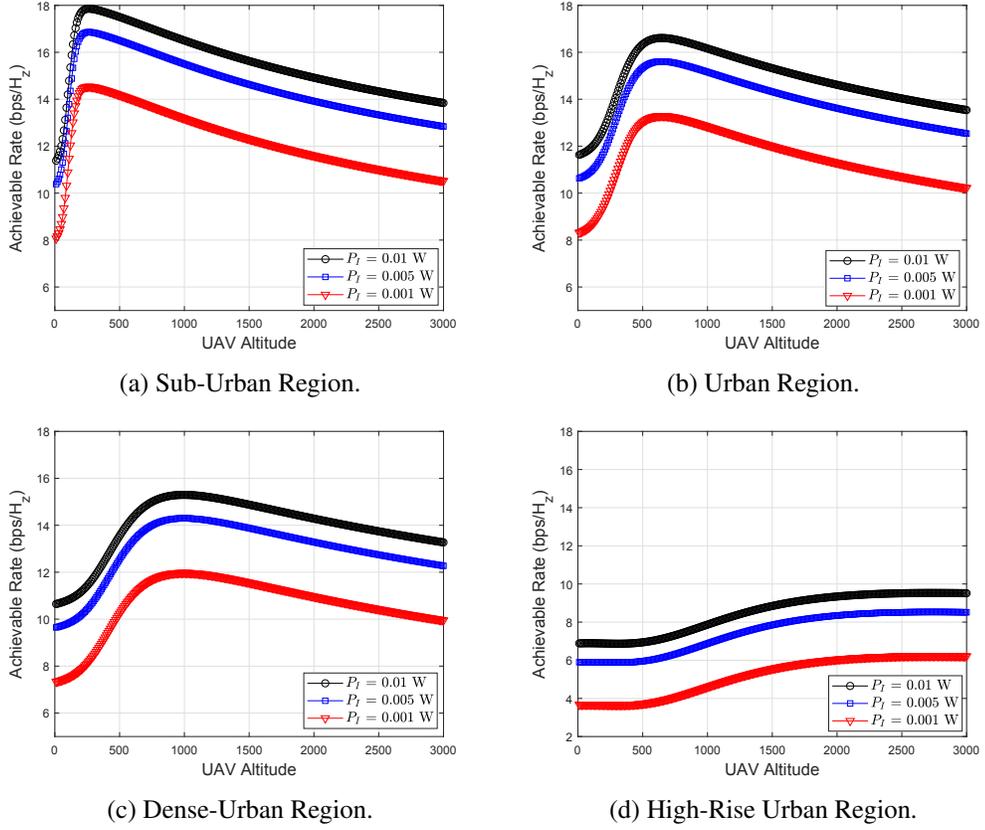


Fig. 3.3: Achievable Rate versus UAV's Altitude for IoT-to-UAV Communication.

After executing the current action (line 11), the UAV evaluates the expected reward (line 15), which represents the sum of the AoI for all IoTDs. In this phase, we collect $G \times L$ trajectories for training, represented as $(s_0, a_0, r_0, s_1, a_1, \dots)$, where G is the number of episodes and L is the total number of rollout steps each episode takes between updates. The trajectories' data are stored for the next phase (line 16). In order to achieve parallel processing among the episodes, a Message Passing Interface protocol is employed. Last but not least, the estimations of the advantage function for each rollout step are computed (line 17), according to Eq. 3.20, to achieve efficient training.

The optimization phase is the third part (lines 18-22). The network parameter θ of the policy π_θ is updated for each epoch. The PPO Clip objective function in each epoch is computed according to Eq. 3.22. The policy π_θ is optimized or improved by conducting SGA on the stored sampled data based on the PPO Clip objective function. After optimizing the network parameter, the policy is updated (lines 21-22), according to the overall objective function in Eq. 3.23. A random policy

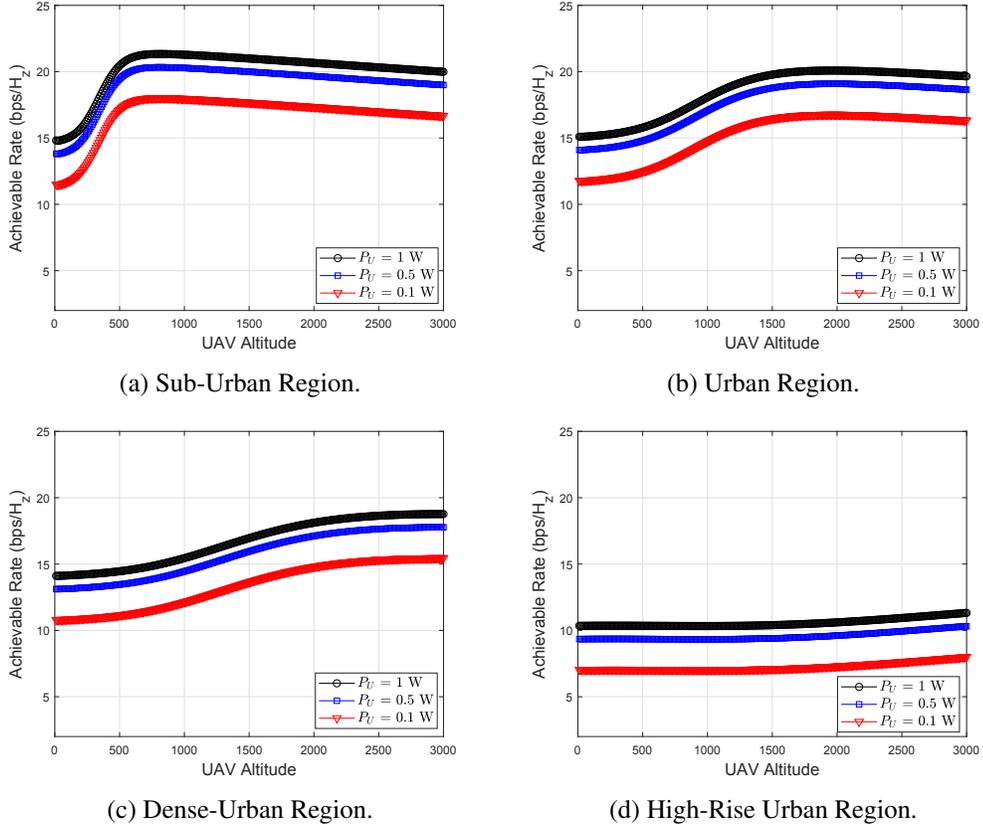


Fig. 3.4: Achievable Rate versus UAV's Altitude for UAV-to-BS Communication.

is exploited in this phase, according to the entropy bonus in Eq. 3.23 for exploration. Finally, the sampling policy $\pi_{\theta_{old}}$ is updated with the policy π_{θ} and the stored samples are removed (lines 23-24). The next iteration then begins.

3.3.3 PPO Analysis

Complexity

In this subsection, the complexity analysis is discussed. In practice, PPO usually is implemented in Actor-Critic framework, where more objective functions are added to the surrogate objective. The complexity of DRL approaches such as Actor-Critic framework is usually computed by the number of multiplications in each iteration. Therefore, based on [53], the total computational complexity for the fully connected layers can be expressed as the number of multiplications: $O(\sum_{p=1}^{P-1} n_p \cdot n_{p-1})$, where n_p is the number of neural units in the p -th hidden layer. The number of multiplications at

input and output layers is insignificant compared with the number of multiplications at the hidden layers, therefore, the number of multiplications at input and output layers is ignored in our analysis. In this chapter, we design the PPO architecture with the same number of neural units, denoted by n_p , in all hidden layers. Therefore, the complexity of the proposed PPO can be reduced to $O((P-1).n_p^2) \sim O(n_p^2)$.

Convergence

In general, the convergence of a neural network (NN) is challenging and hard to be analytically analyzed [54]. The reason lies in that the convergence of a NN is highly dependent on DRL hyper-parameters, in which the quantitative relationship between the NN convergence and the hyper-parameters is sophisticated. Therefore, a reasonable choice of the hyper-parameters is required in order to achieve the convergence. In fact, we tried various values for PPO hyper-parameters in the process of debugging, we found the best values are the ones that are given in the simulation parameters table while changing these values was often counterproductive. Similar to [54], in this chapter, we limit our analysis of convergence to simulations (see Fig. 3.5) where it is observed that our PPO algorithm converges under a reasonable choice of the hyper-parameters.

3.4 Simulation Results and Discussion

In this section, we conduct a series of simulations to evaluate the performance of the proposed algorithm. Firstly, we describe the simulation parameters and system settings and then present results and discussions. The main input parameters are listed in Table 3.2.

Simulation Setup

Unless otherwise specified, a square area of 1 km \times 1 km is considered as a single cluster for ease of illustration. Multiple IoTds were distributed randomly within the 2D-field. A single UAV is deployed at the center of the given area to relay multiple traffic streams from IoTds to the BS located at (2000,500,25) m. Assume that all the IoTds have the same transmission power and all devices have the same weight, which normalized to unity. All IoTds can communicate with the

Table 3.2: Simulation Parameters in UAV-Relay assisted IoT Networks

Parameter	Value
UAV vertical Max Speed, ω_{max}	10m/s
Geographical area size	1Km ²
Activation Functions	Softmax and Tanh
Minimum flying altitude of UAV, h_{min}	10m
Total number of time slots, N	300
Maximum flying altitude of UAV, h_{max}	1000m
Learning Rate	0.001
Reward Discount	0.9
Number of Hidden Layers for Networks	3
Number of Neurons	64
Loss Coefficients K_1 and K_2	0.5 and 0.01
Update Policy Length, L	240
Total number of Epochs	2000
Number of Episodes (actors)	4
Clip Fraction, ϵ	0.2
Optimizer Technique	Adam
UAV transmission power, P_U	30dBm
IoT transmission power, P_I	10dBm
Channel Gain, γ_0	-50dB
Noise Power, N_o	-110dBm

deployed UAV at different rates according to channel conditions. The results are collected after the training phase (3M samples) and each sample corresponds to a snapshot of the IoT network at a particular time slot. Similar to [55], CSIU and CSIB are obtained for both IoT-to-UAV and UAV-to-BS.

For each network (that is, the actor and critic networks), all simulations are run for fully connected three-layer neural networks that comprise of 64 neurons in each layer. The *hyperbolic tangent* (tanh) function is utilized for activation of both networks while *Softmax* is used in the last layer. The generated samples are used to train the deep neural network by utilizing PyTorch Deep Learning library to determine an optimal policy for the deployed UAVs. After establishing the altitude control and scheduling policy from the proposed algorithm, another sample set is used to test the performance of the proposed algorithm.

Benchmark Schemes

To the best of our knowledge, there is no existing approach that aims to solve a similar problem in UAV assisted IoT networks; thus, for the sake of comparison, we develop two other baseline approaches:

- *Random Deployment with Random Scheduling (RDRS)*: In the RDRS scheme, at each time slot, the deployed UAV randomly changes its vertical movement. Also, the UAV either randomly selects an IoTD to upload its status update packets to the UAV or to the BS. Meanwhile, if there is no status update packet in the UAV's virtual queue, then the UAV abandons this action and randomly selects another action.
- *Heuristic Deployment with Greedy Scheduling (HDGS)*: In the HDGS approach, the deployed UAV iteratively searches for the lowest height that satisfies the reliability constraint of the BS. Then, at each time slot, the UAV selects an IoTD with the highest AoI to upload its status update packets to the UAV. The UAV selects packets from the virtual queue to be uploaded to the BS in the next consecutive time slot. Meanwhile, if the reliability constraint of the UAV is not satisfied, the UAV selects the next IoTD with the highest AoI.

Results and Discussions

Before delving into the performance of PPO algorithm, we first investigate the impact of the UAV's altitude on the achievable rate under different environments. The simulation results are demonstrated in Fig. 3.3 for a single IoTD located 1km from the deployed UAV. As depicted in Fig. 3.3 and 3.4, the achievable rate curves rise to their maximum value and then decrease with increasing UAV's altitude. Thus, the required achievable rate and environment that the UAV operates at indicate the best altitude of the UAV. The same behavior is observed for different environments. Detailed parameters regarding the environment are listed in Table 3.3. When the UAV flies at the optimal altitude with respect to the IoTD, the path loss between the UAV and the BS increases because of obstacles blocking the way. When the UAV flies at the optimal altitude with respect to the BS, the path loss between the UAV and the IoTD increases due to longer distance. It was also observed that the achievable rates in sub urban and urban environments are larger than in the dense

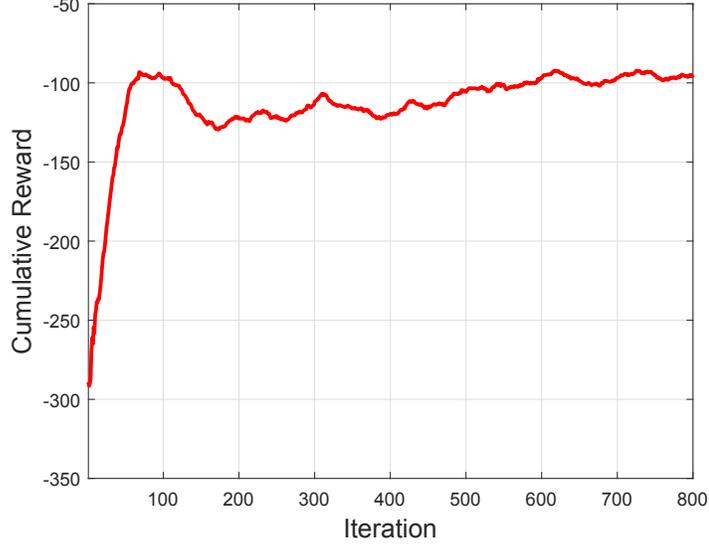


Fig. 3.5: Accumulated reward vs iterations.

urban and high rise urban environments due to the presence of more obstacles such as buildings. As the transmission power is further increased, a higher performance is achieved. The findings here show that based on the environment, attaining a certain target performance requires the optimization of the altitude of the UAV. Therefore, the AoI is strongly dependent on the optimum altitude of the UAV under specific conditions of the environment.

Next, the convergence performance of the proposed PPO versus the number of iterations is studied. The convergence is evaluated with $M = 20$ IoTDS and $S_{th} = 15\text{bps/Hz}$ in Fig. 3.5. As presented in the figure, the cumulative reward increases relatively quickly at the beginning of learning after which the increase becomes relatively slow. The reason is that, at the beginning of the iterations, the AI agent learns the altitude violation of the UAV such as minimum and maximum allowable altitude. Moreover, many IoTDS are not yet properly scheduled to transmit their status update packets to the UAV and from the UAV to the BS. This is because the UAVs have not yet learned the suitable scheduling policy in the deployed environment in order to attain the required reliability that minimizes the EWSA. The trained AI agent can significantly enhance the defined reward with each iteration. This improvement gradually becomes less obvious when the AI-agent is well trained about the environment and it starts to effectively adapt the scheduling policy.

To better understand how the action-space affects the performance of the proposed algorithm,

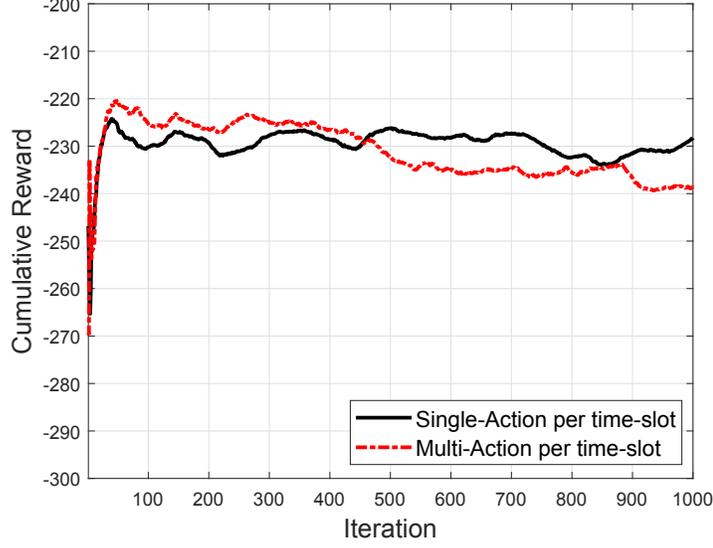


Fig. 3.6: Accumulated reward vs iterations.

an AI-agent is trained for multiple actions (that is, concurrent actions) per time slot and results are compared to those for a single action per time slot. For the evaluation, $M = 50$ is considered as the number of IoTDs. For multiple actions per time slot, all possible combinations of actions are modeled as separate actions. The action space reaches $(2 * M * 3)$ actions, where 2 represents the scheduling decision (that is, IoTD to UAV and UAV to BS) and 3 represents the altitude control action (that is, flying up, down and hovering). As shown in Fig. 3.6, due to a large action space, it is harder for the AI-agent to learn the value of each of the true actions in multiple actions per time slot compared to single action representations. A similar observation has been reported in [56]. It can be concluded that the suggested single action per time slot approach achieves better performance after a finite number of iterations.

Table 3.3: List of Parameters for different environments.

Parameter	Sub-Urban	Urban	Dense-Urban	High-Rise Urban
c_1, c_3	0.43	0.16	0.11	0.08
c_2, c_4	4.88	9.61	12.08	27.23
c_5, c_7	0.1	1.6	1	2.3
c_6, c_8	21	20	23	34

The plot, Fig. 3.7, depicts the impact of learning on the UAV altitude for single and multiple

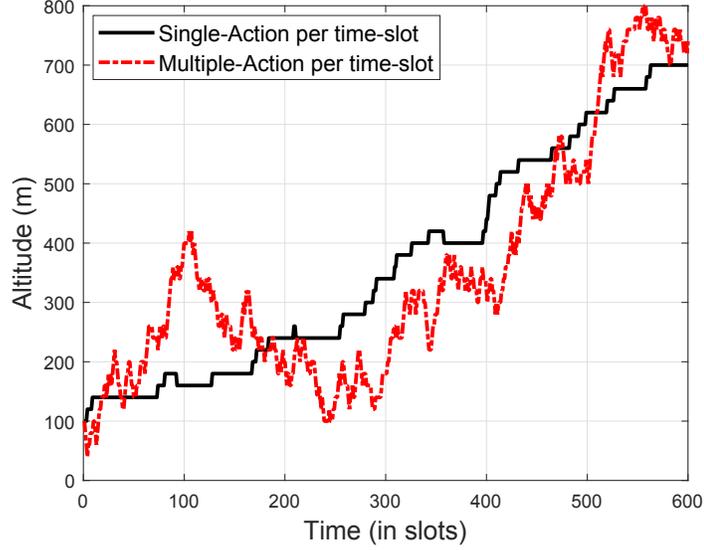
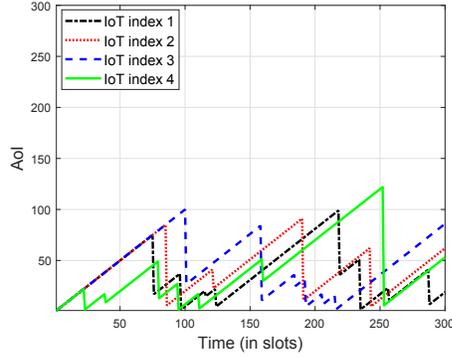


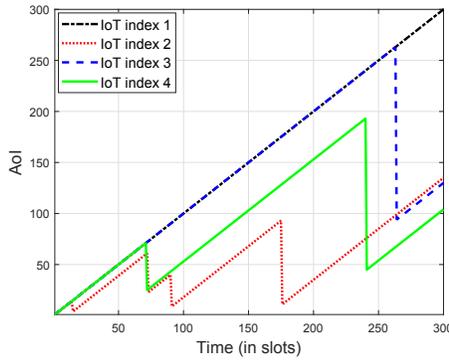
Fig. 3.7: UAV altitude vs time.

actions per time slot. The UAV is deployed initially at an altitude of 100m in the urban region and the minimum achievable rate to ensure reliable transmission is set to $S_{th} = 15\text{bps/Hz}$ for $M = 20$. It is evident that there is a certain range of altitude, also indicated in Fig. 3.3(b) and 3.4(b), that satisfies the reliability constraint between IoTD to UAV and from UAV to BS. Single and multiple actions per time slot techniques enable the adjustment of the altitude of the UAV within the optimum altitude range in order to establish effective communication links. However, due to insufficient learning for multiple actions per time slot, the AI agent takes wrong decisions while adjusting the altitude of the UAV. For example, the altitudes for the duration do not satisfy the reliability constraint for both the UAV and the BS.

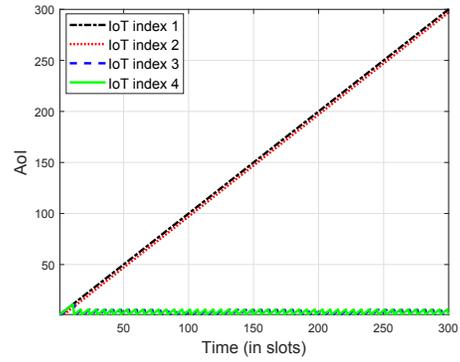
In Fig. 3.8, the AoI evolution over time for all approaches is presented for a selected set of four IoTDs in a network of 20 IoTDs. It can be observed that the AoI evolution can be drastically different for the different policies. By leveraging the PPO algorithm, the AoI of the four IoTDs is much smaller than that of the baseline approaches. This is understandable since, as explained above, the AI agent learns how to adjust the altitude of the UAV within the allowable altitude range to establish an effective communication link to an IoTD with the highest AoI value. Transmission failures on the links between IoTD to UAV and UAV to BS increase for the baseline approaches because the UAV is unable to efficiently adjust its altitude to satisfy the reliability constraint of the



(a) PPO.



(b) RDRS.



(c) HDGS.

Fig. 3.8: The performance comparison of different policies for a sample of four IoTDs.

BS and UAV. Furthermore, the HDGS approach, on the one hand, significantly decreases the AoI for some IoTDs. On the other hand, it increases the AoI to the maximum for other IoTDs. This is because the HDGS approach only schedules transmission for IoTDs that satisfy the reliability constraint for both links (IoT-to-UAV and UAV-to-BS).

To evaluate the effectiveness of the proposed algorithm, the impact of the number of IoTDs on the PPO approach compared to the RDRS and HDGS approaches is studied. A UAV is deployed to relay the status update, where the minimum achievable rate to ensure reliable transmission is set to $S_{th} = 15\text{bps/Hz}$. As shown in Fig. 3.9, the proposed PPO algorithm is able to minimize the EWSA for a lower number of IoTDs since each IoTD enjoys more service. In contrast, as the number of IoTDs increases, the EWSA increases, as expected, since more scheduling is required to decrease the EWSA. Besides, the performance of the HDGS approach is shown to be higher than the RDRS. This is because for the HDGS approach, which uses the greedy scheduling policy always selects the

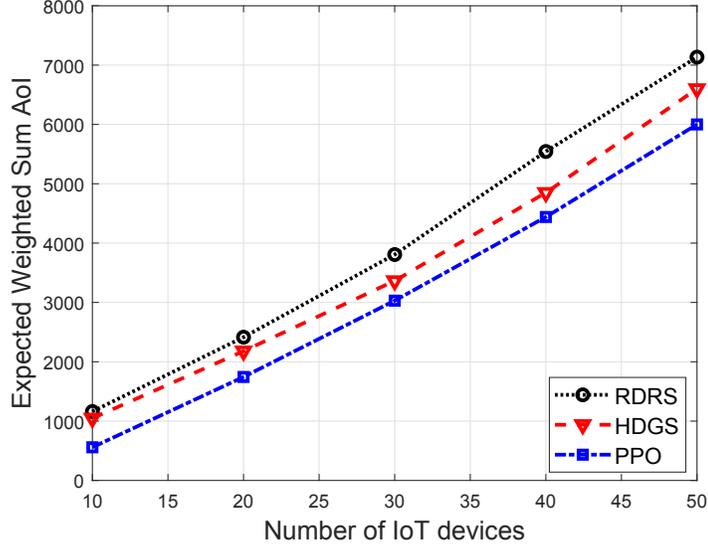


Fig. 3.9: Impact of number of IoTDs and comparisons.

IoTd with the highest AoI value at each time slot.

Last, the average age is another performance metric that we studied. Fig. 3.10 depicts the average age for a set of IoTds, where the minimum rate to ensure reliable transmission is set to $S_{th} = 15\text{bps/Hz}$ and $M = 20$. The average age of IoTd i within mission time N is captured by $\frac{1}{N} \sum_{n=0}^N A_i^n, \forall i$. Clearly, the proposed PPO algorithm minimizes the average AoI in the system compared to the other considered approaches. Also, the average age performance gap among the approaches is relatively high, which demonstrates the importance of optimizing the altitude of the UAV with scheduling. This finding justifies the robustness of the proposed algorithm in terms of minimizing average AoI.

3.5 Summary

This chapter addresses the problem of joint scheduling policy and dynamic UAV altitude control in UAV-assisted IoT networks that maintain the freshness of information status. A UAV is employed as a mobile relay between IoTds and the BS to minimize the Expected Weighted Sum Age-of-Information (EWSA) at the BS under unreliable transmission condition. It is assumed that before its deployment, the UAV has no prior knowledge of the channel and it can obtain instantaneous

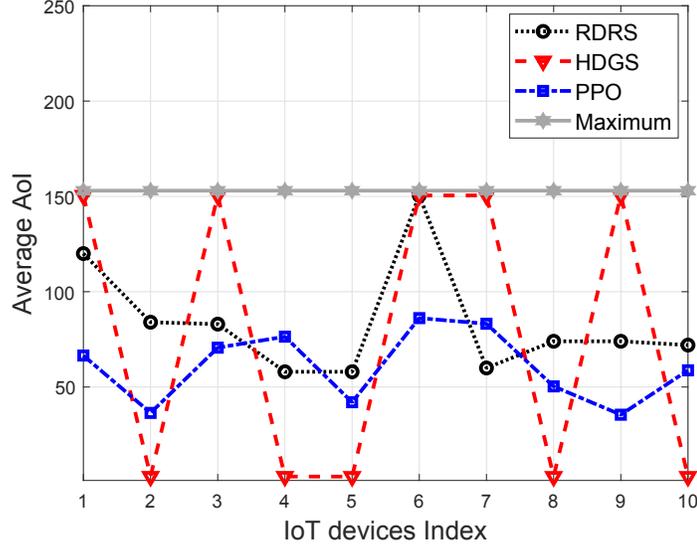


Fig. 3.10: Average age per IoTD.

IoT-to-UAV and UAV-to-BS CSI during its deployment. To maintain the freshness of information, the stochastic control problem is modeled as a Markov Decision Process and an online Deep Reinforcement Learning approach is proposed to obtain an optimal control policy that minimizes the EWSA. Numerical results demonstrate the effectiveness of the proposed online design, which was also verified by extensive comparisons with other baseline approaches.

3.6 Closed-Form and Upper Bound

Firstly, we derive an upper bound for the b_i^{n+1} and A_n^{n+1} as

$$b_{i,ub}^{n+1} \geq (b_{i,ub}^n + 1) \cdot (\beta_j^n) \cdot \left(1 - \alpha_{1,i}^n \cdot \mathbb{1}[S_i^n(\Delta_{i \rightarrow U}^n) \geq S_{th}]\right), \quad (3.25a)$$

$$b_{i,ub}^{n+1} \geq \alpha_{1,i}^n \cdot \mathbb{1}[S_i^n(\Delta_{i \rightarrow U}^n) \geq S_{th}]. \quad (3.25b)$$

$$b_{i,ub}^{n+1} \leq (b_{i,ub}^n + 1). \quad (3.25c)$$

$$A_{i,ub}^{n+1} \geq (b_{i,ub}^n + 1) \cdot \left(\beta_j^n \cdot \alpha_{2,j}^n \cdot \mathbb{1}[S_U^n(\Delta_{U \rightarrow S}^n) \geq S_{th}]\right), \quad (3.26a)$$

$$A_{i,ub}^{n+1} \geq (A_{i,ub}^n + 1) \cdot \left(1 - \left(\beta_j^n \cdot \alpha_{2,j}^n \cdot \mathbb{1}[S_U^n(\Delta_{U \rightarrow S}^n) \geq S_{th}] \right) \right), \quad (3.26b)$$

$$A_{i,ub}^{n+1} \leq (A_{i,ub}^n + 1), \quad (3.26c)$$

Similarly, Eq. 3.26 can be equivalently expressed as

$$\beta_{j,ub}^{n+1} \geq \beta_{j,ub}^n \cdot \left(1 - \alpha_{2,j}^n \cdot \mathbb{1}[S_U^n(\Delta_{U \rightarrow S}^n) \geq S_{th}] \right), \quad (3.27a)$$

$$\beta_{j,ub}^{n+1} \geq \alpha_{1,i}^n \cdot \mathbb{1}[S_i^n(\Delta_{i \rightarrow U}^n) \geq S_{th}], \quad (3.27b)$$

Secondly, for more traceability form, we introduce a new binary variable z_i^n that takes the value of 1 if the reliable transmission condition is achieved between the IoTD i and the UAV and equals 0 otherwise; similarly, the new binary variable g^n that takes the value of 1 if the reliable transmission condition between the UAV and the BS is achieved and equals 0 otherwise. By introducing the well-known big-number technique, $\Lambda \gg 1$, the binary variables $\mathbf{Z} = [z_i^n, \forall n, i]$ and $\mathbf{G} = [g^n, \forall n]$ can be expressed as

$$z_i^n > \frac{S_i^n(\Delta_{i \rightarrow U}^n) - S_{th}}{\Lambda}, \quad \forall n, i, \quad (3.28a)$$

$$z_i^n \leq 1 + \frac{S_i^n(\Delta_{i \rightarrow U}^n) - S_{th}}{\Lambda}, \quad \forall n, i, \quad (3.28b)$$

$$g^n > \frac{S_U^n(\Delta_{U \rightarrow S}^n) - S_{th}}{\Lambda}, \quad \forall n, \quad (3.29a)$$

$$g^n \leq 1 + \frac{S_U^n(\Delta_{U \rightarrow S}^n) - S_{th}}{\Lambda}, \quad \forall n, \quad (3.29b)$$

where Λ is a large number that is used to ensure the validity of the above equations.

Finally, $b_{i,ub}^{n+1}$, $A_{i,ub}^{n+1}$ and $\beta_{j,ub}^{n+1}$ can be rewritten as

$$b_{i,ub}^{n+1} \geq (b_{i,ub}^n + 1) \cdot \beta_{j,ub}^n \cdot (1 - \alpha_{1,i}^n \cdot z_i^n), \quad (3.30a)$$

$$b_{i,ub}^{n+1} \geq \alpha_{1,i}^n \cdot z_i^n. \quad (3.30b)$$

$$A_{i,ub}^{n+1} \geq (b_{i,ub}^n + 1) \cdot (\beta_{j,ub}^n \cdot \alpha_{2,j}^n \cdot g^n), \quad (3.31a)$$

$$A_{i,ub}^{n+1} \geq (A_{i,ub}^n + 1) \cdot \left(1 - (\beta_{j,ub}^n \cdot \alpha_{2,j}^n \cdot g^n)\right), \quad (3.31b)$$

$$\beta_{j,ub}^{n+1} \geq \beta_{j,ub}^n \cdot \left(1 - \alpha_{2,j}^n \cdot g^n\right), \quad (3.32a)$$

$$\beta_{j,ub}^{n+1} \geq \alpha_{1,i}^n \cdot z_i^n, \quad (3.32b)$$

Chapter 4

UAV-mounted RIS as Passive Relays in IoT Networks for Fresh Information

4.1 Background, Related Works, and Contributions

Numerous emerging smart city applications rely on freshness of sensory data (i.e., status-updates) which is being monitored and generated by a plethora of IoT devices. For instance, smart environmental monitoring, industrial control systems and intelligent transportation systems all require reliability and timeliness in delivering status-update information. Outdated updates may be inconsistent with the current status of the physical process being monitored and controlled, which may lead to erroneous decisions. Despite their great benefits, as discussed in previous chapter, IoT devices have limited capabilities and cannot communicate over longer distances in a reliable manner. As a result, providing a timely and reliable communication service for IoT devices is a challenging task, which may hinder their expected benefits. Undoubtedly, emerging IoT services will strongly benefit from enhancing wireless connectivity, which is considered as an enabler for the evolution of future networks and their services. While many key enabling technologies are considered to unleash the potential of future networks, a revolutionary one (which exploits the radio environment has a new degree of

The work presented in this chapter has been submitted to IEEE journal [57]

freedom) has recently emerged and is under intense investigation.

Reconfigurable Intelligent Surfaces (RISs) leverage the tuning capabilities of their reflective elements to enhance the propagation environment by improving the desired signal at the receiver and mitigating interference. They are energy efficient and expected to greatly enhance the spectral efficiency of wireless networks, particularly when combined with other promising technologies. UAVs are among those technologies that have shown great promise in assisting networks by improving connectivity and coverage. Unlike UAVs, as conventional mobile relaying elements, integrating reconfigurable intelligent surfaces with the UAVs allows several benefits. First, the data transmission from the IoT devices to the Base-Station (BS) through RIS-empowered UAV will experience less intermediate delays to relay the information compared to UAVs acting as mobile active relays. This is because, in a decode and forward half-duplex relaying mode, the transmission is executed over two time-slots. On the other hand, an IRS-integrated UAV, which is denoted as aerial RIS (ARIS), requires only one time-slot since the RIS operates in a full-duplex relaying mode explained in Chapter 3. This may enhance the freshness of information and lead to reducing the AoI. Second, the power consumption due to processing the relayed information at the UAVs can be avoided, which increases the flight endurance of the UAV. In fact, RIS is composed of a large number of passive low-cost elements, each of which is capable of independently tuning the phase-shift of the incident radio waves [58]. For instance, by appropriately configuring the phase-shift with the aid of the RIS controller, the reflected signals can be constructively added, and therefore, enhancing the reliability of IoT networks, accordingly, the AoI is minimized. Therefore, in contrast to UAV as a mobile active relay as explained in Chapter 3, integrating RIS with the UAV does not require any radio-frequency chain circuits for relaying the transmission from the IoT devices to the BS which makes UAV equipped with RIS a cost-effective solution with minimal energy consumption.

Recently, some efforts have been directed towards the integration between the UAVs and the RIS with optimizing and designing the phase-shift of the RIS elements in order to intensify different utilities. However, these solutions may not be necessarily optimal from the perspective of preserving freshness of information. To the best of our knowledge, none of the previous works reported in the literature has addressed the optimization of RIS configuration while considering the freshness of information, which thus motivates this work. In this chapter, we study a wireless network where

IoTDs with limited transmission capabilities sample a stochastic process and the sampled data needs to be processed by a BS. A single UAV equipped with RIS is deployed to act as a passive relay node to forward the sampled data to the BS while considering the different activation patterns of IoTDs. The sampled data is successfully delivered to the BS if and only if the signal-to-noise ratio (SNR) exceeds a predefined threshold, upon which the AoI decreases. This framework is formulated as an optimization problem with the objective of minimizing the expected sum AoI (ESA) while considering the SNR constraints, UAV altitude constraint, and the IoTDs scheduling constraints. Then the optimization problem is shown to be difficult to solve while considering a realistic challenging scenario where the activation patterns of IoTDs are unknown. Therefore, we opt to apply a Deep Reinforcement Learning (DRL) framework based on Proximal Policy Optimization (PPO) to learn randomness of the IoTDs' activation patterns and control the altitude of the UAV, the phase-shift of RIS elements along with communication scheduling to minimize the ESA.

4.2 System Model and Problem Formulation

As illustrated in Fig.4.1, we consider an IoT wireless network where a set \mathcal{M} of M IoTDs with limited capabilities are deployed to provide time-stamped, status-update information. Due to IoTDs' capabilities constraints and environmental obstacles, the existence of a strong direct Line-of-Sight (LoS) communication link is difficult to obtain. Therefore, a single UAV equipped with RIS consisting of F reflecting elements is deployed to passively relay the status-update information to the BS. We consider the system over multiple time frames. Each of these frames is further divided into equal segments, that is, N time-slot of length δ_t , which is normalized to unity.

The planar coordinates of the deployed UAV are assumed to be placed at (x_U, y_U) . In Cartesian coordinates, the locations of IoTDs are assumed to be known and located at $(x_i, y_i, 0), \forall i \in M$ at ground level. Depending on the services/applications, the activation patterns for IoTDs are different¹. In addition, we assume that the BS is located at (x_s, y_s, H_S) , where H_S denotes the height of the BS. At any given time-slot n , the deployed UAV can adapt its altitude $H_U[n]$ such that $H_U[n] \in (H_{min}, H_{max})$, where H_{min} and H_{max} are the minimum and the maximum altitude range

¹As mentioned in the 3rd generation partnership project (3GPP).

specified by aviation authorities, respectively. Consequently, the altitude control should meet the following constraints:

$$H_{min} \leq H_U[n] \leq H_{max}, \forall n, \quad (4.1)$$

$$\left| H_U[n+1] - H_U[n] \right| \leq D_{max}, n = 1, \dots, N-1, \quad (4.2)$$

$$H_U[1] = H_S, \quad (4.3)$$

where $D_{max} = V_{max} \delta_t$ is the maximum vertical distance by the UAV in one time-slot based on its maximum speed V_{max} and H_S denotes initial vertical location.

The BS continuously controls the altitude of the UAV as well as the phase-shift of the reflecting elements in order to serve the IoTDS and maintain their required quality of service (QoS). Let $\Phi[n] = \text{diag}\{e^{j\phi_1[n]}, e^{j\phi_2[n]}, \dots, e^{j\phi_f[n]}, \dots, e^{j\phi_F[n]}\} \in \mathbb{C}^{F \times F}$ be the RIS's diagonal phase-shift matrix in the n th time-slot, where $\phi_f[n] \in [0, 2\pi)$, $\forall f \in F$ is the phase-shift for the f th reflecting element. With time division multiple access, the UAV schedules at most one IoTDS to transmit its status-update. Therefore, the transmission scheduling should meet the constraint below:

$$\sum_{i=1}^M \alpha_i[n] \leq 1, \quad \forall n, \quad (4.4)$$

where $\alpha_i[n]$ is a binary variable, which indicates that IoTDS i is scheduled in time-slot n , and 0 otherwise.

Before we proceed, we define the distance model and the adopted channel gain model. We denote $d_{i \rightarrow U}[n]$ and $d_{U \rightarrow S}[n]$ as the distance between the IoTDSs and the UAV in the n th time-slot and between the UAV and the BS, respectively, which are given as follows.

$$d_{i \rightarrow U}[n] = \sqrt{(x_i - x_U)^2 + (y_i - y_U)^2 + (H_U[n])^2}, \quad (4.5)$$

and

$$d_{U \rightarrow S}[n] = \sqrt{(x_S - x_U)^2 + (y_S - y_U)^2 + (H_S - H_U[n])^2}. \quad (4.6)$$

Meanwhile, the channel gain between the IoTDSs and the UAV, and between the UAV and the BS are denoted as $h_{i \rightarrow U}[n]$ and $h_{U \rightarrow S}[n]$ and can be expressed as follows,

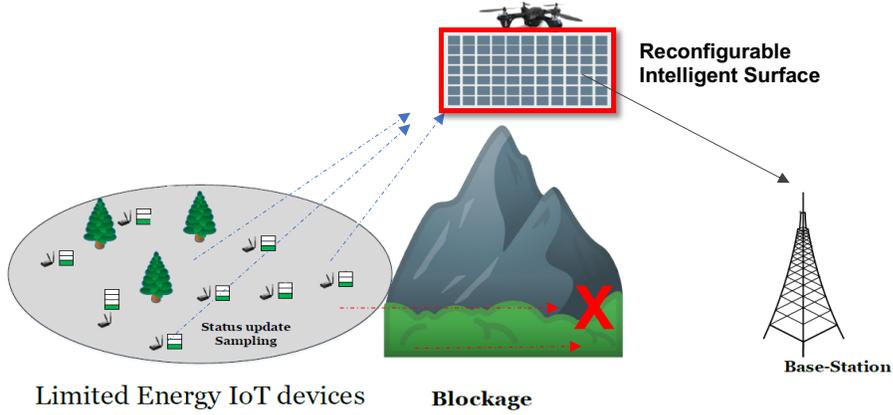


Fig. 4.1: ARIS-assisted IoT wireless networks

$$h_{i \rightarrow U}[n] = \widehat{h}_{i \rightarrow U}[n] \Delta_{i \rightarrow U}[n], \quad (4.7)$$

and

$$h_{U \rightarrow S}[n] = \widehat{h}_{U \rightarrow S}[n] \Delta_{U \rightarrow S}[n], \quad (4.8)$$

where $\widehat{h}_{i \rightarrow U}[n]$, $\widehat{h}_{U \rightarrow S}[n]$, $\Delta_{i \rightarrow U}[n]$ and $\Delta_{U \rightarrow S}[n]$ represent the small-scale fading between the IoTDs and the UAV, the small-scale fading between the UAV and the BS, the path-loss coefficients between the IoTDs and the UAV, the path-loss coefficients between the UAV and the BS, respectively. In particular, the path-loss coefficients $\Delta_{i \rightarrow U}[n]$ and $\Delta_{U \rightarrow S}[n]$ can be written as $\Delta_{i \rightarrow U}[n] = \sqrt{\gamma_0 d_{i \rightarrow U}^{-\eta}[n]}$ and $\Delta_{U \rightarrow S}[n] = \sqrt{\gamma_0 d_{U \rightarrow S}^{-\eta}[n]}$, where γ_0 is the path-loss average channel power gain at a reference distance $d_0 = 1m$, η is the path-loss exponent. Similar to [59, 60], we consider a Rician fading with a dominant LoS. Thus, the small-scale fading for the communication link between the IoTD and the UAV $\widehat{h}_{i \rightarrow U}[n]$ can be given as follows

$$\widehat{h}_{i \rightarrow U}[n] = \sqrt{\frac{K_1}{K_1 + 1}} \bar{h}_{i \rightarrow U}[n], \quad (4.9)$$

where K_1 is the Rician factor. $\bar{h}_{i \rightarrow U}[n] = [e^{j\psi_{i,1}}, e^{j\psi_{i,2}}, \dots, e^{j\psi_{i,F}}]$ is a fixed component vector with elements of unit power, and $\psi_{i,f} \in [0, 2\pi]$. Similarly, the small-scale fading for the communication link between the UAV and the BS $\widehat{h}_{U \rightarrow S}[n]$ can be given by

$$\widehat{h}_{U \rightarrow S}[n] = \sqrt{\frac{K_2}{K_2 + 1}} \bar{h}_{U \rightarrow S}[n], \quad (4.10)$$

where K_2 is the Rician factor. $\bar{\mathbf{h}}_{U \rightarrow S}[n] = [e^{j\omega_{i,1}}, e^{j\omega_{i,2}}, \dots, e^{j\omega_{i,F}}]$ is a fixed component vector with elements of unit power, and $\omega_{i,f} \in [0, 2\pi]$. All IoTDs are assumed to have the same transmit power denoted by P .

Based on the defined channel model in (4.5) - (4.10), the SNR at the BS in time-slot n can be expressed as

$$\Upsilon_i(\Phi[n], H_U[n]) = \frac{P \left| \mathbf{h}_{i \rightarrow U}^H[n] \Phi[n] \mathbf{h}_{U \rightarrow S}[n] \right|^2}{\sigma^2}, \quad (4.11)$$

where σ^2 is the thermal noise power. Note that, the overall channel gain between an IoTD and the BS, i.e., $\mathbf{h}_{i \rightarrow U}^H[n] \Phi[n] \mathbf{h}_{U \rightarrow S}[n]$, can be written as

$$\mathbf{h}_{i \rightarrow U}^H[n] \Phi[n] \mathbf{h}_{U \rightarrow S}[n] = \frac{\gamma_0 \sum_{f=1}^F |[h_{U \rightarrow S}]_f| |[h_{i \rightarrow U}]_f| \cdot e^{j(\phi_f[n] - \psi_{i,f} - \omega_{i,f})}}{d_{U \rightarrow S}^{-\eta/2}[n] d_{i \rightarrow U}^{-\eta/2}[n]}, \quad (4.12)$$

where $[h]_f$ is the f th element of h and γ_0 is the path-loss average channel power gain at a reference equal to 1m. The maximum SNR at the BS can be achieved when the phase-shift is chosen as $\phi_f[n] = \psi_{i,f} + \omega_{i,f}$ [58].

In order to achieve a successful transmission, $\Upsilon_i(\Phi[n], H_U[n])$ should be strictly greater than or equal to Υ_{th} , where Υ_{th} is the minimum threshold to ensure reliable decoding [61]. A single packet queuing discipline is assumed to be employed by the IoTDs such that the older status-update packet is dropped and replaced with the newly arrived sample. A *per time-slot sampling* policy is considered for sampling the information, where the scheduled activated IoTD samples the status-update information at the beginning of each time-slot to transmit its status-update information. Therefore, the deployed UAV has to control the scheduling, altitude and phases of RIS elements properly to relay the status-update information to the BS while considering the activation patterns of IoTDs. Clearly, AoI depends on the altitude of the UAV, the communication scheduling, phase-shift of the RIS elements and the activation pattern of the IoTDs. Thus, the evolution of $A_i[n]$ of IoTD i can be

written²

$$A_i[n+1] = \begin{cases} 1, & \text{if } G_i[n] = 1, \alpha_i[n] = 1, \text{ and } \Upsilon_i(\Phi[n], H_U[n]) \geq \Upsilon_{th}, \\ A_i[n] + 1, & \text{otherwise,} \end{cases} \quad (4.13)$$

where $G_i[n]$ is a binary variable, which indicates that IoT i is active in time-slot n , and 0 otherwise. To obtain the AoI within the relay mission time, we use the the ESA $\frac{1}{NM} \mathbb{E} \left[\sum_{n=1}^N \sum_{i=1}^M A_i[n] | A_i[0] = 0 \right]$ as our metric to evaluate the freshness of sampled data. For the sake of tractability, the AoI can be expressed as the constraints below

$$A_i[n+1] = 1 + A_i[n] - A_i[n]G_i[n]\alpha_i[n], \quad (4.14)$$

$$\Upsilon_i(\Phi[n], H_U[n]) \geq G_i[n]\alpha_i[n]\Upsilon_{th}. \quad (4.15)$$

With the quest of enhancing the performance of the IoTs, a framework to optimize communication scheduling, phase-shift matrix of RIS, and the altitude of the UAV is investigated. This framework is formulated as an optimization problem with the objective of minimizing the ESA. For ease of notation, let us denote $\mathbf{L} = \{H_U[n], \forall n\}$, $\mathbf{S} = \{\alpha_i[n], \forall i, n\}$ and $\Theta = \{\Phi[n], \forall n\}$. Thus, our problem can be written as:

$$(\mathcal{O} \mathcal{P}): \min_{\mathbf{L}, \mathbf{S}, \Theta} \frac{1}{NM} \mathbb{E} \left[\sum_{n=1}^N \sum_{i=1}^M A_i[n] | A_i[0] = 0 \right] \quad (4.16a)$$

$$\text{s.t. } (4.1) - (4.4), (4.14), (4.15),$$

$$\alpha_i[n] \in \{0, 1\}, \quad \forall i, n, \quad (4.16b)$$

$$\phi_f[n] \in [0, 2\pi), \quad \forall f, n, \quad (4.16c)$$

Owing to the randomness of the activation pattern, $G_i[n]$, of the IoTs, $(\mathcal{O} \mathcal{P})$ is a stochastic optimization problem over the service time N . In fact, obtaining the activation patterns of IoTs are crucial before dispatching the UAV to a target area. This is because the formulated problem aims to find the control policy that minimizes the AoI from the active IoTs within the service time N .

²For more tractable analysis, the initial values of AoI is neglected, that is, $A_i[0] = 0, \forall i$.

However, obtaining complete information on the activation pattern requires extensive measurement, which is not easy to be obtained especially in remote areas. We also observe that $(\mathcal{O} \mathcal{P})$ is a mixed-integer non-convex optimization problem which is hard to be solved. This is because $(\mathcal{O} \mathcal{P})$ contains both binary variables $\alpha_i[n]$ and continuous variables $\Phi[n]$ and $H_U[n]$. In addition, it is a challenging task to solve a non-convex optimization problem in the absence of a complete information on the activation pattern. Therefore, our problem is reformulated as MDP and a model-free DRL based on PPO is exploited to find the effective control policy that minimizes the ESA. The proposed PPO algorithm does not rely on a prior knowledge of the activation patterns.

4.3 Proposed Solution

Finding the control policy that governs the UAV's altitude, the scheduling policy and the phase-shift matrix for the RIS is a non-trivial challenge. The reason is that the considered work is a hybrid *discrete-continuous* action space problem and the altitude and scheduling are also closely coupled and should be carefully considered with the adjustments of RIS's phase-shift. Discretizing the altitude of the UAV and the phases of RIS elements into discrete actions could be one way to tackle this challenge. However, combining the three actions (UAV's altitude, scheduling policy and phases of RIS) into one single action space is still a challenge that needs to be addressed. The reason behind that, for a deployment with M IoTDs, \mathcal{K} discrete altitude actions and Q discrete phases, a total of $(\mathcal{K} \times M \times F \times Q)$ possible actions need to be considered. In fact, efficient DRL algorithms are difficult or even often impossible to apply to solve large discrete action spaces since it increases the difficulty of learning. Since only one IoTD is scheduled per time slot, the phase-shift matrix for the RIS is properly configured so that the reflected signals can be constructively added at that IoTD. Due to that and to reduce the complexity of the learning and maintain a small size of the action space, we use the UAV altitude and scheduling policy as the main control objective. In other words, at each time-slot, the RL-agent (deployed at the BS) will adjust the altitude of the UAV and schedule a transmission along with an appropriate adjustment for the RIS's phase-shift according to the user of interest. Thus, the RL-agent executes two actions simultaneously. Specifically, the deployed UAV adjusts its altitude and at the same time, schedules an IoTD depending on the phases

of RIS elements and the activation pattern.

4.3.1 MDP Formulation

The considered problem is first modeled as a Markov Decision Process (MDP). Afterward, a DRL based on PPO algorithm is proposed for finding the control policy that governs both the UAV's altitude and the scheduling decision within unknown activation pattern. We define the state \mathcal{S} , action \mathcal{A} , and reward \mathcal{R} functions as follows:

1) **State** \mathcal{S} in the PPO model: The state at time-slot n is defined as $s[n] = (A[n], \Upsilon[n], H_U[n])$, where $A[n] = (A_1[n], \dots, A_i[n], \dots, A_M[n])$ and $\Upsilon[n] = (\Upsilon_1[n], \dots, \Upsilon_i[n], \dots, \Upsilon_M[n])$ is a vector of size M containing the SNR at time-slot n when the signals from different paths are coherently combined through the phases of RIS elements.

2) **Action** \mathcal{A} in the PPO model: At each step-slot n , the RL-agent executes an action $a[n]$ denoted by $a[n] = (\xi[n], \kappa[n])$, where $\xi[n] \in \mathcal{W} = [\alpha_1[n], \dots, \alpha_i[n], \dots, \alpha_M[n]]$. $\kappa[n] \in \mathcal{K} = \{(1, 0, 0), (0, 1, 0), (0, 0, 1)\}$, where $\kappa[n]$ represents the variable quantity of the altitude distance. $\kappa[n] = (1, 0, 0)$ means the UAV adjusted its altitude upward at time-slot n ; $\kappa[n] = (0, 1, 0)$ means the UAV adjusted its altitude downward at time-slot n ; and $\kappa[n] = (0, 0, 1)$ means the UAV is hovering. Hence, the system action space at time-slot n is $a[n] = (\xi[n], \kappa[n]) \in \mathcal{A} = \mathcal{W} \times \mathcal{K}$. Thus, the RL-agent adjusts the UAV's altitude and decides which IoT-D to transmit its status-update to the BS.

3) **Reward** \mathcal{R} in the PPO algorithm: The immediate reward $r[n]$ is defined as a negative summation of AoI. Therefore, the RL-agent is motivated to minimize the AoI by optimizing scheduling decisions and altitude control of the UAV.

Generally, MDP problems with predefined state transition probabilities can be solved using Dynamic Programming (DP). However, since the UAV is dispatched with no prior knowledge on the activation patterns of IoT-Ds, then DP algorithm cannot be applied. Thus, to solve the considered problem a DRL based on PPO algorithm³ is employed in the next subsection.

³It is noteworthy that, PPO demonstrates performance comparable to or better than state-of-the-art DRL approaches. Therefore, PPO has become the default reinforcement learning algorithm at OpenAI.

Algorithm 5: The PPO Based Framework.

```
1 Initialize  $\theta$  randomly,  $\pi_{\theta_{old}}$  with  $\theta_{old} \leftarrow \theta$  and  $H_U[1] = H_S$ .
2 for  $Iteration = 0, 1, \dots$  do
3   for  $l = 0, 1, \dots, L$  do
4     Get  $(A[n], Y[n], H_U[n])$  from the environment.
5     Sample action  $a[l] \sim \pi_{\theta_{old}}$ .
6     Take the action  $a[l]$  that specifies the UAV's altitude and the IoTD scheduling.
7     Configure  $\Phi[n]$  that maximizes the received SNR of the scheduled IoTD at the BS.
8     if UAV violates the allowable altitude range then
9       Add a penalty, cancel the movement of UAV and update  $s[l+1]$ .
10    Get relevant reward  $r[l]$  and  $s[l+1]$ .
11    Store  $(s[l], a[l], r[l], s[l+1])$  as one transition in the experience replay.
12    Compute advantage estimate.
13  for  $epoch = 0, 1, \dots$  do
14    Use Eq. (4.17) to compute the PPO objective.
15    Optimize the overall objective function and update the policy via stochastic
    gradient ascent with ADAM.
16  Synchronize the sampling policy with  $\theta_{old} \leftarrow \theta$ .
17  Clear up the stored data.
```

4.3.2 Proposed Solution Description

At the initialization stage (Line 1), the proposed algorithm randomly initializes the initial altitude of the UAV at H_S to guarantee Eq. (4.3). Besides, the deep neural networks (NNs) parameter θ are randomly initialized, where Deep NNs have the same structure. In each training iteration, the PPO algorithm alternates between the sampling phase by running L episodes (Lines 3-12) and the optimization/exploration phase (Lines 13-15). During each episode l , the current channel state information $\mathbf{h}_{i \rightarrow U}^H[n]$ and $\mathbf{h}_{U \rightarrow S}[n]$ are obtained. Then all possible phases of RIS elements $\Phi[n]$ that achieve a coherent combination with the signals from different paths at the BS are obtained. The RL-agent (Line 4) then get observations $A[n]$, $Y[n]$ and $H_U[n]$ from environment at each time-slot. The UAV then executes action $a[l]$ from the policy $\pi_{\theta_{old}}$ (Lines 6-7), where the sampled action, $a[l]$, represents the current altitude of the UAV and the scheduling policy according to the phases of RIS configuration for each IoTD that achieve the maximum SNR. Eq. (4.16c), Eq. (4.14) and Eq. (4.15) are implicitly defined in Lines 4-7, where the states of the MDP and actions are defined. In this step, the PPO algorithm assigns the binary value "1" to the scheduled IoTD to transmit its status-update

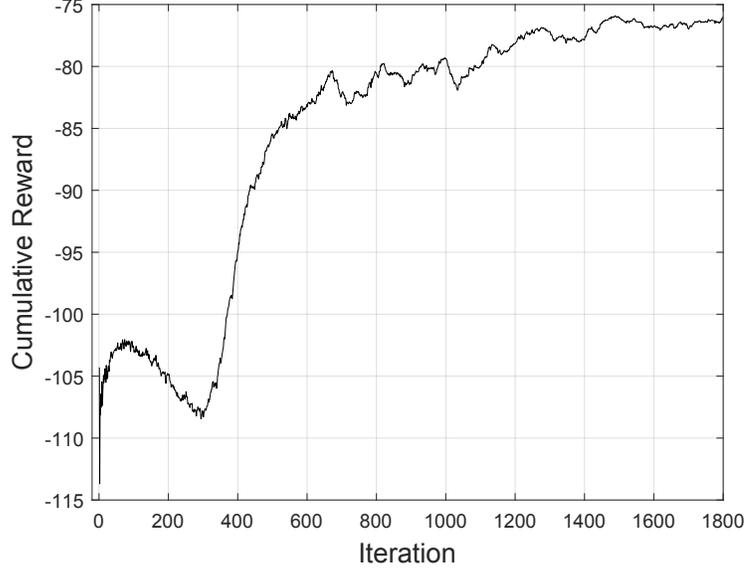


Fig. 4.2: Convergence.

and assign "0" for other IoTDs to guarantee Eq. (4.16b) and Eq. (4.4) .

During each episode (Lines 8-9), the RL-agent guides the UAV to avoid the action that violates the altitude constraint (i.e., Eq. (4.1)) by abandoning the corresponding altitude action and apply a penalty to the reward. After taking the current action, the UAV receives the relevant reward (Line 10), which represents the sum of the AoI for all IoTDs. After, the tuple of RL trajectory data of episode l , $(s[l], a[l], r[l], s[l+1])$ are buffered for the next phase (Line 11), then advantage estimate is computed (Line 12) to achieve efficient training of the policy. To update policy network π_θ , the PPO clip objective function in each epoch is computed according to the below

$$L^{CLIP}(\theta) = \mathbb{E}_n \left[\min \left(\frac{\pi_\theta(a[n]|s[n])}{\pi_{\theta_{old}}(a[n]|s[n])} \mathfrak{A}_{\pi_{\theta_{old}}}(s[n], a[n]), \right. \right. \quad (4.17)$$

$$\left. \left. clip \left(\frac{\pi_\theta(a[n]|s[n])}{\pi_{\theta_{old}}(a[n]|s[n])}, 1 + \varepsilon, 1 - \varepsilon \right) \mathfrak{A}_{\pi_{\theta_{old}}}(s[n], a[n]) \right) \right],$$

where ε is the clip fraction used to control the clip range. $\mathfrak{A}(s[n], a[n])$ is the advantage estimate in time-slot n that is used to mitigate the high variance of the gradient. Then the PPO overall objective function is then optimized via stochastic gradient ascent (SGA) with ADAM. Finally, the policy $\pi_{\theta_{old}}$ is updated with the policy π_θ and the buffered data are dropped (Line 17) then a new iteration begins.

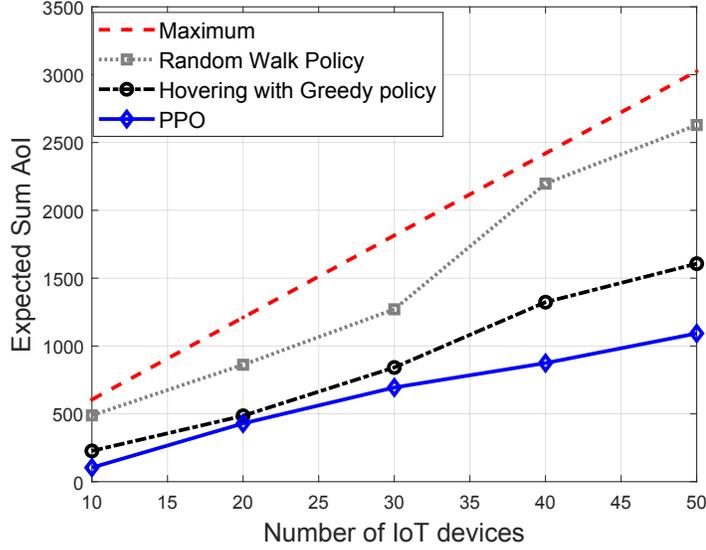


Fig. 4.3: Impact of number of IoTDs.

The complexity of the proposed algorithm can be expressed as the number of multiplications: $O(\sum_{p=1}^{P-1} n_p \cdot n_{p-1})$, where n_p is the number of neural units in the p -th hidden layer. The convergence of the algorithm is limited to simulations since it is hard to be analytically analyzed.

4.4 Simulation Results and Discussion

We consider a square area of $0.5\text{km} \times 0.5\text{km}$, where IoTDs are distributed randomly. An UAV equipped with RIS is placed at the center of the area to relay the status-update information from IoTDs to the BS located at $(2000, 500, 25)\text{m}$. We assume that all the IoTDs have the same power budget. The frame duration N is taken as $N = 120$ while the communication parameters are taken as: $P = 20\text{dBm}$, the path-loss exponent $\eta = 2.3$, the channel gain $\gamma_0 = -20\text{dBm}$, $\sigma^2 = -110\text{dBm}$, $\Upsilon_{th} = 0\text{dB}$ [61] and $K_1 = K_2 = 8\text{dB}$. The UAV altitude parameters are assumed as $H_S = 100\text{m}$, $H_{min} = 10\text{m}$, $H_{max} = 1000\text{m}$ and $D_{max} = 10\text{m/s}$. The activation pattern of each IoTD is randomly generated according to Uniform distribution⁴. Table 4.1 provides PPO hyperparameters. The results are collected by utilizing PyTorch library after 240K samples, where each sample corresponds to a snapshot of the network at a particular time-slot.

⁴The same solution approach can be applied to any activation distribution.

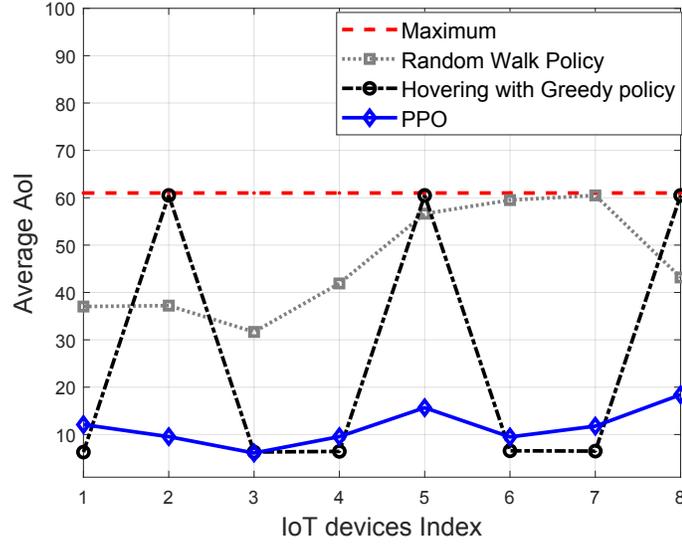


Fig. 4.4: Average age per IoTD.

We first verify the convergence of the proposed PPO algorithm in Fig. 4.2. It can be found that trained RL-agent can significantly improve the defined reward. This improvement begins to diminish when the RL-agent is well trained about the activation patterns of IoTDs and it starts to effectively adapt the UAVs' altitude and the scheduling policy that minimizes the ESA. It is observed that our PPO algorithm converges to the steady point quite quickly under a reasonable choice of the hyperparameters. This indicates that our proposed algorithm deals effectively with incomplete knowledge of the activation pattern of the IoTDs.

To evaluate the effectiveness of the proposed algorithm, we develop two baselines policies as follows: 1) a random walk policy which randomly selects an IoTD to relay its status-update information along with adjusting the phases of RIS so that the reflected signals can be constructively added at the selected IoTD while changing the altitude of UAV randomly, and 2) hovering with the greedy policy where the UAV iteratively searches for the best height that satisfies the reliability constraint for the most IoTDs. Then, the UAV selects the IoTD with the maximum current AoI. Similar to the random walk policy, the phase-shift matrix of the RIS is adjusted with the same way. The baseline policies are adequate policies since the former policy explores all possible actions, thus, may obtain some actions that result in decreasing the AoI, while the latter policy is heuristically a good policy since it always selects the IoTDs with higher AoI to relay their status.

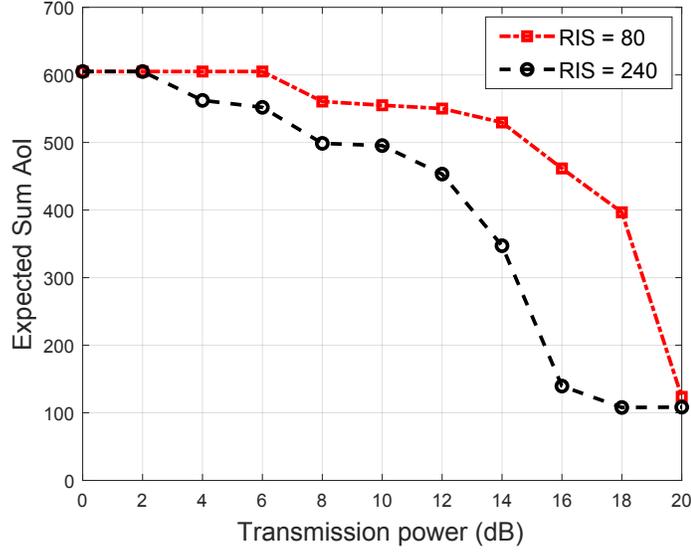


Fig. 4.5: Impact of number of RIS elements.

Fig. 4.3 illustrates the impact of the number of IoTDs on the PPO algorithm compared to the baselines policies. It can be noted that the proposed algorithm is able to minimize the ESA for a lower number of IoTDs since each IoTD enjoys more frequent scheduling. However, for a large number of IoTDs we can see that the ESA increases since more scheduling is needed to decrease the ESA. Besides, the hovering with the greedy policy is more effective than the random walk policy since it always selects the IoTD with the highest AoI value. We also can observe that the proposed algorithm outperforms all the baselines. This is expected since the baselines are unable to learn the activation pattern of the IoTDs and the altitude adaptation of the UAV is not considered in the baselines.

We plot the average age as another performance metric for a set of IoTDs in Fig. 4.4. The average age of IoTD i within time N is calculated by $\frac{1}{N} \sum_{n=0}^N A_i^n, \forall i$. Obviously, it can be seen that the proposed PPO has a lower average sum AoI per IoTD compared to the baseline policies. The average age gap among the policies is relatively high, which demonstrates the importance of learning of the activation pattern of IoTD and adjust the altitude of the UAV with communication scheduling. Furthermore, the hovering with the greedy policy, on the one hand, significantly decreases the AoI for some IoTDs. On the other hand, it increases the AoI to the maximum for other IoTDs. This is because the hovering with the greedy policy only schedules transmission for IoTDs that satisfy

Table 4.1: Simulation Parameters in UAV-RIS assisted IoT Networks

Parameter	Value
Activation Functions	Softmax and Tanh
Learning Rate	0.001
Reward Discount	0.9
Number of Hidden Layers for Networks	3
Number of Neurons	64
Loss Coefficients K_1 and K_2	0.5 and 0.01
Update Policy Length, L	240
Clip Fraction, ϵ	0.2
Optimizer Technique	Adam

the SNR threshold. This finding justifies the robustness of the proposed PPO algorithm in terms of minimizing average AoI.

Finally, in Fig. 4.5, we show that both the power budget of the IoTDs and the number of RIS's elements have a great impact on the ESA. Increasing the transmit power of IoTD leads to a direct enhancement of the achieved SNR at the BS. However, this may not be allowable in certain IoT applications. As a result, it is better to increase the number of reflecting elements per RIS which results in enhancing the quality of the communication link between the IoTD and the BS, which in turns improves the achieved SNR and the ESA.

4.5 Summary

This chapter proposed a new relaying system to maintain the freshness of information of remote Internet of Things wireless network by integrating the Unmanned Aerial Vehicle (UAV) and the reconfigurable intelligent surfaces (RIS). The altitude of the UAV, the transmission scheduling, and phase-shift matrix of RIS elements are optimized to minimize the expected sum *Age-of-Information*. To tackle this mixed-integer non-convex problem, Proximal Policy Optimization algorithm is proposed. Numerical results demonstrate that the proposed algorithm can significantly minimize the AoI compared to other baselines such as random walk and heuristic greedy algorithms.

Chapter 5

Learning-based Trajectory Planning of Aerial Base Stations in Vehicular Networks

5.1 Background, Related Works, and Contributions

The Internet-of-Vehicles (IoV) paradigm is expected to play a crucial role in the automotive industry by creating innovative services and revenue sources. By 2025, around a quarter of a billion vehicles navigating along global roads will be equipped with wireless communication capabilities [65]. It is, therefore, indispensable to study the use of UAVs in vehicular networks. Thus, after addressing three key problems in the static environment, the deployment of UAVs in highly dynamic environments such as vehicular networks is investigated.

One of the main applications of future wireless networks is to provide ubiquitous coverage to suburban, rural highways and volatile environments, where vehicles might need access to different types of information including safety commands, maps, and route guidance during the entire navigation period. However, the seamless provision of connectivity and the uninterrupted delivery of

The work done in this chapter leads to two IEEE published journals and one conference [62–64]

services along highways also pose various challenges that future wireless technologies promise to solve. The situation is particularly challenging in highways such as cross-border highways where the communication services might be unavailable. Nevertheless, cellular infrastructure could either be inadequately provisioned to cover all vehicles or could be exposed to unexpected hardware failure or direct damage. Besides, providing ubiquitous coverage to highways, where terrestrial infrastructure is economically infeasible due to geographical constraints, is a challenging task that current and future wireless networks have to consider.

Alternatively, intelligent cell-free networks, where cellular boundaries may no longer exist, with the provision of some isolated AI operations [66] are expected to play an important role in future wireless networks, where the full potential of mobile base-stations such as unmanned aerial vehicles (UAVs) or dronecells will be realized to provide effective coverage whenever needed. Indeed, due to their agility and mobility, UAVs are being promoted as a promising paradigm in future wireless networks to provide network connectivity when the infrastructure is partially or fully unavailable. Furthermore, they can be deployed to enhance the coverage of cellular networks during an unplanned surge in traffic demand [67]. An additional advantage of using UAVs is that they can be relocated from one zone to another in order to provide network connectivity based on the actual traffic demand. Moreover, in vehicular networks that are typically characterized by high mobility and varying vehicle arrival pattern, multiple UAVs with autonomous control are required to cooperatively provide network coverage and adapt to current traffic conditions. Thus, the existence of a swarm of UAVs to provide wireless connectivity will be necessary.

Nonetheless, UAVs have limited communication ranges and are constrained by their energy budget. Thus, they cannot serve on entire highway all the time or keep flying back and forth for long periods. It is thus challenging to find the trajectories of a minimum number of UAVs in order to achieve effective coverage in the long run under UAV's energy budget constraint, while maintaining a certain Quality of service (QoS). To this end, we propose to leverage AI technique particularly Deep Reinforcement Learning (DRL) in order to control the trajectories of UAVs and present a novel and highly efficient solution.

Related Works

Much research has been recently done to address various challenges in UAV communication systems. Optimizing the trajectory of the UAV is one of the important research challenges. For instance, the work in [7] maximized the minimum rate among ground users by optimizing the trajectory and user scheduling for a single-UAV. In [11], the authors optimized the UAV's trajectory to minimize the time to completely disseminate a common file to a number of distributed ground terminals. In [13], the UAV trajectory and ground terminal transmit power are jointly optimized for both circular and straight trajectories to reveal a fundamental trade-off between the UAV propulsion energy consumption and ground terminal communication energy consumption. In [68] the authors analyzed the coverage properties and proposed a UAV-based deployment in the emergency scenario. The authors in [69] analyzed the optimal height for the UAV to minimize the transmitted power for covering a target area. In [70] the horizontal positions are optimized while fixing the altitude of the UAVs to minimize the number of UAVs required to cover a fixed number of stationary users. The authors in [71] studied a similar problem for a drone-cell placement optimization in three-dimensional space.

Recently, few works have been conducted to investigate the use of multiple UAVs. In fact, compared to a single UAV, the use of a swarm of UAVs allows to operate in challenging missions with higher performance and efficiency. However, new issues should be considered with using the swarm of UAVs such as energy efficiency, path planning, etc. For example, the work in [10] jointly optimized the trajectory, multi-user scheduling and power control for multiple UAVs to maximize the minimum rate of ground users. In our recent work in [72], we optimized the trajectories and radio resources of the minimum number of UAVs to serve vehicles in a mobility environment. In [19], the authors deployed multiple UAVs for collecting data from ground IoT devices, where the total uplink transmit power of these IoT devices is minimized in a time-varying network by optimizing the UAV's trajectory and IoT power control.

Machine learning has received significant attention and particularly has been recently utilized for solving challenging problems with UAVs. Specifically, the authors of [73] employed echo state network based prediction algorithm for predicting the future locations of ground users and then a

multi-agent Q-learning based algorithm is proposed to predict the locations of UAVs in each time-slot. In [74], a centralized Deep Reinforcement Learning is proposed to control the trajectory of UAVs in a static environment for providing effective communication coverage while considering fairness and energy consumption for a fixed number of UAVs. In [75] the authors proposed a decentralized Deep Reinforcement Learning solution to obtain the trajectories of multiple UAVs to achieve the energy efficiency.

Unlike the works in [7, 11, 13, 68–71] that consider a single UAV, multiple UAVs are employed and coverage services is maximized under UAVs' energy budget constraint. In contrast to the works in [19, 72–75] which study the performance of multiple UAVs in static environment, these results cannot readily extend to cases in a highly dynamic environment such as vehicular networks where the network's topology frequently changes. To this end, we consider a vehicular network in our analysis and a machine learning approach is exploited to learn the vehicular environment and its dynamics to handle the complex continuous action space. In other words, we are interested in further leveraging the Deep Reinforcement Learning technique for UAV control and present a Deep Reinforcement Learning based method to offer network coverage in dynamic environment.

Motivation and Work Objectives

Despite several studies related to the deployment and trajectory optimization of UAVs, there are still many open questions that are yet to be answered. In particular, for vehicular networks, there is no framework that can provide the minimum number of UAVs to serve vehicles on a given highway segment in a high mobility scenario under UAV's energy budget constraint while maintaining an acceptable Quality of Service (QoS) for each vehicle. Most of the existing coverage work relies on users which are stationary where a complete knowledge about the environment (such as the users' instantaneous location) is available in order to obtain results. However, users could be mobile (e.g., vehicles) with random speeds, hence, the assumptions of a global knowledge of the network is not valid, especially in highly dynamic environments such as vehicular networks. Nonetheless, prior work relies on optimization frameworks which require high computation resources to attain results [76]. Furthermore, none of the existing literature provides a solution for a real scenario in

highly dynamic environments such as a vehicular network, where the network's topology is frequently changed.

In our work in [72], we provided a mathematical optimization framework that can provide the minimum number of UAVs and their optimal trajectories to serve vehicles on a given highway segment in a high-mobility scenario. However, we considered a complete knowledge about the environment in advance in order to obtain our results. The assumption of knowing the instantaneous location of vehicles in advance is not valid in a realistic scenario. Furthermore, the energy consumption of the UAVs has been neglected. However, the energy consumption needs to be carefully considered if the trajectory requires serving for a long time.

Unlike our study presented in [63], where a Reinforcement Learning approach is utilized to govern the UAVs' trajectories with a set of actions (traveling distance), in this work, we consider UAVs trajectories without traveling restrictions, constraining their mobility to finite distances (i.e. the UAV trajectories is continuous), where the action space dimension is infinite. Hence, the required computational time and effort to realize an optimal number of UAVs and their trajectories using Reinforcement Learning is infeasible. Furthermore, fixing the coverage of a UAV may not be the best solution to minimize the number of UAVs which should be dynamically changeable according to the current traffic condition.

Obviously, we are dealing with a continuous control task since each UAV can carry out infinite actions (traveling distance) to serve the existing and newly arriving vehicles, and hence the use of Deep Reinforcement Learning (DRL) techniques is necessary to explore the effect of the UAVs' actions on the vehicular environment. It is important to mention that the control task is not dependent only on one UAV but on the joint actions of all UAVs. Deep Q-Network (DQN) could be adapted to solve continuous control task through discretizing the action space. However, one of the major limitation of this technique is the curse of dimensionality [77]: the number of actions space increases exponentially with the number of degrees of freedom. For instance, a 20 degree of freedom action (as traveling distance in both direction) for 2 UAVs leads to an action space with dimensionality: $2^{20} = 1048576$. The situation is worse with increasing the number of UAVs and their action space. The commonly-used method for continuous control task is the Actor-Critic algorithm (AC) which uses neural network approximator to learn policies in continuous action spaces. So, we adapt one

of the state-of-the-art method of Actor-Critic, Deep Deterministic Policy Gradient (DDPG) [78], to solve our problem.

To this end, this work proposes the exploitation of the Actor-Critic algorithm, which has been shown to deliver superior performance in continuous action spaces. Taking advantage of the ability of Actor-Critic framework in exploring unknown environments, we design the Actor-Critic framework to find the trajectories for a minimum number of UAVs to provide network connectivity for vehicles under UAV's energy budget constraints. To achieve that, a central agent in the external network is trained to observe the environment and then control the decision of realizing the minimum number of UAVs and their trajectories to provide effective communication coverage while maintaining an acceptable Quality of Service (QoS) for each vehicle. This task is quite challenging because UAVs have limited energy and cannot fly all day. So, the UAVs should fly in an energy-efficient manner during the coverage process and back to the charging station when needed. Hence, a UAV (or more) are dispatched to provide coverage for vehicles. Furthermore, in highly dynamic environments such as a vehicular network, the network's topology frequently changes. The trajectories of UAVs should be adapted to account for the aforementioned changes in network topologies.

It is clear that, at most, one needs to deploy a total of $\left\lceil \frac{d}{R} \right\rceil$ UAVs in order to cover the segment, where d and R are the total length of the considered highway and the coverage range of each UAV, respectively. However, given the agility of the UAVs and the dynamic nature of the vehicular network, a fewer UAVs may only be needed to provide the anticipated service based on the vehicles' requirements. To this end, the aim of this work is to find a control policy that specifies the trajectories of a minimum number of UAVs at each time-slot to achieve an effective coverage on the highway while maintaining an acceptable Quality of Service for each vehicle.

Organization

The rest of the work is organized as follows. Section 5.2 presents the communication scenario of the vehicle-to-UAVs. In Section 5.3, the problem formulation of trajectory design to minimize the number of UAVs is presented. Section 5.4 lays out a detailed presentation of the DRL framework. Simulation results are presented in Section 5.5. Finally, conclusions are drawn in Section 5.6, and future research directions are highlighted.

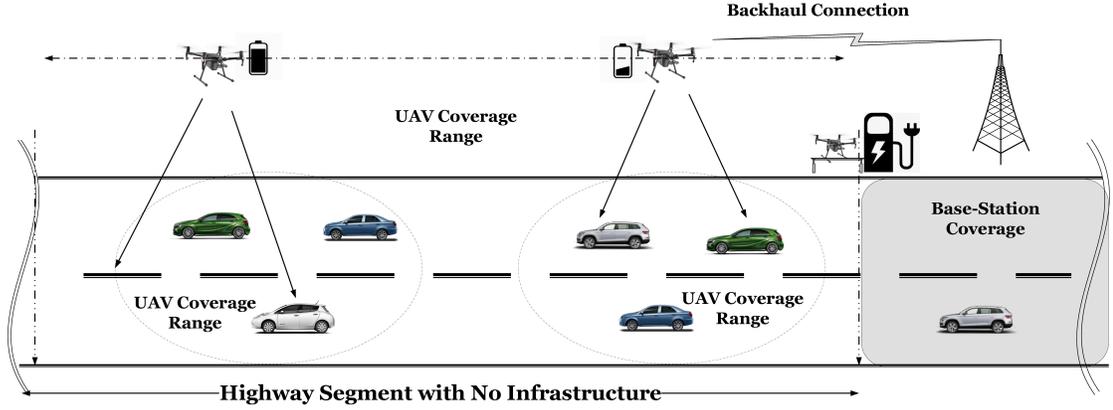


Fig. 5.1: A highway segment with no communication infrastructure where multiple UAVs covering vehicles.

5.2 The Communication Scenario

As illustrated in Fig. 5.1, we consider a highway segment with a communication infrastructure that is either damaged (e.g. following a natural disaster) or non-existent. Furthermore, this segment has unidirectional traffic flow of vehicles that exits the coverage of a fixed Base-Station (BS), where a set \mathcal{M} of M UAVs, indexed by $m = 1, \dots, M$, intended to serve as mobile BSs for vehicles crossing the highway segment. The UAVs are assumed to have high capacity fronthaul links such as free space optics (FSO) or millimeter-wave (mmWave) links with ingress ground BS, where a central unit with Actor-Critic agent resides. The Actor-Critic agent observes the dynamic vehicular environment and steadily learns the optimal trajectory policy and manages the cooperation between the deployed UAVs. Therefore, a vehicle that cannot be covered while being within the coverage of one UAV will be covered by other deployed UAVs. We consider a multiple time frames system where each frame (of duration T) is divided into N equal time-slots, each with length δ_t , indexed by $n = 1, \dots, N$. We use \mathcal{V}^n to denote the subset of vehicles to be served, in time-slot n , where $\mathcal{V} = \mathcal{V}^1 \dots \cup \mathcal{V}^n \dots \cup \mathcal{V}^N$.

In this work, we consider a scenario where the time-frequency resources are sufficient to mitigate the various possible sources of interference. We assume each UAV can simultaneously communicate with multiple vehicles within its coverage by allocating appropriate orthogonal resources to ensure interference-free communication; this interference-free model has been widely used in

the literature [19]. Furthermore, we assume neighboring UAVs are allocated different parts of the spectrum, so that inter-UAV communication is also interference free. Thus, we assume the vehicles are served when they lie within the coverage of any UAV without interference from other UAVs, and henceforth our study is concerned with dealing with the UAVs coverage issue. We adopt a widely used traffic model on the highway [79], [80], where vehicles travel with random speeds; the vehicles' speeds distribution is assumed to be a truncated Gaussian in the range $[v_{min}; v_{max}]$ [81]. We assume that vehicles' speeds are non-constant during their entire navigation period within the given highway segment. Thus, the number of vehicles V within the segment will follow a Poisson distribution with vehicular density ρ_p Veh/Km [82]. According to federal aviation regulations, all UAVs are assumed to fly at a constant altitude H above ground level and the time-varying horizontal coordinate of UAV m at time-slot n is located at $(w_m^n, 0, H)$. During the considered time frame, vehicles enter and leave the highway segment resulting in a change in the number of vehicles in \mathcal{V}^n . We are interested in the arrival and departure times of vehicles causing that change.

In the UAVs-to-Vehicles scenario, four basic assumptions are considered in our analysis as follows:

- Each UAV leaves one station at the beginning of the highway segment and a UAV is rushed to a charging station before its energy is depleted (i.e., before excess a given threshold energy).
- Once a UAV is used, it will continue to be deployed as long as it has sufficient energy above the given threshold energy to serve; whether a deployed UAV may or may not serve vehicles depends on the number of vehicles under its coverage.
- Each vehicle is guaranteed specific QoS (if covered) during its residence on the highway segment.
- A UAV spends its entire energy in flying and hovering. In fact, the energy consumption of a UAV is dominated by the propulsion energy, since the communication energy is minimal compared to propulsion energy. Thus, for more tractable analysis, we neglect communication energy in our work [83]¹.

¹The transmission power is usually relatively small compared to the propulsion energy, e.g., a few watts [84] versus hundreds of watts [85], and thus the transmission power is ignored in this work

In a typical UAV assisted communication, the channel is generally modeled using large-scale fading and small scale fading [86]. However, in highway scenarios, such the one considered in this work, the UAV-to-vehicle channel can be characterized with strong Line-of-Sight and therefore the small scale fading can be neglected [86] [25]. All vehicles are assumed to transmit with constant power P leading to a received power $P_{i,m}^n = h_{i,m}^n P$ in slot n , where $h_{i,m}^n$ is the channel gain from UAV m to vehicle i in time-slot n . This channel gain can be written as:

$$h_{i,m}^n = h_o \left(\sqrt{(x_i^n - w_m^n)^2 + H^2} \right)^{-2}, \forall n, m, i, \quad (5.1)$$

where h_o is the median of the mean path gain at reference distance $d_0 = 1$ m. x_i^n is the instantaneous position of vehicle i in time-slot n . In addition, a total bandwidth B is allocated for each UAV. If v^n vehicles communicate with a UAV at time-slot n simultaneously, the bandwidth each vehicle obtains at time-slot n is calculated by

$$B_i^n = B \varphi(v^n), \quad (5.2)$$

where $\varphi(v^n)$ is the channel utilization function which is a decreasing function of contending vehicle number v^n . Thus, the achievable rate $r_{i,m}^n$ between vehicle i and UAV m at time-slot n can be written as

$$r_{i,m}^n = \begin{cases} B_i^n \log_2 \left(1 + \frac{P h_{i,m}^n}{\sigma^2} \right), & \text{if } a_i \leq n \leq d_i, \\ 0, & \text{otherwise,} \end{cases} \quad (5.3)$$

where a_i and d_i are the arrival and departure times of vehicle i to the highway segment, respectively. where $\sigma^2 = B_i^n N_o$ with N_o denoting the power spectral density of the additive white Gaussian noise (AWGN) at the receivers. In practice, a vehicle i is considered to be covered by a UAV with an acceptable QoS if the achievable rate $r_{i,m}^n$ served by UAV m at time-slot n is greater than a threshold value r_{min} , which indicates an acceptable rate for each vehicle.

While flying to serve vehicles on the highway segment, UAVs determine their trajectories in a way to save on their total consumed energy consumption. We follow the energy consumption model for a UAV presented in [85], where the total power consumption for constant speed UAV ω can be

modeled as

$$P(\omega)_{total} = \underbrace{K \left(1 + 3 \frac{\omega^2}{\omega_b^2} \right)}_{\text{Blade profile power}} + \underbrace{\frac{1}{2} \rho \omega^3 F}_{\text{Parasite power}} + \underbrace{mg \sqrt{\left(\frac{-\omega^2 + \sqrt{\omega^4 + \left(\frac{mUg}{\rho A} \right)^2}}{2} \right)}}_{\text{Induced power}}, \quad (5.4)$$

where ω_b represents the blade's rotor speed, K and F are two constants which depend on the dimensions of the blade and the UAV drag coefficient, respectively, ρ is the air density, m_U and g respectively denote the mass of the UAV and the standard gravity, A is the area of the UAV. The total energy consumption to cover a distance d at a constant speed UAV ω can be computed as

$$E(\omega)_{total} = \int_0^{d/\omega} P(\omega) dt = P(\omega) \frac{d}{\omega}, \quad (5.5)$$

5.3 Optimization Problem Formulation

The objective of this chapter aims at optimizing the UAVs' trajectories to minimize the number of UAVs that serve vehicles within the highway segment under the mobility of UAVs and vehicles constraints as well as the UAVs' energy budget constraint. To mathematically formulate the problem, we introduce a binary decision variable $\gamma_m \in \{0, 1\}$, $\forall m$, that takes the value of 1 if the UAV m is deployed and 0 otherwise, $y_{i,m}^n \in \{0, 1\}$, $\forall n, m, i \in V^n$ to indicate whether UAV m is serving vehicle i in time-slot n ; the binary variable $c_{i,m}^n \in \{0, 1\}$, $\forall n, m, i \in V^n$ to indicate whether vehicle i is covered by UAV m with an acceptable QoS r_{min} in time-slot n , $c_{i,m}^n$ is define as follows:

$$c_{i,m}^n = \begin{cases} 1, & \text{if } \sum_{m=1}^M y_{i,m}^n r_{i,m}^n > r_{min} \forall n, m, i \in V^n, \\ 0, & \text{otherwise.} \end{cases} \quad (5.6)$$

We also define the binary variable z_m indicating that the residual energy of UAV m is barely enough to travel to the charging station, z_m is define as follows:

$$\text{where: } z_m^n = \begin{cases} 1, & \text{if } E_m^n \geq E_{Travel} \quad \forall n, m, \\ 0, & \text{otherwise.} \end{cases} \quad (5.7)$$

where E_m^n is the residual energy of UAV m in time-slot n and E_{Travel} is the required energy for traveling to the charging station, respectively. In other words, once residual energy is less than the required energy to travel back to the charging station, immediately the deployed UAV m will be changed to out-of-service. Due to the untractability of equations (5.6) and (5.7), we introduce new binary variables $c_{i,m}^n$ and z_m ; and big number method to reformulate the coverage in (5.6) and the energy variable in (5.7) into the following:

$$c_{i,m}^n \geq \frac{\sum_{m=1}^M y_{i,m}^n r_{i,m}^n - r_{min}}{\Lambda}, \quad \forall n, m, i \in V^n, \quad (5.8a)$$

$$c_{i,m}^n < 1 + \frac{\sum_{m=1}^M y_{i,m}^n r_{i,m}^n - r_{min}}{\Lambda}, \quad \forall n, m, i \in V^n, \quad (5.8b)$$

$$z_m < 1 + \left(\frac{E_m^n - E_{Travel}}{\Lambda} \right) \quad \forall n, m, \quad (5.9a)$$

$$z_m \geq \left(\frac{E_m^n - E_{Travel}}{\Lambda} \right) \quad \forall n, m, \quad (5.9b)$$

where Λ is a large number that is used to ensure the validity of the above equations. We represent the UAVs trajectories by $\mathbf{W} = [(w_m^n, 0, H), \forall n]$, the required UAVs by $\mathbf{K} = [\gamma_m, \forall m \in M]$, the UAV energies by $\mathbf{Z} = [z_m, \forall m]$, the UAV serving indicator by $\mathbf{Y} = [y_{i,m}^n, \forall n, m, i \in V^n]$, and the coverage indicator by $\mathbf{C} = [c_{i,m}^n, \forall n, m, i \in V^n]$. To this end, our optimization problem is formulated as:

$$\begin{aligned} \mathcal{O} \mathcal{P}: \quad & \max_{\mathbf{W}, \mathbf{Y}, \mathbf{Z}, \mathbf{K}, \mathbf{C}} \psi \sum_{\forall n} \sum_{i \in \mathcal{A}} \sum_{m=1}^M c_{i,m}^n - \xi \sum_{m=1}^M \gamma_m \\ \text{s.t.} \quad & \mathcal{C}1: c_{i,m}^n \geq \frac{\sum_{m=1}^M y_{i,m}^n r_{i,m}^n - r_{min}}{\Lambda}, \forall n, m, i \in V^n, \\ & \mathcal{C}2: c_{i,m}^n < 1 + \frac{\sum_{m=1}^M y_{i,m}^n r_{i,m}^n - r_{min}}{\Lambda}, \forall n, m, i \in V^n, \\ & \mathcal{C}3: |w_m^{n+1} - w_m^n| \leq \gamma_m V_{\max} \delta_t, \quad n = 1, \dots, N-1, \forall m, \\ & \mathcal{C}4: y_{i,m}^n, \gamma_m, z_m, c_{i,m}^n \in [0, 1], \quad \forall n, m, i \in V^n, \\ & \mathcal{C}5: y_{i,m}^n \leq \gamma_m, \quad \forall n, m, i \in V^n, \\ & \mathcal{C}6: \sum_{m=1}^M y_{i,m}^n \leq 1, \quad \forall n, i \in V^n, \end{aligned}$$

$$\begin{aligned}
\mathcal{C}7 : y_{i,m}^n &\leq z_m, \quad \forall n, m, i \in V^n, \\
\mathcal{C}8 : z_m &< 1 + \left(\frac{E_m^n - E_{Travel}}{\Lambda} \right), \quad \forall m, \\
\mathcal{C}9 : z_m &\geq \left(\frac{E_m^n - E_{Travel}}{\Lambda} \right), \quad \forall m, \\
\mathcal{C}10 : (1 - \gamma_m)w_m &= w_s, \quad \forall m,
\end{aligned}$$

where ξ and ψ are weight parameters, and $\xi + \psi = 1$. A larger value for ψ will render the coverage the dominant factor, so the solution should deploy a UAV for a small number of vehicles (or just one vehicle), which economically could be expensive for the operator; a larger value for ξ will render deploying a new UAV the dominant factor, hence, the solution will provide a non-continuous coverage with less deployed UAVs.

Constraints $\mathcal{C}1$ and $\mathcal{C}2$ guarantee that each vehicle is covered with an acceptable QoS r_{min} (bps/Hz) within their residence on the highway segment. $\mathcal{C}3$ limits the distance traveled by the deployed UAV m in one time-slot based on its maximum speed. $\mathcal{C}5$ constrain the serving to UAVs that are dispatched. $\mathcal{C}6$ ensures that one vehicle is served by at most one UAV at a time. Constraint $\mathcal{C}7$ ensures that once a UAV m is used, the UAV will continue deployed as long it has energy to serve. Constraints $\mathcal{C}8$ and $\mathcal{C}9$ ensure that the residual energy of UAV m for traveling is sufficient enough to serve and fly before it runs out of energy. Finally, $\mathcal{C}10$ indicates the initial positions of the UAVs.

We observe that $\mathcal{O}\mathcal{P}$ is a mixed integer non-linear program (MINLP), due to the existence of the binary variables $y_{i,m}^n, \gamma_m, z_m$, and $c_{i,m}^n$ in $\mathcal{C}4$ and non-convex constraints $\mathcal{C}1$ and $\mathcal{C}2$ [10], even if the binary variables $y_{i,m}^n, \gamma_m, z_m$, and $c_{i,m}^n$ are relaxed to take any value between 0 and 1. The relaxed version of $\mathcal{O}\mathcal{P}$ is, nevertheless, non-convex due to the trajectory variable w_m^n in $\mathcal{C}1$ and $\mathcal{C}2$. To the best of our knowledge, there is no solver for solving $\mathcal{O}\mathcal{P}$ efficiently.

Clearly, the solution of the $\mathcal{O}\mathcal{P}$ (if it exists), which yields a trajectory for a minimum number of UAVs during a time frame N , relies on the knowledge of the instantaneous position of vehicles at each time-slot during their residence on the highway segment; given that by the time a trajectory is designed, there is no possible way of obtaining the instantaneous position in future slots, and thus we cannot properly solve $\mathcal{O}\mathcal{P}$. Unrealistic assumptions lead to inaccurate solutions with an excessive

complexity. In order to solve this problem at a low complexity and find an optimal solution, a Deep Reinforcement Learning algorithm will be invoked in next section.

5.4 The Proposed Deep Reinforcement Learning Approach

In this work, an Actor-Critic agent is deployed at the central unit, and interacts with the vehicular environment in a sequence of actions, observations, rewards and penalties. At each time-slot n , the agent selects an action from the feasible continuous actions at that time. The deployed UAVs will either travel along the highway in a specific direction or hover to serve the vehicles in a fixed position. It is important to understand, the real trajectory of UAVs can fly in arbitrary distances without any mobility constraint below the maximum speed. The agent then observes the dynamic changes in the vehicular environment and modifies the system state representation. The agent also receives a reward or penalty accordingly. In order to achieve the maximum effective coverage on the highway, all UAVs should operate in a consistent, orderly and energy efficient way to provide the vehicles with acceptable QoS. After each selected action (either traveling or hovering), each UAV receives a step reward which is a normalized indicator of how well the selected action accomplishes the previously-mentioned goals. The objective of Actor-Critic is to construct an optimal action selection policy for each UAV that covers the vehicles along the highway segment in order to achieve an effective coverage with acceptable QoS. It is worth mentioning that the received reward by each UAV depends on the entire previous sequence of actions and the observations from the vehicular environment. As such, the impact of the action may only be seen after several time steps. In the following, we first briefly review AC, a learning technique which is suitable for controlling autonomous machines such as UAVs. Then, we introduce our approach using Actor-Critic for efficient UAV coverage.

5.4.1 Deep Reinforcement Learning Background

Standard Reinforcement Learning is a branch of machine learning paradigm, which deals with multi-state decision process of a software agent (a central unit in our case) while interacting with an environment in discrete decision epochs. In general, RL assumes the system consists of multiple

states S , where at each epoch n , the agent observes state $s_n \in S$, executes action a_n from a finite number of actions A according to an agent's policy π (i.e., the next UAVs' position) and receives a reward r_n , and moves to the next state s_{n+1} .

The goal of RL is to learn from the transition tuple $\langle s_n, a_n, r(s_n, a_n), s_{n+1} \rangle$, and find an optimal policy π^* that will maximize the discounted cumulative sum of all future rewards. Note that the policy $\pi = \{a_1, a_2, \dots, a_N\}$ defines which action a_n should be applied at state s_n . If we let $r(s_n, \pi(a_n))$ denote the reward obtained by choosing policy π , the cumulative discounted sum of all future rewards using policy π is given by:

$$R_\pi = \sum_{n=1}^N \lambda^{n-1} r(s_n, \pi(a_n)), \quad (5.11)$$

where $\lambda \in [0, 1)$ is a discount factor, which measures the weight given to the future rewards.

Q-learning is one of the widely used methods of RL algorithms, which allows the agent to optimally act in an environment represented by a Markov decision process (MDP). Q-learning iteratively improves the state-action value function (also known as Q-function or Q-value), and by estimating the future reward if action a_n is taken, the agent presents the higher probability of going from state s_n to s_{n+1} using policy π . The Q-value function is usually stored in a table. However, Q-learning only works with a low-dimensional finite discrete action state space. DRL is a deep version of RL, where one (or multiple) deep neural networks (NNs) is used as the approximator of the action-value function $Q(\cdot)$. Deep Q-Network approach is one of the approaches of DRL, where a single neural network (NN) is trained through minimizing a loss function L , as follows:

$$L(\theta^Q) = \mathbb{E}[T_n - Q(s_n, a_n | \theta^Q)], \quad (5.12)$$

where θ^Q are the function parameters (weights) of Deep NN; and T_n is a target value, which can be computed by

$$T_n = r_n + \lambda^{n-1} \max_{a_{n+1}} Q(s_{n+1}, a_{n+1}). \quad (5.13)$$

However, Deep Q-Network tends to diverge with the non-linear function approximator. Some

techniques are utilized in order to avoid the divergence of Deep Q-Network, namely: experience replay, fixed target network and reward normalization [87]. In experience replay, a random mini-batch of samples from the past experience is used during the training process to reduce the correlation between samples. In addition, in fixed target network, the same NNs' parameters are used to calculate the target function. Reward normalization techniques are used to limit the scale of the error derivatives and ensure the stability of the algorithm. However, it is unfeasible to apply both Q-learning and Deep Q-Network to continuous control because it is necessary to figure out the value for each action that maximizes the Q-function, which is quite difficult. Deep Deterministic Policy Gradient (DDPG) [78] with the assistance of experience replay, fixed target network and reward normalization techniques was designed for continuous control actions that uses an Actor-Critic approach, that is, the use of two Deep NNs namely actor and critic networks, where critic network is a Deep Q-Network, which is represented as $Q(s_n, a_n | \theta^Q)$. Therefore, the same loss function with different parameters is used for training the actor and critic networks, θ^Q and θ^π respectively. The actor network $\pi(s_n | \theta^\pi)$ is trained to obtain the optimal actions a_n for a given states s_n . The actor network is updated by applying the chain rule to the expected return from the start distribution J with respect to the actor parameter θ^π [78]:

$$\nabla_{\theta^\pi} J \approx \mathbb{E} \left[\nabla_a Q(s, a | \theta^Q) \Big|_{s=s_n, a=\pi(s_n)} \cdot \nabla_{\theta^\pi} \pi(s | \theta^\pi) \Big|_{s=s_n} \right]. \quad (5.14)$$

The weights of these networks are then updated by having them slowly track the learned networks $\theta' := \tau\theta + (1 - \tau)\theta'$, with $\tau \ll 1$.

For more information on Deep Deterministic Policy Gradient, the reader is referred to [78]. The next subsection presents the system state representation as well as the rewards and penalties associated with the agent's actions.

5.4.2 Input From the Environment

At the beginning of the coverage mission, the agent observes the vehicular network environment that defines the states of the system, collects all the parameters associated with the set of in-range vehicles, and executes an action for each UAV at time-slot n . The input of UAVs from the vehicular

environment at time-slot n is:

- E_n^m : a vector of size M containing the remaining energy of each UAV at time-slot n , where $m \in \mathcal{M}$, $0 \leq E_n^m \leq E_{total}$, where E_{total} is the total energy of each UAV.
- V^n : the number of vehicles residing within the considered highway segment, at time-slot n .
- x_i^n : a vector of size V^n containing the instantaneous position of vehicle $i \in (1, 2, \dots, V^n)$, at time-slot n .
- w_m^n : a vector of size M containing the ground level position of each UAV, at time-slot n .
- γ_m^n : a vector of size M containing the status of the UAVs whether UAV m is deployed or not, at time-slot n .
- C_i^n : a vector of size V^n , containing the coverage indicators of each vehicle. If $C_i^n = 1$, vehicle i lies within the coverage of a UAV at time-slot n ; otherwise, $C_i^n = 0$. To this end, the coverage indicators C_i^n at time-slot n can be written as:

$$C_i^n = \begin{cases} 1, & \text{if } \sum_{m=1}^M y_{i,m}^n r_{i,m}^n \geq r_{min} \wedge \sum_{m=1}^M y_{i,m}^n \leq 1, \\ & \forall i \in \mathcal{V}^n, a_i \leq n \leq d_i, \\ 0, & \text{otherwise,} \end{cases} \quad (5.15)$$

where $y_{i,m}^n \in \{0, 1\}$, $\forall m$ is a binary decision variable, that takes the value of 1 if the vehicle i is served by UAV m and 0 otherwise.

Each UAV fully observes the current vehicular network environment and updates the central unit which is able to realize the system state representation s_n at time-slot n .

5.4.3 Actions and Expected Rewards

At each step, each UAV m carries out an action a_m^n which represents a traveling distance d_n^m in a specific direction Φ_m , depending on its current state. The UAVs may travel with arbitrary velocities in different directions, which makes the problem non-trivial to be solved. However, by assuming

that the width of the highway lane is ignored as compared to the transmission range of vehicles and UAVs [88], the model may be simplified to as few as two directions (left and right) in the middle of the highway. Hence, at time-slot n , each UAV chooses its action (distance and direction), and accordingly, the vehicular network environment pays an immediate reward; that is, a scalar value that reflects the righteousness of the UAVs' actions. The immediate reward r_n is the sum of the following normalized quantities:

- Penalty incurred on network due to the existence of a vehicle within the highway without UAVs' coverage: the value of this penalty is a normalized quantity proportional to the coverage indicator of each vehicle. As such, the UAVs are encouraged to cover the vehicles within the considered highway segment. Recall that a vehicle communicates its exit point upon its arrival to the highway, and the UAVs should coordinate to continuously cover that vehicle within the highway. The coverage penalty due to non-coverage can be written as:

$$\mathcal{P}_c^n = \xi \sum_{i \in \mathcal{V}^n} 1 - C_i^n, \quad (5.16)$$

where ξ is weight with a high value to avoid UAVs missing to cover a vehicle.

- Penalty incurred on network due to the deployment of a new UAV: the network receives this penalty when the current deployed UAVs cannot cover the newly arrived vehicles and a new UAV is required to be deployed. The value of this penalty is proportional to the number of deployed UAVs. As a result, the network learns to optimize the trajectories of the minimum number of UAVs to cover the current and newly arrived vehicles. The deployment penalty due to the deployment of a new UAV can be written as:

$$\mathcal{P}_U^n = \psi \sum_{m=1}^M \gamma_m, \quad (5.17)$$

where γ_m is binary variable that takes the value of 1 if UAV m is deployed and 0 otherwise, and ψ is weight with a high value to avoid unnecessary deployment of UAVs.

- Penalty incurred on network if the residual energy of each UAV exceeds the required energy for traveling to the charging station: the Actor-Critic agent strives to maximize its rewards

(minimize negative rewards, i.e., costs), it learns how to minimize the total energy consumption of UAVs to serve more vehicles and avoid this penalty. This penalty is referred as the energy penalty.

- Penalty incurred on network if the deployed UAV flies outside the given highway segment: the Actor-Critic agent learns how to continue the flying on the given highway segment.

Obviously, we are dealing with an infinite control task since each UAV can carry out infinite actions (traveling distance), and hence the use of Actor-Critic techniques is necessary to solve our problem. Now, it is important to mention that the reward function is not dependent only on one UAV but on a joint actions of all UAVs. It is noteworthy that even if the impact of the occurrence of the above described event is unveiled in a single time-step (i.e., when a vehicle arrives to the considered highway segment or departs from it), the Actor-Critic agent realizes that the deployed and non-deployed UAVs and their previous trajectories lead to this current system state. This is a clear example that the feedback from an action may sometimes be delayed after many thousands of time-slots have elapsed.

In fact, the most recent observation such as the number of vehicles, their positions, their coverage status and the current position of the deployed UAVs is completely sufficient statistic of the history to make a decision. In other words, the future is independent of the past given some current aggregate statistic about the present which satisfy the Markov property.

5.4.4 Solution Algorithm

Recall that, our objective is to find a control policy that governs the trajectories of the deployed UAVs at each time-slot to achieve an effective coverage with a minimum number of UAVs under energy budget. This problem has been formulated as an MDP whose vehicular environment states are modeled as a Markov chain.

The implementation of the proposed DRL approach is shown in Fig. 5.2, which is composed of the vehicular environment, the coverage reward including the penalties, an actor-network, a critic network, and a temporal difference error. The vehicular environment can be observed by the UAVs, which is then sent to the central control agent, where the actor and critic networks decide the best

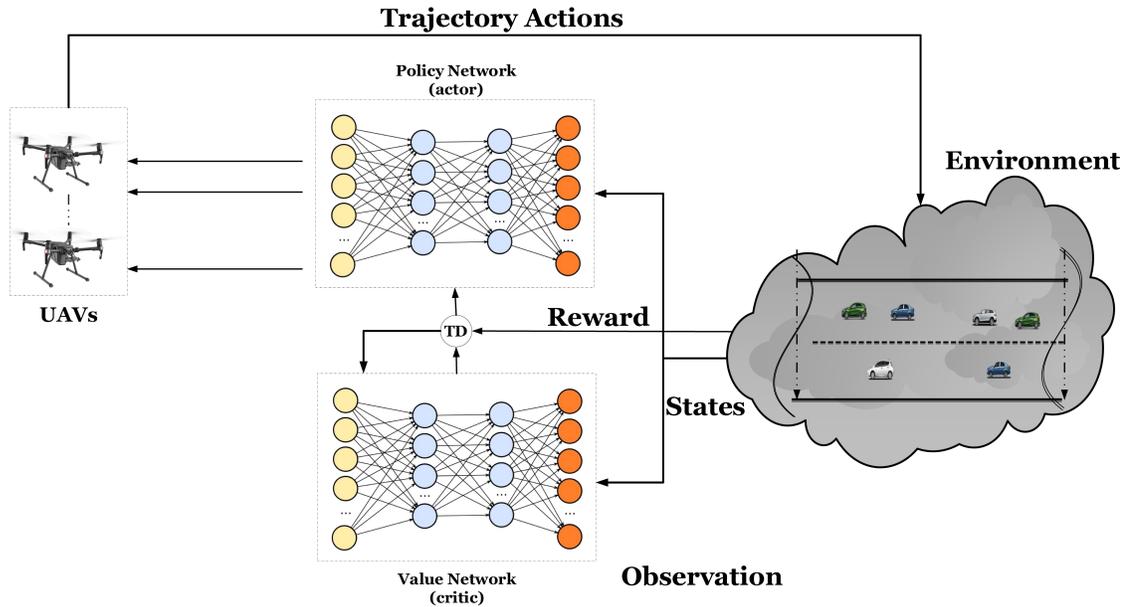


Fig. 5.2: DRL-based proposed approach to obtain the control policy that governs the trajectories of the deployed UAVs.

control policy for the deployed UAVs. As mentioned before, since we are dealing with continuous control actions to obtain the trajectories of UAVs, we adapt the Deep Deterministic Policy Gradient (DDPG) to solve our problem. The DRL algorithm to obtain UAVs' trajectories is presented in Algorithm 1. The proposed algorithm works as follows.

In the first part, after defining the input and output of the algorithm (Lines 1-2), the proposed algorithm randomly initializes the replay buffer of size Z , the weights parameters for the actor-network θ^π and critic network θ^Q (Lines 3-4). Further, as mentioned in Subsection (5.4.1), we create the target networks $\pi'(\cdot)$ and $Q'(\cdot)$ to enhance the training stability, where the target, critic and actor networks have the same structures. The target network weight parameters $\pi'(\cdot)$ and $Q'(\cdot)$ are initialized (Line 5), where at later steps (Lines 12- 23), those parameters are slowly updated according to the control parameter $\tau = 0.001$ in order to enhance the stability.

The exploration phase, reward, and penalties are explained in the second part (Lines 6-29). During the exploration phase, the algorithm obtains a trajectory action from the current actor-network θ^π bounded with the maximum velocity of the UAVs, ω_{max} , and then a random noise is added that decays over time with a rate of 0.9995, where the random noise is generated from a uniform

distribution with a zero mean and a variance of 1. During the training phase, the proposed algorithm guide the Actor-Critic agent to avoid actions that violate the highway border (i.e., flies outside the given highway segment) by applying a specific penalty to the reward (Lines 13-15), where, a penalty p is deducted from the overall reward, and the corresponding trajectory action of the UAV m is cancelled. Likewise, the proposed algorithm trains the agent to stop serving and return to the charging station if the residual energy is below a threshold. Furthermore, during the training phase, the proposed algorithm trains the Actor-Critic agent to dispatch the minimum number of UAVs by applying a one-time penalty for each dispatching UAV as shown in Algorithm 2. During the UAVs' trajectories, the deployed UAVs are serving the vehicles according to closest distance as mentioned in Algorithm 3. In our algorithm, the defined penalties are set to a large value compared to the corresponding reward, which is 5 times.

In the last part, the weights and parameters of the neural network (Lines 24-34) are updated according to the DDPG algorithm. Firstly, the collected samples including (s_n, a_n, r_n, s_{n+1}) are stored in the replay buffer of size Z after each executed action, and then a random mini-batch of size H is sampled from the buffer Z to updated the actor and critic networks. As explained in Subsection (5.4.1), the weights parameter of the critic network are updated to minimize (5.12), while the actor-network weights parameters are updated according to (5.14).

5.4.5 Complexity Analysis

In this subsection, the complexity analysis is discussed. After adequate training, the Deep Reinforcement Learning agent observes the vehicular network environment that previously defined states as input, the Deep Reinforcement Learning agent utilizes its trained actor network $\pi(s|\theta^\pi)$ to carry out an action a_m^n which represents a traveling distance and direction. Based on [89], the total computational complexity for the fully connected layers can be expressed as the number of multiplications: $O(\sum_{p=1}^P n_p \cdot n_{p-1})$, where n_p is the number of neural units in fully-connected layer p .

5.5 Simulation Results and Discussion

In this section, we evaluate the performance of the proposed algorithm numerically. The main input parameters that are used in this simulation are listed in Table 5.1. In order to deliver realistic results, the simulation parameters should be an accurate representation of a real highway scenario. It is assumed that a highway segment of length 5km is simulated, on which multiple UAVs are ready to be dispatched to ensure a network coverage to vehicles. The flow of vehicles entering the highway segment follows a Poisson distribution that is used to run the simulation; we generate 2.4 million samples each corresponding to a snapshot of the system at a particular time-slot. Vehicles velocities are randomly generated using a truncated Gaussian distribution with mean equal 27.5m/s, variance 4.5m/s, and velocities can be varied between 22 – 33m/s, where the vehicles randomly change their velocities within the given highway according to a normal distribution.

In our simulation, 2-layer fully connected neural network is used for each network (i.e., the actor and critic networks), which includes 20 and 80 neurons in the first and second layers respectively, and utilized the rectified linear unit (ReLU) function for activation for both networks. As for, the activation function, hyperbolic tangent (tanh) is utilized in the last layer for that actor-network to limit the traveling actions according to the maximum traveling distance of the UAVs. The generated samples are used to train the deep neural network using Tensor Processing Unit (TPU) to realize an optimal trajectory for the deployed UAVs. After establishing the optimal trajectory determined by the proposed algorithm, another set of mobility traces was used to test the performance of the proposed trajectory policy.

We start by first studying the convergence performance of the proposed DRL algorithm. The total reward is calculated as the summation of the cost of each action for UAVs, which is the weighted sum of the penalty of UAVs' deployment, coverage for each vehicle and energy consumption of UAVs. As shown in Fig. 5.3, it can be seen that the cumulative reward increases very fast over time at the beginning of learning. This is because, at the beginning of the training, the deployed UAVs start to learn the border of the highways to avoid the penalties due to flying outside the border. Moreover, many vehicles were not yet covered since the UAVs did not learn the suitable trajectories in the dynamic environment to cover the vehicles. After that, the trained UAVs can result in

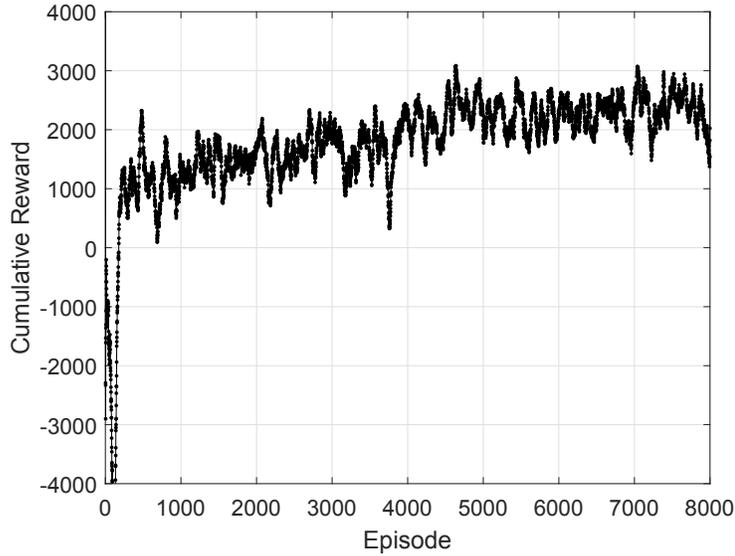


Fig. 5.3: Accumulated reward over time.

significant improvement in the reward. Furthermore, this improvement starts to diminish when the deployed UAVs are well trained about the borders of the highway and they start to effectively cover the vehicles. It is worth mentioning that due to the non-stationarity (stable dynamics) of the environment, the reward is highly varying around the average while on average the cumulative reward is increasing with training. A similar observation has been made in [90].

To better understand the impact of the dynamics in the environment of the vehicular network, we simulate a scenario that takes into account accurate prior knowledge about vehicles' instantaneous positions. As shown in Fig. 5.4, the performance of the algorithm can be drastically different. The proposed algorithm converges very fast; starting from around the 300-th episode the algorithm already converges. The high convergence rate stems from the prior knowledge about the environment as well as the adopted DDGP algorithm in which the critic-network judges and guides the actor-network to learn the suitable trajectories in advance.

To observe the efficiency of the proposed Deep Reinforcement Learning algorithm in terms of time, its performance is compared with the maximum performance. This result is presented in Fig. 5.5, which clearly indicates that our algorithm requires few hours, 16 hours, to learn the dynamics of the vehicular environment in order to attain a good performance, 70%. It can also be observed

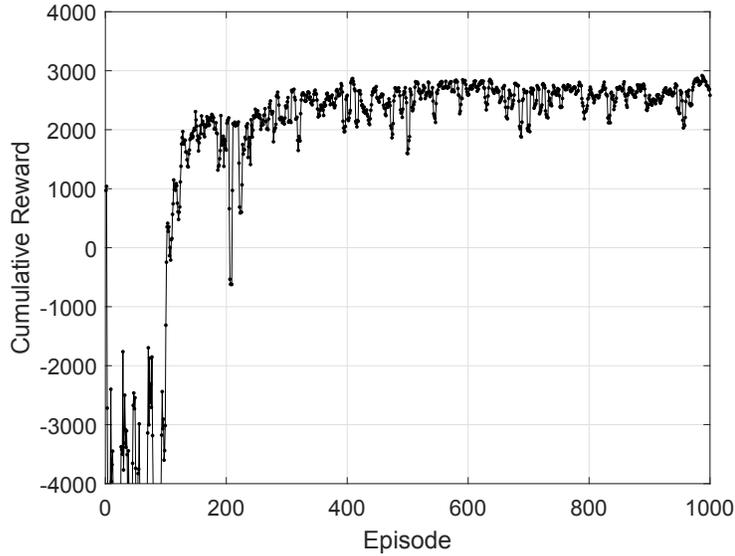


Fig. 5.4: Accumulated reward over time for with prior knowledge.

that more samples/updates from the vehicular environment are beneficial for improving the performance. This is quite reasonable considering the fact that the global information about the vehicular environment is unknown and frequently changed.

The percentage of average coverage is another performance metric we study. Fig. 5.6 depicts this metric versus the number of deployed UAVs, and for different minimum rates (in bps/Hz) with vehicular density 12Veh/Km. Clearly, as we increase the minimum rate r_{min} , the UAVs will adapt its trajectory to fly closer to a vehicle or subset of vehicles to meet their requirements, therefore, by increasing the minimum rate more UAVs are needed to fulfill the requirements of vehicles for the same average coverage. For example, to achieve the minimum rate of 11bps/Hz with 78% average coverage, 2UAV are required. The same average coverage can be achieved for the minimum rate (i.e. $r_{min} = 12\text{bps/Hz}$) by increasing the number of UAVs, where 5UAVs become significantly needed to fulfill the requirements of vehicles with the same percentage. It is also obvious that while increasing the number of deployed UAVs the proposed algorithm achieves higher average coverage for the same minimum rate.

Next, we study the impact of vehicular density on the proposed DRL solution for different minimum rates (in bps/Hz). As shown in Fig. 5.7, at lower vehicular density with higher requirements, one dispatched UAV is able to serve only a few number of vehicles, and this is due to the fact that

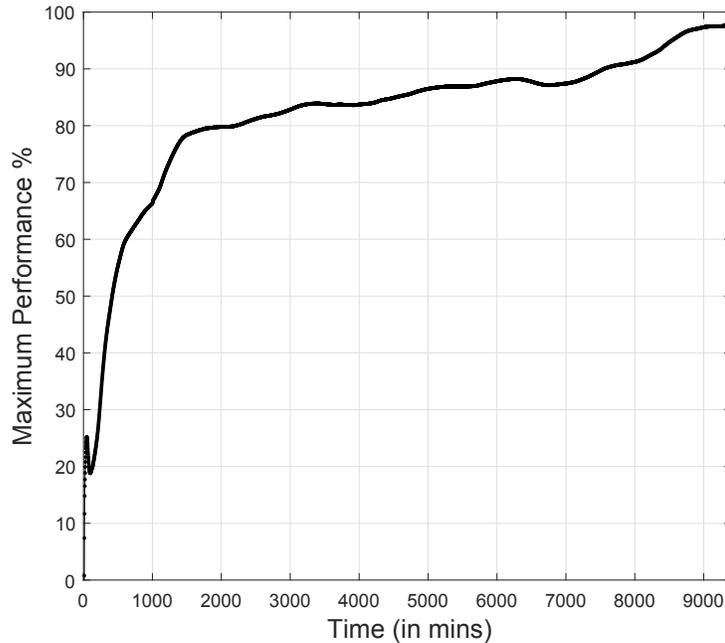


Fig. 5.5: Performance vs time.

the dispatched UAV optimizes its trajectory to fly closer to vehicles to fulfill the vehicles' requirements and wastes more time in flying to reach other vehicles, therefore, the UAV covers a fewer number of vehicles. Surprisingly, when the vehicular density increases, our proposed algorithm covers more vehicles, since with increasing the vehicular density, the vehicles' velocities decrease and thus enjoys more services. For instance, when the vehicular density is 4Veh/Km for 2UAVs with minimum requirements of 11bps/Hz, the proposed algorithm covers 10% more vehicles compared to 4Veh/Km. This shows the efficiency of the proposed framework in achieving effective coverage for the vehicles within the given highway segment since the major goal is to maximize the vehicular coverage. We can also observe from the figure that when the vehicular density increases the average coverage decreases as expected since more UAVs are required to fulfill the vehicles' requirements.

We next compare our proposed approach with three others trajectories approaches for the minimum rate (i.e. r_{min} 4bps/Hz) to show the efficiency of our proposed approach: 1) Random UAV dispatching approach where, at random time based on a normal distribution, the central unit randomly dispatches one UAV with maximum speed, 50m/s, to serve the vehicles within the highway segment. 2) Fixed Dispatching Rate, where the central unit decides to dispatch one UAV every

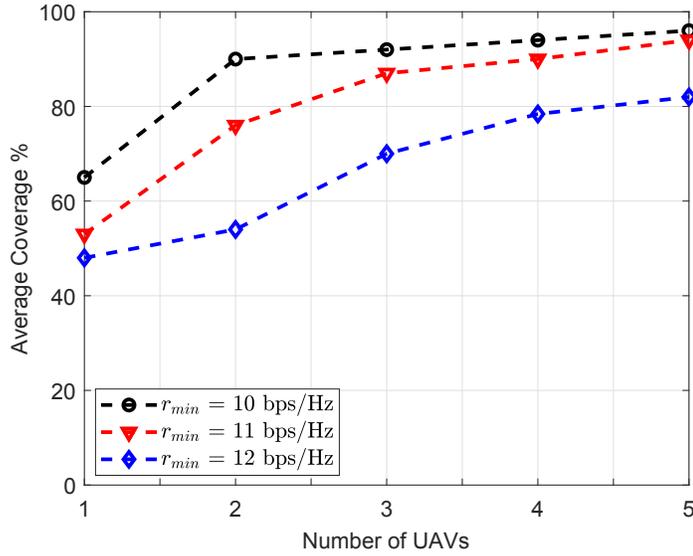


Fig. 5.6: Impact of r_{min} .

time period n' , ever 35sec, with the maximum speed. 3) Fixed Hovering UAVs, where UAVs are hovering a fixed distance, every 1Km, to serve the vehicles. It can be seen in Fig. 5.8, the proposed algorithm consistently outperforms other approaches in term of number of required UAVs. For example, to achieve 100% average coverage the proposed algorithm achieves a lower number of deployed UAVs compared to the other approaches because the former provides more flexibility for the UAV to predict and adapt its trajectory to serve the vehicles. In contrast, fixed and random dispatching approaches, the fixed velocity does not have a significant impact on coverage, which well justify the robustness of the proposed algorithm in terms of coverage.

Finally, we show the impact of energy penalty on the energy consumption and the average coverage with the minimum rate (i.e. $r_{min} = 12\text{bps/Hz}$). From Fig. 5.9, we can see that, while considering the energy penalty on the deployed UAVs, the proposed algorithm almost achieves the same coverage with less energy consumption. For example, in Fig. 5.9(a), when the average coverage is 80% with 5UAVs, the average energy consumption while applying the energy penalty is 16% reduction compared to without applying as shown in Fig. 5.9(b). We can make an interesting observation that the proposed algorithm choose the action that achieves the almost same coverage with less energy consumption, which can somehow reduce the energy consumption. This is a clear implication of the penalty incurred on the Actor-Critic agent, in the training phase, due to the impact

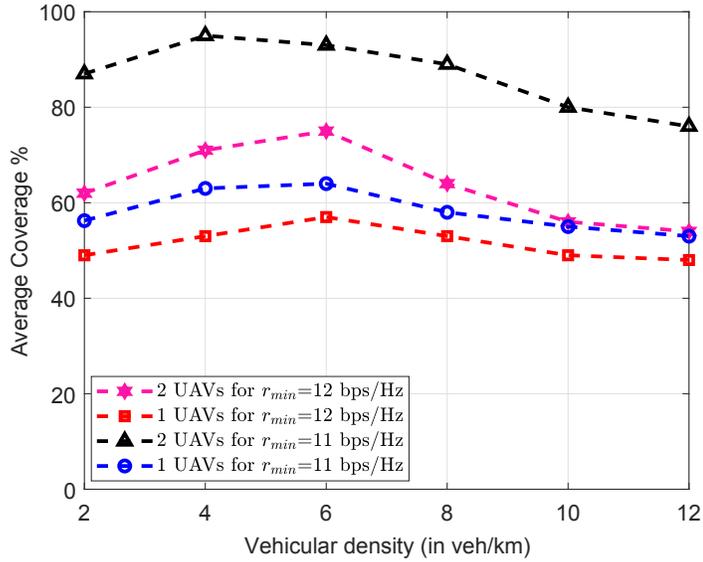


Fig. 5.7: Impact of vehicular density.

of the penalty when the residual energy exceeds the required energy for traveling to charging station. Recall that since this penalty is proportional to the residual energy of the deployed UAVs, the Actor-Critic agent learns to minimize the energy consumption of the UAVs in order to avoid penalizing its total rewards.

5.6 Summary

In this work, we proposed the Deep Reinforcement Learning framework that controls the trajectories of multiple UAVs to efficiently cover vehicles in a dynamic environment where communication infrastructure is not available. Specifically, the proposed approach maximizes the vehicular coverage with the minimum number of UAVs with minimum energy consumption. It was demonstrated that the proposed algorithm was capable to learn the vehicle environment and its dynamics to control the UAVs to provide effective coverage for the vehicles. Our results showed that our proposed solution outperformed alternative approaches including fixed and random deployment approaches, and static UAV placement in terms of the percentage of average coverage (average improvement of 40%).

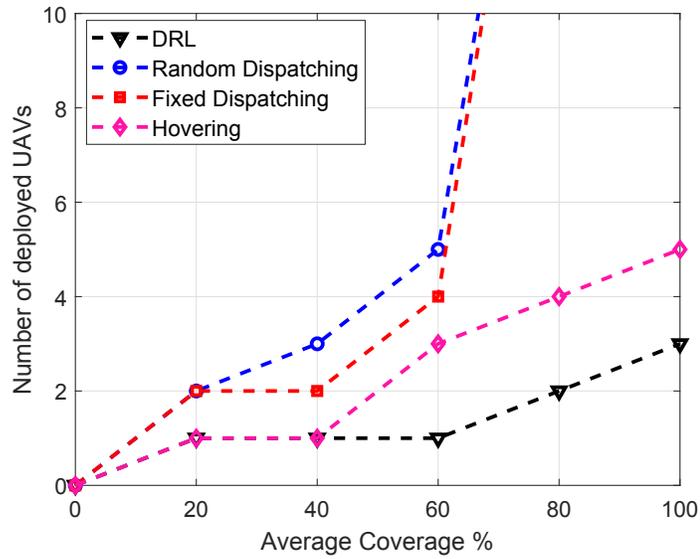


Fig. 5.8: Performance evaluation and comparisons.

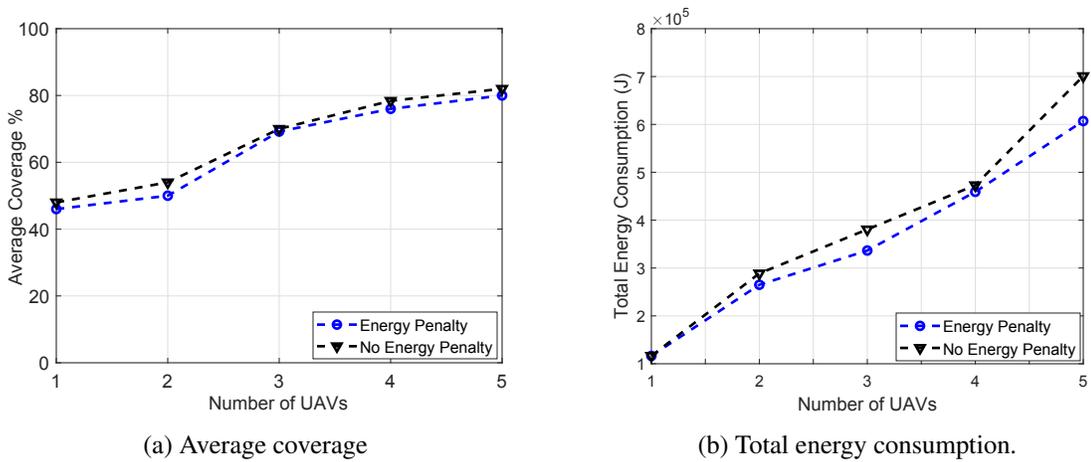


Fig. 5.9: Impact of energy saving.

Algorithm 6: Proposed Solution: DRL to obtain UAVs' trajectories.

```

1 Input: Discount factor, learning rate for actor and critic network, buffer size, mini-patch
  size, UAV energy parameters, penalties;
2 Output: The trajectories of UAVs.
3 Initialize replay buffer  $Z$ .
4 Randomly initialize critic network  $Q(s, a | \theta^Q)$  and actor network  $\pi(s | \theta^\pi)$  with weights  $\theta^Q$ 
  and  $\theta^\pi$ ;
5 Initialize target networks  $Q'$  and  $\pi'$  with weights  $\theta^{Q'} \leftarrow \theta^Q, \theta^{\pi'} \leftarrow \theta^\pi$ ,
6 for  $episode = 1, P$  do
7   Collect network characteristics to realize state  $s_1$ .
8   for all  $n \in N$  do
9     Observe:  $E_n^m, V^n, s_i^n, x_i^n, w_m^n$ , and  $C_i^n$ ,
10    Execute: Select action  $a_n^m = \pi(s_n)$ , and add a random noise that decays over time;
11    Evaluate: obtain the reward  $r_n$  and  $s_{n+1}$ ,
12    for UAV  $m := 1, \dots, M$  do
13      if UAV  $m$  (flies outside the border) then
14         $r_n = r_n - P$ .
15        Cancel the movement of UAV  $m$  and update  $s_{n+1}$ .
16      else if The residual energy,  $E_m^n$ , of UAV  $m$  is less than  $E_{Travel}$ . then
17         $r_n = r_n - P$ .
18        Changed the status of UAV  $m$  to out-of-service, mark UAV  $m$ , and update
19         $M$ .
20      else
21        Apply Algorithm 7 and update  $r_n$ .
22        Apply Algorithm 8 and update  $r_n$ .
23      Store transition  $(s_n, a_n, r_n, s_{n+1})$  in  $Z$ .
24      Sample random mini-batch of transitions  $(s_n, a_n, r_n, s_{n+1})$  of size  $H$  samples from  $Z$ .
25       $T_n := r_n + \lambda Q'(s_{n+1}, \pi'(s_{n+1} | \theta^{\pi'}) | \theta^{Q'})$ ;
26      Update weights  $\theta^\pi$  of  $Q(\cdot)$  by minimizing the loss:
27       $L(\theta^Q) = \frac{1}{H} \sum_{n=1}^H (T_n - Q(s_n, a_n))^2$ .
28      Update the weights  $\theta^\pi$  of  $\pi(\cdot)$ :
29       $\nabla_{\theta^\pi} J \approx \frac{1}{H} \sum_{l=1}^H \nabla_a Q(s, a | \theta^Q) \Big|_{s=s_l, a=\pi(s_l)} \nabla_{\theta^\pi} \pi(s | \theta^\pi) \Big|_{s=s_l}$ ;
30      Update the corresponding target networks:
31       $\theta^{Q'} := \tau \theta^Q + (1 - \tau) \theta^{Q'}$ ;
32       $\theta^{\pi'} := \tau \theta^\pi + (1 - \tau) \theta^{\pi'}$ ;

```

Algorithm 7: Minimizing the Number of UAV Algorithm.

```
1 Total number of available UAVs and their current positions.
2 Set the binary variables  $\gamma_m = 0 \forall m$ .
3 for UAV  $m := 1, \dots, M$  do
4   while ( $\gamma_m == 0$  and  $w_m^n > 0$ ) do
5      $r_n = r_n - P$ .
6     Change the status of the UAV  $\gamma_m = 1$ .
```

Algorithm 8: Vehicle Admission Algorithm.

```
1 Sort all vehicles based on the distance to the current location of the UAV  $m$ ,  $d_{i,m}$ , where the
   closest vehicle is at the top of the list.
2 for Vehicle  $i := 1, \dots, V^n$  do
3   Select the closest unmarked vehicle to the current location of the UAV  $m$ .
4   while ( $r_i^n \geq r_{min}$ ) do
5     Mark vehicle  $i$ , increase  $r_n$ , and update the number of served vehicles.
```

Table 5.1: Simulation Parameters in Intelligent Coverage Networks

Parameter	Value
Minimum vehicle speed, v_{min}	22m/s
Maximum vehicle speed, v_{max}	33m/s
UAV max speed, ω_{max}	50m/s
highway segment of length d	5Km
Rotor speed, ω_b	100
Blade dimension constant, K	570
Air density, ρ	1.225
Drag and reference area coefficient, F	0.4
UAV mass, m_U	5Kg
UAV surface area, A	$0.25m^2$
Buffer size	10000
Patch size	120
Activation functions	ReLU and tanh
Number of Layers	2
Learning rate for actor	0.001
Learning rate for critic	0.002
Reward discount	0.8
action variation	50
Decay the action randomness	0.995
Soft replacement value	0.01
Optimizer technique	Adam
UAV altitude, H	100m
Channel power gain, γ_0	-50dB
Noise power, σ^2	-110dBm

Chapter 6

AoI-Aware Data Collection in Vehicular Networks with Intelligent UAVs

6.1 Background, Related Works, and Contributions

Maintaining information freshness is a key requirement for Intelligent Transportation System (ITS) applications such as autonomous intersection management, traffic control and autonomous driving. Such ITS applications rely on the timeliness of the real-time information such as drivers' behaviour, emergency braking, etc. This information is generated by a large number of LIDAR sensors on board intelligent and internet-connected vehicles. Therefore, collecting timely and fresh information is becoming more critical for safety and for enhancing driving assistance [92]. This information may be collected either through cellular networks or through Road Side Units (RSUs), which subsequently will be processed at edge servers for proper analytics. Nonetheless, terrestrial networks may either be not deployed (e.g., in rural areas), damaged or temporarily unavailable (e.g., during maintenance periods or natural disasters). Hence, UAVs are expected to play an important role in future networks for enabling more connectivity.

The work done in this chapter leads an IEEE published journal [91]

There are two different paradigms to acquire real-time information in Internet of vehicles settings, non-cooperative and cooperative. In the non-cooperative paradigm, UAVs can infer information about the vehicles by relying on its sensors/cameras. This paradigm suffers from high-cost equipment, highly impacted by weather, and cannot reveal some information about vehicles (e.g. predetermined route). In the cooperative paradigm, communication links between vehicles-to-UAVs are established and cooperative communication is introduced where vehicles sample the real-time data and upload them to the UAVs. This paradigm overcomes the limitations of the non-cooperative paradigm, however, maintaining vehicle privacy could be a challenge in this paradigm. In this work, the cooperative paradigm is considered.

There are a few significant works in UAV-assisted vehicular networks where high throughput, low latency or high coverage have been achieved as performance metrics as explained in Chapter 5. However, to the best of our knowledge, the impact of UAVs' and vehicle's mobility on the AoI has not been explored. Existing designs for UAV-assisted vehicular networks are insufficient to maintain fresh information. For instance, in [64], a non-trivial throughput-latency trade-off is revealed in the vehicular network when exploiting UAV mobility for throughput maximization. Due to the mobility of vehicles, each vehicle is served once during its entire navigation period and it then loses connectivity with the external networks. In spite of increasing the throughput, the information remains outdated, due to long-serving delays. Indeed, this impacts the applications that require frequent updates.

In this chapter, we study the AoI in an Internet of Vehicles setting and we consider a road segment where the current infrastructure is either temporarily unavailable or needs to be offloaded, and a set of UAVs is dispatched to collect/process multiple traffic streams from vehicles before the information they carry loses its value. Vehicles are sampling a stochastic process, which traffic is enqueued in a separate (per stream) queue. At every time-slot, considering different streams generated by the operating vehicles, the deployed UAVs must decide on choosing the best streams to be uploaded. Our goal is to develop an effective solution for finding the trajectories and scheduling policies of the deployed UAVs that keep the information fresh by minimizing the Expected Weighted Sum AoI (EWSA).

Related Work

There have been extensive studies recently done to address various challenges in UAV-assisted communication networks. Designing the optimal trajectory of the UAV is one of the significant research challenges. In particular, finding the optimal trajectory of UAVs depends on various design aspects (e.g., achievable-rate, latency, power, and flight duration). For example, in [64], the communication throughput in the vehicular network was maximized by optimizing the trajectory of a single-UAV and the scheduling of vehicles, however the authors did not consider Age-of-Information in their work. In [6], a UAV was dispatched to collect data from Internet of Things (IoT) devices under strict deadline constraints. The total number of IoT devices was maximized via jointly optimizing the UAV trajectory and radio resource allocation.

Recently, some works investigated the deployment of multiple-UAVs. In fact, the deployment of multiple-UAVs provides a higher performance compared to a single UAV. However, new challenges should be considered with using multiple-UAVs such as path planning, safety distances, coordination between UAVs, etc. For example, in [10], the minimum throughput of ground terminals was maximized by optimizing the trajectory for a set of UAVs, power control and scheduling of multiple ground terminals. In [72], a mathematical optimization model is elaborated which can provide the minimum number of UAVs and their trajectories for content delivery on a given highway segment without considering the AoI.

On the other hand, machine learning (ML) has received significant attention for solving challenging problems with UAVs. For instance, a Reinforcement Learning approach is utilized to govern the UAVs' trajectories with a set of actions (traveling distances) to provide coverage services [63]. The trajectories of UAVs with travel restrictions are assumed, where mobility of UAVs is constrained to finite distances. In [62], a Deep Reinforcement Learning approach is proposed to control a minimum number of UAVs' trajectories; the authors considered UAVs' trajectories without traveling restrictions (i.e. the UAV trajectories are continuous) while the coverage metric is explored without considering the AoI as a performance metric.

Optimizing the trajectory of multiple-UAVs while considering the AoI is another issue. In [22], the AoI was studied, where a single UAV was dispatched to collect data from multiple sensors, and

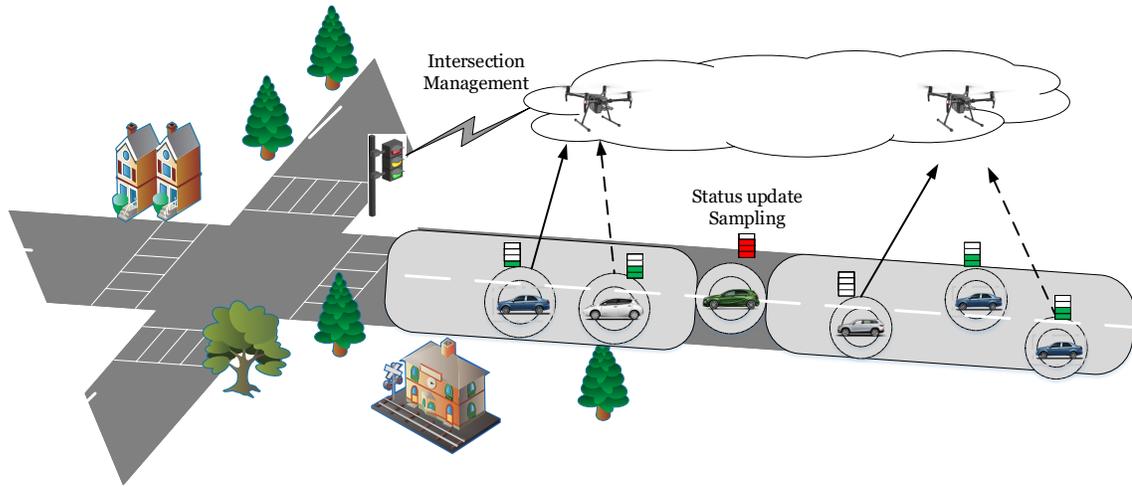


Fig. 6.1: Intelligent transportation systems with multiple-UAVs used for collecting and processing status-update packets from vehicles.

the maximum and average AoI was minimized by designing the UAV trajectory. In [23], a UAV as a mobile relay was considered to serve multiple IoT devices. The average peak AoI was minimized by optimizing the UAV's energy and its trajectory. The authors of [46] proposed dynamic programming approach to optimize the UAV trajectory and data acquisition problem where the AoI is considered. In [40], the authors exploited a ML based on RL algorithm to optimize a UAV trajectory for data collection mission to minimize the expired data packets. The authors in [47] utilized affinity propagation clustering algorithm to associate sensor nodes to collection points, then dynamic programming is exploited for optimizing the trajectory planning of a single UAV to address max-min AoI problem among all the ground sensors. In [39], the authors leveraged DRL algorithm to optimize the UAV trajectory and scheduling that minimize the Weighted Sum-AoI in static network.

In this work, different from the aforementioned works, we are interested in studying the AoI in UAV-assisted vehicular networks. Particularly, the objective of this research work is to leverage ML to find the solution that minimize the EWSA in intelligent transportation systems.

Motivation and Work Objectives

None of the previous work in the literature has addressed the problem of timely UAV-assisted data collection in high-mobility settings within a road segment. All the aforementioned works on

data collection relies on stationary users to formulate the AoI and obtain results. In reality, users could be mobile (e.g. vehicles) with random speeds. Nonetheless, prior work relies on optimization frameworks which require high computation resources in order to obtain results [76]. These frameworks are not scalable for dense networks with multiple vehicles and/or multiple applications.

UAV-assisted vehicular networks have been studied in [62, 63, 72], the main difference in their work is that UAVs are dispatched to provide coverage services for vehicles in highway segments without factoring AoI. In contrast, in this work, we consider a vehicular network in which vehicles periodically generate status-update packets that need to be collected/processed by multiple-UAVs. Unlike the previous works in a stationary environment, the mobility of vehicles and UAVs affects the transmission delay and the evolution of AoI over time, which in turn affects the results significantly.

The main contribution of this chapter is to provide a practical DRL algorithm for finding the trajectories of the deployed UAVs and scheduling of status-updates in order to minimize the EWSA. A Reinforcement Learning based framework is designed to transform the minimum EWSA optimization problem into a Markov Decision Process (MDP) by defining states, actions and rewards. In practical scenarios, the action space (traveling distance) of UAVs can be considered infinite to serve the existing and newly arriving vehicles. Hence, the use of DRL is a more appropriate framework to capture the infinite characteristics of the action space. In addition, the actions of UAVs hold interrelation with each other. Deep Q-Network (DQN) algorithm could be adopted to solve continuous control tasks through discretizing the action space. However, one of the major limitations of this technique is the curse of dimensionality [77]: the degree of freedom (DoF) may increase the size of action space exponentially. For instance, a 20 DoF (traveling distance in all directions) for two UAVs entails an action space of size, $2^{20} = 1048576$. To this end, we adapt one of the proposed Actor-Critic in the literature, namely, Deep Deterministic Policy Gradient (DDPG) [78], with Greedy scheduling policy to solve the stated problem.

Organization

The remainder of the chapter is organized as follows. Section 6.2 presents the communication scenario of the vehicle-to-UAVs and the problem formulation of trajectory design. Section 6.3 lays out a detailed presentation of the proposed DRL framework. The, simulation results are conducted

in section 6.4. Finally, we summarize the chapter in Section 6.5.

6.2 System Model

6.2.1 The Communication Scenario

We consider in Fig. 6.1 a road segment, without infrastructure or with infrastructure that needs to be offloaded, that has an unidirectional traffic¹ flow of vehicles. A set of \mathcal{M} UAVs with cardinality M is dispatched as flying data-collectors from navigating vehicles. We consider the system over multiple time frames. Each of these frames is further equally segmented into N time-slots of length δ_t , normalized to unity. At the beginning of every time-slot n , the deployed UAVs decide on the next location and each either remains idle or selects a vehicle to serve within its coverage over the wireless communication channel.

We consider a similar traffic model as in the works [79] and [80], where vehicles travel at random speeds. The vehicles' speed follows a truncated Gaussian distribution ranging from v_{min} to v_{max} [81]. We assume that arrival of vehicles into the road segment follows a Poisson distribution with density ρ_p Veh/Km [82]. During the considered time frames, vehicles entrance and exit the road segment, which induces variations in the number of vehicles in \mathcal{V}^n , where \mathcal{V}^n denotes the set of available vehicles in time-slot n . The width of a lane is typically small compared with the transmission range of vehicles and UAVs [88]. Therefore, we ignore the width of the road and model multiple lanes as one lane for each direction. The instantaneous position of i -th vehicle at time-slot n is denoted by $x_i^n, i \in \mathcal{V}^n$, while all the UAVs fly at $(w_m^n, 0, H)$ with a fixed² altitude H at time-slot n , where w_m^n is the x-axis position of UAV m at time-slot n .

We assume that the information-sampling behavior of each vehicle follows a *per time-slot sampling* policy, where each vehicle samples the status-update information at the beginning of each time-slot, such as shares LiDAR data. This policy is the special case of the periodic sampling policy with time period equals one. It is the most widely applied policy in literature [93], [94]. With Time Division Multiple Access (TDMA), one packet at most is scheduled for transmission by the

¹The same framework can be applied to the bi-directional traffic flow.

²In practice, the deployed UAVs can operate at various altitudes within the range allowed by the federal aviation regulations. However, we constrain their altitude to fixed height for simplicity.

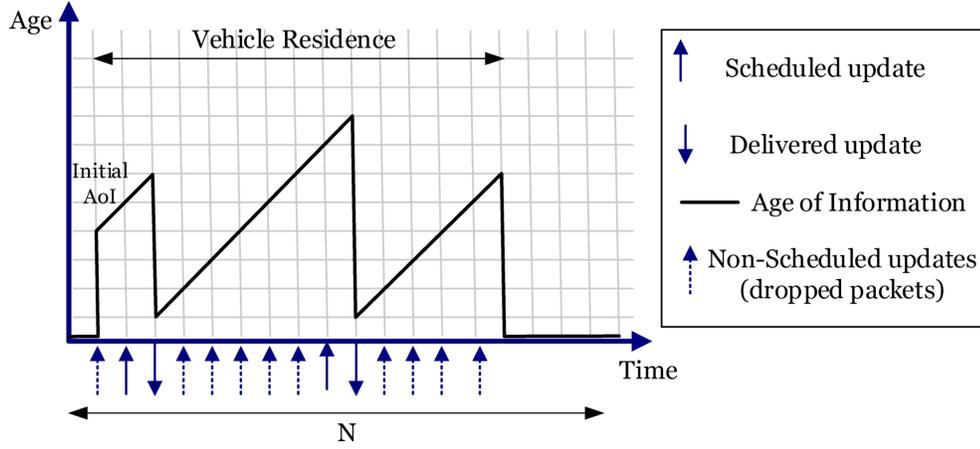


Fig. 6.2: The evolution of AoI associated with vehicle i with the initial AoI of four time-slots, $A_i(1) = 4$.

deployed UAVs from any vehicle within its coverage at any given time-slot. Thus, we assume that in each time-slot, each UAV only schedules at most one vehicle to transmit its status-update; therefore, the transmission scheduling should meet the constraint below:

$$\sum_{i=1}^{\mathcal{V}^n} \alpha_{i,m}^n \leq 1, \quad \forall n, m, \quad (6.1)$$

$$\sum_{m=1}^M \alpha_{i,m}^n \leq 1, \quad \forall i \in \mathcal{V}^n, n, \quad (6.2)$$

where $\alpha_{i,m}^n$ is a binary variable, which indicates that vehicle i is scheduled by the UAV m in time-slot n to transmit its status-update if $\alpha_{i,m}^n = 1$, and 0 otherwise. In other words, each deployed UAV only schedule at most one vehicle to transmit its status-update information (i.e., Eq. 6.1) and each vehicle is only served by at most one UAV (i.e., Eq. 6.2). In general, the wireless communication channels are composed of large-scale and small-scale fading [86]. However, in UAV-assisted vehicular networks, the channels are usually dominant by the Line-of-Sight component and the impact of small-scale fading is negligible [86] [25]. We consider uniform power of (P) for all the transmission at the vehicles. Thus, the channel gain from vehicle i to UAV m at time-slot n can be written as:

$$h_{i,m}^n = h_o \left(\sqrt{(x_i^n - w_m^n)^2 + H^2} \right)^{-2}, \quad \forall n, m, i, \quad (6.3)$$

where h_o is the median of the mean path gain at reference distance $d_0 = 1$ m, and B is the allocated bandwidth.

Thus, the rate expression for vehicle i to UAV m at time-slot n within its residence on the road segment is given by

$$r_{i,m}^n = \alpha_{i,m}^n B \log_2 \left(1 + \frac{Ph_{i,m}^n}{\sigma^2} \right), \quad (6.4)$$

where $\sigma^2 = BN_o$ with N_o denoting the power spectral density of the additive white Gaussian noise (AWGN) at the receivers. Therefore, depending on the scheduling and the distance between UAVs and vehicle, only a fraction of the transmitted bits can be successfully received.

Let s_i^{\min} be the minimum data size of status-update bits required for ensuring a reliable recovering/decoding. In this case, we consider an update to be successful if the number of bits, $s_{i,m}^n$, successfully received and decoded at UAV m in time-slot n , is strictly greater than s_i^{\min} . The minimum data size $s_{i,m}^n$ can be written as [25] [72] [6]:

$$s_{i,m}^n = \begin{cases} \delta_i r_{i,m}^n, & \text{if } a_i \leq n \leq d_i, \\ 0, & \text{otherwise,} \end{cases} \quad (6.5)$$

where a_i and d_i are the arrival and departure times of vehicle i to the road segment, respectively. Next, we will detail the definition of AoI in next section.

6.2.2 AoI Definition in Vehicular Networks

The concept of AoI describes the freshness of the information from the perspective of the receivers (i.e., UAVs). In order to track the AoI, we define age A_i^n as AoI of vehicle i in time-slot n . Whenever an update is successful (i.e., any UAV receives a minimum amount of bits from vehicle i), then AoI will reset to 1 since packets were generated at the beginning of time-slot n , otherwise,

the AoI increases by 1. The evolution of A_i^n of vehicle i can be written as³

$$A_i^{n+1} = \begin{cases} 0, & \text{if } a_i > n \text{ or } d_i < n, \\ 1, & \text{if } s_{i,m}^n \geq s_i^{\min}, \\ A_i^n + 1, & \text{otherwise.} \end{cases} \quad (6.6)$$

The characteristics of Eq. 6.6 deserves more elaboration. Clearly, the AoI of one vehicle is totally determined by the trajectory of the UAV, the mobility of the vehicle, the transmission scheduling and the total number of vehicles within the time of data collection mission. The EWSA of each vehicle i within the data-collection mission time N is captured by $\mathbb{E} \left[\sum_{n=1}^N A_i(n) \right]$, where \mathbb{E} is the expected operation with respect to the randomness in the transmission scheduling as well as the vehicles' arrival and departure times, the UAVs' trajectories and the mobility of the vehicles. In Fig. 6.2, we illustrate the evolution associated with one vehicle which arrives and departs the given road within the data collection mission time N while employing single packet queues⁴. Thus, in order to compute the AoI within the data-collection mission time using scheduling policy π , we use

$$EWSA = \mathbb{E} \left[\sum_{n=1}^N \sum_{i \in \gamma^n} \xi_i A_i(n) | \mathbf{A}_i(1) \right], \quad (6.7)$$

where ξ_i is a positive weight of vehicle i denoting the relative importance of the vehicle's application. $A_i(1)$ is the initial values of AoI for vehicle i . For notational simplicity, we assume that initial values are the same; and omit $A_i(1)$ henceforth. In particular, the weights ξ_i can be chosen according to the importance of the AoI for different processes. For instance, if the UAVs care most about the AoI for a specific process observed by the sensor of vehicle i , then we assign a high weight for the measurements of that sensor. Then vehicle i has a higher priority to be selected by the optimizer. Table 6.1 provides a summary of the notations used in the chapter.

The objective of this chapter aims at optimizing the UAVs' trajectories and the transmission

³AoI is evolves over time. However, we are interested in AoI of vehicles present within the road segment, therefore, for more tractable analysis, A_i^n is reduced to zero outside the given road segment.

⁴In this work, a single packet queue is employed as a queueing discipline such that the Head-of-Line packet is always the newest packet. In other words, the older status-update packet is replaced with the newly arrived packet (i.e., equivalent to Last-Input-First-Output queues).

Table 6.1: Optimization Problem Formulation

Parameters	Description
\mathcal{M}	Set of UAVs.
x_i^n	X-axis position of vehicle i in time-slot n .
H	UAVs' altitude.
N	Total number of time slots.
δ_t	Time slot duration.
\mathcal{V}^n	Available set of vehicles in time-slot n .
B	Channel bandwidth.
N_o	Noise power.
h_o	Median of the mean path gain at reference distance d_0 .
P	Vehicles transmission power.
A_i^n	AoI of vehicle i in time-slot n .
$r_{i,m}^n$	rate for vehicle i to UAV m at time-slot n .
s_n, a_n, r_n	State, action, reward at time-slot n .
s_i^{\min}	Minimum number of bits required for reliable decoding.
$s_{i,m}^n$	Number of bits that are reliably decoded from vehicle i to UAV m at time-slot n .
a_i, d_i	Arrival, departure times of vehicle i .
V_{\max}	UAVs' maximum speed.
d_{\min}	Minimum safety distance between UAVs.
q_m	Initial position of each UAV.
Variables	Description
w_m^n	X-axis position of UAV m at time-slot n .
$\alpha_{i,m}^n$	Indicates if vehicle i is scheduled from UAV m for transmission at time-slot n .

scheduling to minimize the EWSA within the road segment under the mobility of UAVs and vehicles constraints. Thus, our optimization problem is formulated as:

$$\begin{aligned}
 (\mathcal{O} \mathcal{P}): \quad & \min_{w_m^n, \alpha_{i,m}^n} \mathbb{E} \left[\sum_{n=1}^N \sum_{i \in \mathcal{V}^n} \xi_i A_i^n (w_m^n, \alpha_{i,m}^n) \right] \\
 \text{s.t.} \quad & \mathcal{C}1: \alpha_{i,m}^n \in \{0, 1\}, \forall m, i \in \mathcal{V}^n, \forall n, \\
 & \mathcal{C}2: \sum_{i=1}^{\mathcal{V}^n} \alpha_{i,m}^n \leq 1, \forall m, \forall n, \\
 & \mathcal{C}3: \sum_{m=1}^M \alpha_{i,m}^n \leq 1, \forall i \in \mathcal{V}^n, \forall n, \\
 & \mathcal{C}4: w_m^0 = q_m, \forall m, \\
 & \mathcal{C}5: |w_m^n - w_j^n| \geq d_{\min}, \forall m, m \neq j, n = 2, \dots, N-1,
 \end{aligned}$$

$$\mathcal{C}6: |w_m^{n+1} - w_m^n| \leq V_{\max} \delta_t, n = 1, \dots, N-1, \forall m.$$

Constraints $\mathcal{C}2$ and $\mathcal{C}3$ ensure that each UAV only schedules at most one vehicle and each vehicle is only scheduled by at most one UAV in one time-slot. $\mathcal{C}4$ indicates the initial position of each UAV located at position q_m . $\mathcal{C}5$ guarantees that the deployed UAVs are sufficiently separated at a minimum safety distance d_{\min} . Finally, $\mathcal{C}6$ limits the distance traveled by one UAV in one time slot based on its maximum speed.

We observe that $\mathcal{O} \mathcal{P}$ is a mixed integer non-linear program (MINLP), which is generally hard to be solved, due to the existence of the binary variables $\alpha_{i,m}^n$, and non-convex objective function. Even if the binary variables $\alpha_{i,m}^n$ is relaxed to take any value between 0 and 1, the relaxed version of $\mathcal{O} \mathcal{P}$ is, nevertheless, non-convex due to the trajectory variable w_m^n in the objective. To the best of our knowledge, there is no solver for solving $\mathcal{O} \mathcal{P}$ efficiently.

In addition, the trajectory planning is usually optimized via offline processing before dispatching the UAVs. Since there is no possible way to obtain a complete knowledge about newly arrived vehicles in advance before the UAV's flight. Moreover, in theory, some heuristic approaches may be able to relieve this problem, it would be, however, impractical to explore and adapt to all possible changes in the vehicular environment. That means we cannot properly solve $\mathcal{O} \mathcal{P}$. In order to solve this problem with an efficient solution, a Deep Reinforcement Learning algorithm is utilized to learn the environment and solve the proposed problem. In the next sections, the proposed algorithm for solving our formulated problem will be described.

6.3 The Proposed Deep Reinforcement Learning Approach

In this work, an artificial intelligence (AI) agent is deployed at the central unit at the ingress Ground Base-Station (BS) and interacts with the vehicular environment in a sequence of actions, observations, rewards and penalties. The AI agent observes the dynamic vehicular environment and steadily learns the trajectory and scheduling policy as well as manages the cooperation among the deployed UAVs. Therefore, a vehicle that cannot be served by one UAV might be served by other deployed UAVs. The vehicular environment can be observed by the deployed UAVs, which

is then sent to the AI agent, where the actor and critic networks decide the best control policy for the deployed UAVs. We will discuss this in more details in subsection 6.3.1. It is assumed that each vehicle periodically broadcasts on the control channel announcement beacon messages containing information identifying the type of services/applications, information about the speed, location [82], direction of travel, status-update packet size, and AoI initial value. The deployed UAVs, which act as mobile-relays between the vehicles and the control units, monitor the control channel of the network, aggregate the mobility features of vehicles and coordinate with the control unit where the AI agent resides. The deployed UAVs periodically consult the AI agent to decide their next travelling distance and direction as well as the scheduling decision. On the other hand, for non-control data exchanges between vehicles and UAVs and that between UAVs and the Ground BS, communication links are established on service channels (SCHs). Moreover, control channel and SCHs use different frequency bands, and therefore, the control information and data exchange can be executed simultaneously.

At each time-slot n , the AI agent decides an action for each UAV. The deployed UAVs will either travel along the road in a specific direction or hovering along with scheduling the transmission from the vehicles. It is important to understand, the real trajectory of UAVs can fly in arbitrary distances without any mobility constraint below the maximum speed. The agent then observes the dynamic changes in the vehicular environment and modifies the representation of the system state. The agent also receives a reward or penalty accordingly. In order to minimize the EWSA on the road, all UAVs should operate in a consistent, orderly and efficient way to provide the vehicles with acceptable service with minimum AoI. After each selected action (traveling and scheduling), each UAV receives a step reward which is a normalized indicator of how well the selected action accomplishes the previously-mentioned goals. The objective of the Actor-Critic algorithm is to construct an efficient action selection policy for each UAV that collects the status-update from vehicles along the road segment in order to achieve a minimum AoI. It is worth mentioning that the attained reward by each UAV depends on the entire previous sequence of actions and the observations from the vehicular environment. As such, the impact of the action may only be seen after several time steps. In the following, we first briefly review the Actor-Critic algorithm, a machine learning technique which is suitable for controlling autonomous machines such as UAVs. Then, we introduce our approach

using Actor-Critic algorithm for efficiently collecting vehicles' information with minimum AoI.

6.3.1 Deep Reinforcement Learning Background

Standard Reinforcement Learning is a branch of machine learning paradigm, which involves a multi-state decision process of a software agent (in a central unit in our case) while interacting with an environment in discrete decision epochs. In general, RL assumes that the system consists of multiple states S , where at each epoch n , the agent observes state $s_n \in S$, executes action a_n from a finite number of actions A according to an agent's policy π (i.e., the next UAVs' position and transmission scheduling) and receives a reward r_n , and moves to the next state s_{n+1} .

The goal of RL is to learn from the transition tuple $\langle s_n, a_n, r(s_n, a_n), s_{n+1} \rangle$, and find an optimal policy π^* that will maximize the discounted cumulative sum of all future rewards. Note that the policy $\pi = \{a_1, a_2, \dots, a_N\}$ defines which action a_n should be applied at state s_n . If we let $r(s_n, \pi(a_n))$ denote the reward obtained by choosing policy π , the cumulative discounted sum of all future rewards using policy π is given by:

$$R_\pi = \sum_{n=1}^N \lambda^{n-1} r(s_n, \pi(a_n)), \quad (6.9)$$

where $\lambda \in [0, 1)$ is a discount factor, which measures the weight given to the future rewards.

One of the widely used methods of RL algorithms, Q-learning, allows the agent to optimally act in an environment represented by a Markov decision process (MDP) [95]. Q-learning iteratively improves the state-action value function (also known as Q-function or Q-value), and by estimating the future reward if action a_n is taken, the agent presents the higher probability of going from state s_n to s_{n+1} using policy π . The Q-value function is usually stored in a table. However, Q-learning only works with a low-dimensional finite discrete action state space. For more information on RL and Q-learning, the reader is referred to [95]. DRL is a deep version of RL, where one (or multiple) deep neural networks (NNs) is used as the approximator of the action-value function $Q(\cdot)$. Deep Q-Network approach is one of the approaches of DRL, where a single neural network (NN) is trained

through minimizing a loss function L , as follows:

$$L(\theta^Q) = \mathbb{E}[T_n - Q(s_n, a_n | \theta^Q)], \quad (6.10)$$

where θ^Q are the function parameters (weights) of Deep NN; and T_n is a target value, which can be computed by

$$T_n = r_n + \lambda^{n-1} \max_{a_{n+1}} Q(s_{n+1}, a_{n+1}). \quad (6.11)$$

Deep Q-Network however tends to diverge with the non-linear function appropriator. Some techniques are utilized in order to avoid this divergence, namely; experience replay, fixed target network and reward normalization [87]. In experience replay, a random mini-batch of samples from the past experience is used during the training process to reduce the correlation between samples. In addition, in fixed target network, the same NNs' parameters are used to calculate the target function. Reward normalization techniques are used to limit the scale of the error derivatives and ensure the stability of the algorithm. However, it is unfeasible to apply both Q-learning and Deep Q-Network to continuous control because it is necessary to figure out the value for each action that maximizes the Q-function, which is quite difficult. DDPG with the assistance of experience replay, fixed target network and reward normalization techniques, was designed for continuous control actions. It uses an AC approach, that is, the use of two NNs namely actor and critic networks. The critic network is a Deep Q-Network, which is represented as $Q(s_n, a_n | \theta^Q)$. Therefore, the same loss function with different parameters is used for training the actor and critic networks, θ^Q and θ^π respectively. The actor network $\pi(s_n | \theta^\pi)$ is trained to obtain an action a_n for a given states s_n . The actor network is updated by applying the chain rule to the expected return from the starting distribution G with respect to the actor parameter θ^π [78]:

$$\nabla_{\theta^\pi} \mathbf{G} \approx \mathbb{E} [\nabla_a Q(s, a | \theta^Q)|_{s=s_n, a=\pi(s_n)} \cdot \nabla_{\theta^\pi} \pi(s | \theta^\pi)|_{s=s_n}]. \quad (6.12)$$

The weights of these networks are then updated by having them slowly track the learned networks $\theta' := \tau \theta + (1 - \tau) \theta'$, with $\tau \ll 1$. For more information on DDPG, the reader is referred to [78]. The next subsection presents the representation of the system state as well as the rewards

and penalties associated with the agent's actions.

6.3.2 Input From the Environment

At the beginning of the data collection mission, the agent observes the vehicular network environment that defines the states of the system, collects all the parameters associated with the set of in-range vehicles, and executes an action for each UAV at time-slot n . It is noteworthy that the number of vehicles present within the considered road segment is variable. Therefore, in order to avoid the complexity of having a variable vector size, we assume the maximum expected number of vehicles within the coverage of road segment is U . The number of vehicles present within the road segment follows the Poisson distribution [96]. The input of UAVs from the vehicular environment at time-slot n is:

- V^n : the number of vehicles residing within the considered road segment, at time-slot n .
- x_i^n : a vector of size U containing the instantaneous position of all vehicle at time-slot n .
- w_m^n : a vector of size M containing the ground level position of each UAV, at time-slot n .
- C_i^n : a vector of size U , containing the current AoI of each vehicle $i \in (1, 2, \dots, U)$, at time-slot n .

Each UAV fully observes the current vehicular network environment and updates the central unit which is able to realize the representation of the system state at each time-slot.

6.3.3 Actions and Expected Rewards

At each step-slot n , each UAV m executes an action a_m^n which consists of three parts: 1) $b_m^n \in [0, 1, 2, \dots, U]$: the scheduling decision for the UAV m in time-slot n . If $b_m^n = z_m$, UAV m selects vehicle index $z_m \in U$ to upload its status-update packets at time-slot n , resets the vehicle's AoI to 1 iff (if and only if) the update is successful and increases the AoI for non-served vehicles, $\forall i \in U, i \neq z_m$, by one time-slot older. 2) $d_m^n \in (0, d_{max}]$: the traveling distance, depending on its current state, where $0 \leq d_m^n \leq d_{max}$ and d_{max} is the maximum traveling distance within a time-slot. 3) Φ_m^n : the flying direction for each deployed UAV m at time-slot n . The UAVs' speed and trajectory can be

arbitrary in our considered problem, which is highly challenging to address. However, under our specific network scenarios, the vehicles can be considered navigating in 1D trajectory (in only two directions) [63] Φ_m^n , (left and right). To this end, the action $a^n = [b_1^n, \dots, b_M^n; d_1^n, \dots, d_M^n; \Phi_1^n, \dots, \Phi_M^n]$, has a dimension of $3M$. Hence, at time-slot n , each UAV chooses its trajectory action (distance and direction) and then decides which vehicle to transmit its status update and the DRL agent obtains an immediate reward from the vehicular network accordingly. As a result, the immediate reward r_n total sum of following quantities:

- (1) Penalty from the network when the UAVs collect status-updates from vehicles with high AoI: the value of this penalty is proportional to the EWSA of all current and newly arrived vehicles. As a result, the network learns to maximize this reward by optimizing scheduling decisions and trajectories of the deployed UAVs in order to collect the old status-update (i.e., highest AoI) from the current and newly arrived vehicles.
- (2) Penalty from the network if the flying distance between the deployed UAVs violates the safety distance: in order to maximize the rewards, AI agent learns how to maintain the safety distance between UAVs to avoid this penalty.
- (3) Penalty from the network if the deployed UAV flies outside the given road segment: the AI agent learns how to continue the flying on the given road segment.
- (4) Penalty from the network if the deployed UAVs schedule the same vehicle at the same time-slot: the deployed UAVs are encouraged to minimize this penalty by scheduling different vehicles at each time-slot to ensure each vehicle is only scheduled by at most one UAV.

Obviously, we are dealing with a discrete-continuous hybrid action space since each UAV executes infinite actions (trajectory) while decide which vehicle to upload its status-update packets. Thus, DRL techniques is necessary to be adapted to solve the stated problem.

6.3.4 Solution Algorithm

Our main target is to obtain a policy that governs the trajectories of a set of UAVs and scheduling decision at each time-slot to minimize the EWSA. The considered scenario is discrete-continuous

hybrid action space problem. To handle this problem, the existing literature either approximate the hybrid space by discretization (i.e., discretization the trajectory of UAVs) and use DQN, or relax it into a continuous set (i.e., relax the scheduling decisions) and use DDPG. However, such approaches suffer from a number of limitations: for the continuous part (UAVs trajectories), finding a good approximation usually requires a huge number of discrete action since the degree of freedom (DoF) increases the size of action space exponentially; for the discrete part (scheduling decisions) of hybrid action, relaxing them into a continuous set might provide inaccurate results since relaxing the scheduling decisions will allow multiple vehicles to transmit their status-update at the same time. Parametrized deep Q-network (P-DQN) framework is proposed in [97] to solve the hybrid action space without approximation or relaxation. However P-DQN will increase the complexity of the problem since P-DQN can be viewed as an integration of DQN and DDPG. Even if we discretize the trajectory of UAVs into Q discrete values and combine the two actions into one single action, a total of $(Q \times U)$ possible actions need to be considered. This could increase the learning difficulty of the agent. To reduce the complexity and maintain the continuous action space without approximating the trajectory of the UAVs, we use the UAV trajectory as the main control objective for two reasons. 1) the effective transmission scheduling based on Greedy policy is easy to obtain. 2) the trajectories of the UAVs are continuous, and DDPG is suitable to handle these actions without approximations. Therefore, the DRL agent will learn to optimize the trajectory of the UAVs to maximize the expected future reward while adopting the effective scheduling policy.

To avoid the complexity of having multiple neural networks, our proposed solution update the discrete-action and continuous-action policies separately by combining an efficient transmission scheduling policy based on *Greedy policy* and DDPG. Therefore, the action space can be simplified to $a^n = [d_1^n, \dots, d_M^n; \Phi_1^n, \dots, \Phi_M^n]$, which has a dimension of $2M$. Hence, at time-slot n , the DRL agent decides the trajectory action for each UAV and then select the vehicle to upload its status-update packets with high AoI.

In the following, we discuss the Greedy scheduling policy that minimizes EWSA within the communication coverage of UAVs. Then, for the considered problem networks, we propose DDPG that learn to optimize the trajectory of the UAVs while adopting Greedy scheduling policy

Algorithm 9: Proposed Solution: DRL to obtain UAVs' trajectories policy to minimize AoI.

```

1 Input: Discount factor, learning rate for actor and critic network, buffer size, mini-patch
   size, UAV energy parameters, penalties;
2 Output: The trajectories of UAVs.
3 Initialize replay buffer  $Z$ .
4 Randomly initialize critic network  $Q(s, a | \theta^Q)$  and actor network  $\pi(s | \theta^\pi)$  with weights  $\theta^Q$ 
   and  $\theta^\pi$ ;
5 Initialize target networks  $Q'$  and  $\pi'$  with weights  $\theta^{Q'} \leftarrow \theta^Q, \theta^{\pi'} \leftarrow \theta^\pi$ .
6 for  $episode = 1, P$  do
7   Collect network characteristics to realize state  $s_1$ .
8   for  $n \in N$  do
9     Observe:  $V^n, x_i^n, w_m^n$ , and  $C_i^n$ ,
10    Select: Action  $a_n^m = \pi(s_n)$ , and add a random noise that decays over time
11    Evaluate: obtain the reward  $r_n$  and  $s_{n+1}$ ,
12    for UAV  $m := 1, \dots, M$  do
13      if ( $UAV\ m\ (flies\ outside\ the\ border)$ ) Or ( $|w_m^n - w_j^n| \geq d_{min}, \forall m, m \neq j$ ) then
14         $r_n = r_n - P$ .
15        Cancel the movement of UAV  $m$  and update  $s_{n+1}$ .
16      else
17         $A$ 
18        pply Algorithm 7 for Clustering.
19        Greedy selects vehicle  $i'$  to upload its status-update packets, where
            $i' = \arg \max_i A_i^n, \forall i \in Q_m$ , and update  $r_n$ .
20    Store transition  $(s_n, a_n, r_n, s_{n+1})$  in  $Z$ .
21    Sample random mini-batch of transitions  $(s_n, a_n, r_n, s_{n+1})$  of size  $H$  samples from  $Z$ .
22     $T_n := r_n + \lambda Q'(s_{n+1}, \pi'(s_{n+1} | \theta^{\pi'}) | \theta^{Q'})$ ;
23    Update weights  $\theta^\pi$  of  $Q(\cdot)$  by minimizing the loss:
            $L(\theta^Q) = \frac{1}{H} \sum_{n=1}^H (T_n - Q(s_n, a_n))^2$ 
24    Update the weights  $\theta^\pi$  of  $\pi(\cdot)$  using:
            $\nabla_{\theta^\pi} G \approx \frac{1}{H} \sum_{l=1}^H \nabla_a Q(s, a | \theta^Q) \Big|_{\substack{s=s_l \\ a=\pi(s_l)}} \nabla_{\theta^\pi} \pi(s | \theta^\pi) \Big|_{s=s_l}$ ;
25    Update the corresponding target networks:
            $\theta^{Q'} := \tau \theta^Q + (1 - \tau) \theta^{Q'}$ ;
26     $\theta^{\pi'} := \tau \theta^\pi + (1 - \tau) \theta^{\pi'}$ ;
27
28

```

Greedy Scheduling Policy

Theorem 1 shows the optimality of Greedy scheduling Policy that minimizes the EWSA among the class of admissible policies. In this study, a special scenario is considered to show the optimality of scheduling within the communication range of a UAV.

Algorithm 10: Clustering Algorithm.

- 1 **Input:** The positions of UAVs w_m^n , $\forall m \in M$ and vehicles x_i^n , $\forall i \in V^n$;
 - 2 **Output:** Cluster vectors for each UAV Q_m , $\forall m \in M$;
 - 3 Fix the number of clusters M ; Initialize the cluster vectors Q_m ;
 - 4 **for** Vehicle $i := 1, \dots, V^n$ **do**
 - 5 $m \leftarrow \arg \min_{m \in M} |w_m^n - x_i^n|$;
 - 6 **if** $|w_m^n - x_i^n| \leq \left(\left(2^{\frac{s_i^{\min}}{B}} - 1 \right)^{-1} \frac{Ph_o}{\sigma^2} \right) - H^2$ **then**
 - 7 $Q_m \leftarrow Q_m \cup x_i^n$; //Assign vehicles to cluster vector.
-

Theorem 1 (Optimal scheduling Policy to minimize the EWSA): Consider a set of vehicles are navigating within the communication range of a single⁵ UAV such that all status-updates could be successfully delivered to the UAV, i.e., $s_{i,m}^n \geq s_i^{\min}, \forall i$ and $\xi_i = \xi$, where s_i^{\min} denotes the minimum data size of status-update bits required for ensuring a reliable decoding and $s_{i,m}^n$ is the total number of successfully received bits. The Greedy scheduling Policy within the UAV communication range attains the optimum scheduling policy that minimizes the EWSA.

Proof: The concept of stochastic dominance⁶ is used to prove the optimality of the Greedy scheduling policy within the UAV cluster. Stochastic dominance is utilized for comparing uncertain alternatives, usually applied on decision analysis [1]. In other words, the scheduling decision of Greedy policy G is compared with any arbitrary scheduling⁷ policy χ . Let $\varphi(\chi, n)$ and $\varphi(G, n)$ be random variables (RVs) that represent the sum of AoI for all vehicles when the arbitrary scheduling policy χ and Greedy policy G are employed at time-slot n , respectively. We define the stochastic process $\varphi(\chi)$ associated with all RVs $\varphi(\chi, n), \forall n$ and similarly for $\varphi(G)$ with all RVs $\varphi(G, n), \forall n$. To complete the proof, we first define the stochastic dominance.

Definition (Stochastic Dominance): $\varphi(\chi)$ is said to stochastically dominate $\varphi(G)$ on the set of functions \mathcal{F} , or written as $\varphi(\chi) \geq_{SD} \varphi(G)$, if and only if $P\{\mathcal{F}(\varphi(G)) > z\} \leq P\{\mathcal{F}(\varphi(\chi)) > z\}, \forall z$, for all functions \mathcal{F} , where \mathcal{F} is a set of increasing functions.

Stochastic dominance \geq_{SD} implies dominance of moments: $\mathbb{E} \left[\sum_{n=1}^N \varphi(\chi, n) \right] \geq \mathbb{E} \left[\sum_{n=1}^N \varphi(G, n) \right]$ for all arbitrary scheduling policies. Recall that our main objective is to

⁵Similar analysis is valid for other UAVs.

⁶The concept of stochastic is widely used in the context of game theory in which one decision can be ranked as superior to another decision.

⁷For the sake of simplicity and without loss of optimality, χ is assumed to be work-conserving. i.e., the UAV keeps the resources busy by scheduling one vehicle at every slot.

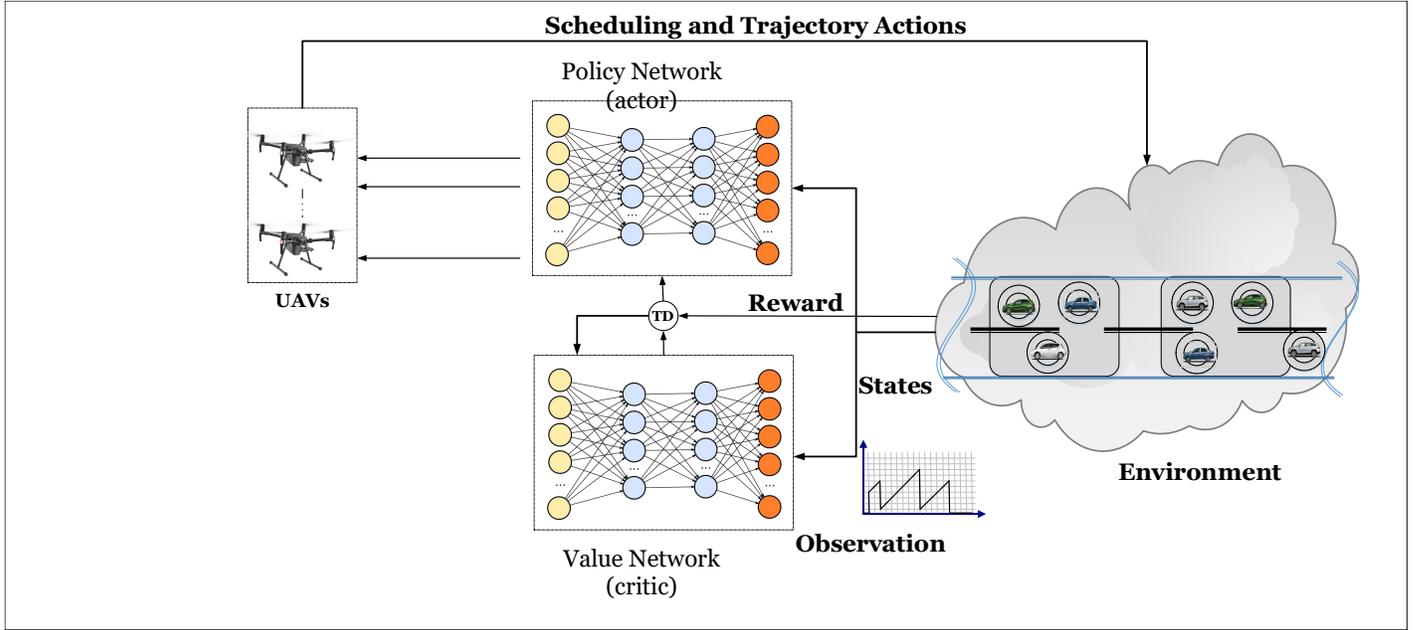


Fig. 6.3: DRL-based proposed approach to obtain the control policy.

minimize the Expected Weighted Sum AoI with equal weights and $s_{i,m}^n \geq s_i^{\min}, \forall i$ are assumed. Thus, the optimality of the Greedy scheduling policy is established if the condition $\mathbb{E} \left[\sum_{n=1}^N \varphi(\chi, n) \right] \geq \mathbb{E} \left[\sum_{n=1}^N \varphi(G, n) \right]$, for all arbitrary scheduling policies, is satisfied. To prove that, it is sufficient to confirm that $\varphi(\chi) \geq_{SD} \varphi(G)$.

According to [98–100], it is sufficient to use the following Lemma to prove stochastic order relations without explicit computation of distributions:

Lemma: Two stochastic processes $\varphi(\chi)$ and $\varphi(G)$ satisfy $\varphi(\chi) \geq_{SD} \varphi(G)$ if there exist two stochastic processes $\widehat{\varphi}(\chi)$ and $\widehat{\varphi}(G)$ on a common probability space; such that $\widehat{\varphi}(\chi)$ and $\varphi(\chi)$ have the same probability distribution as well as for $\widehat{\varphi}(G)$ and $\varphi(G)$; and $\widehat{\varphi}(\chi, n) \geq \widehat{\varphi}(G, n), \forall n$.

Let $\widehat{\varphi}(\chi)$ and $\varphi(\chi)$ be identical with common distribution functions. Next, a new process $\widehat{\varphi}(G)$ created based on the Greedy policy to share the same probability space of $\widehat{\varphi}(\chi)$. The arbitrary and the greedy policies share a common probability space since both decide on UAVs' scheduling. Taking into account that all vehicles have the same weights and satisfy the transmission condition, $s_{i,m}^n \geq s_i^{\min}, \forall i$, thus, $\widehat{\varphi}(G)$ and $\widehat{\varphi}(\chi)$ are stochastically coupled. Correspondingly, the probability distribution for $\widehat{\varphi}(G)$ and $\varphi(G)$ are obtained to be the same as vehicles with the same AoI will be selected. To prove that $\widehat{\varphi}(\chi, n) \geq \widehat{\varphi}(G, n), \forall n$, the characterization of the evolution of $\widehat{\varphi}(\chi)$

and $\widehat{\varphi(G)}$ for each time-slot n needs to be studied. Since all vehicles, within the communication range of the UAV, are able to upload their status-update successfully at any time-slot, intuitively, the Greedy scheduling Policy G always achieves the lowest EWSA. This is explained by the fact that the UAV schedules the vehicle with the highest AoI in each time-slot according to the Greedy policy. Therefore, the Greedy Scheduling Policy in every time-slot yields the minimum value, thus, $\widehat{\varphi(\chi, n)} \geq \widehat{\varphi(G, n)}, \forall n$. **The proof is completed.**

Proposed DDPG Algorithm

The proposed actor-critic algorithm is shown in Fig. 6.3, which is consisted on the vehicular network, the observation including the network dynamics, actor and critic networks, AoI costs, and a temporal difference error. As mentioned before, since our problem is infinite action space (trajectories of UAVs), we adapt the DDPG to solve our problem. The proposed DRL algorithm to obtain UAVs' trajectories is presented in Algorithm 1. The proposed algorithm works as follows.

In the first part, after defining the input and output of the algorithm (Lines 1-2), the proposed algorithm randomly initializes the replay buffer of size Z , the weights parameters for the actor-network θ^π and critic network θ^Q (Lines 3-4) and the initial position of the UAVs is set to location q_m to enforce constraint $\mathcal{C}4$. Further, as mentioned in Subsection (6.3.1), we create the target networks $\pi'(\cdot)$ and $Q'(\cdot)$ to enhance the training stability, where the target, critic and actor networks have the same structures. The target network weight parameters $\pi'(\cdot)$ and $Q'(\cdot)$ are initialized (Line 5), where at later steps (Lines 12- 23), those parameters are slowly updated according to the control parameter $\tau = 0.001$ in order to enhance the stability.

The exploration phase, reward, and penalties are explained in the second part (Lines 6-27). During the exploration phase, the algorithm obtains a trajectory action from the current actor-network θ^π bounded with the maximum velocity of the UAVs, ω_{max} (to enforce constraint $\mathcal{C}6$), and then a random noise is added that decays over time with a rate of 0.9995, where the random noise is generated from a uniform distribution with a zero mean and a variance of 1. During the training phase, the proposed algorithm guide the AI agent to avoid actions that violate the road border (i.e., flies outside the given road segment) by applying a specific penalty to the reward (Lines 13-15), where, a penalty p is deducted from the overall reward, and the corresponding trajectory action of

the UAV m is canceled. Likewise, the proposed algorithm trains the agent to maintain the safety distance between the deployed UAVs to avoid the penalty p . Therefore, Lines 13-15 are proposed to enforce constraint $\mathcal{C}5$ and avoid actions that violate the road border.

In order to ensure that each vehicle is only scheduled by at most one UAV, a modified K-Means cluster algorithm is adopted. At each time-slot n , the AI agent divides the set of vehicles V^n into M disjoint groups based on their geographic locations in which nearby vehicles are grouped together based on the euclidean distance with respect to the euclidean location of the deployed UAVs located at w_m^n . Note that the vehicles' geographic locations vary with respect to the slotted time length. Therefore, the AI agent clusters vehicles every time-slot. In order to obtain a reliable communication with the minimum status-update size, s_i^{\min} , the euclidean distance between the deployed UAVs and vehicles within the cluster should not exceed a given threshold $\left(\left(\left(2^{\frac{s_i^{\min}}{B}} - 1\right)^{-1} \frac{Ph_o}{\sigma^2}\right) - H^2\right)^{0.5}$. Therefore, a vehicle that not achieve this condition will be excluded from the UAVs' cluster. Vehicles' clustering algorithm is explained in Algorithm 2. Greedy algorithm is applied for each cluster where the AI-agent selects vehicle i' to upload its status-update packets, where $i' = \arg \max_i A_i^n, \forall i \in Q_m$. In this step the algorithm assigns a binary variable "1" to the selected vehicle and assigns "0" for other vehicles. Therefore, Lines 17-18 are proposed to enforce constraints $\mathcal{C}1$, $\mathcal{C}2$, and $\mathcal{C}3$.

In the last part, the weights and parameters of the neural network (Lines 21-31) are updated according to the DDPG algorithm. Firstly, the collected samples including (s_n, a_n, r_n, s_{n+1}) are stored in the replay buffer of size Z after each executed action, and then a random mini-batch of size H is sampled from the buffer Z to update the actor and critic networks. As explained in Subsection (6.3.1), the weights parameters of the critic network are updated to minimize (6.10), while the actor-network weights parameters are updated according to (6.12).

6.3.5 Complexity Analysis

In this subsection, the complexity analysis is discussed. After adequate training, the DRL agent observes the environment where the states of MDP are defined as input for DRL algorithm in Section III-B. The agent utilizes its trained actor network $\pi(s|\theta^\pi)$ to carry out an action a_m^n which represents a traveling distance and a direction. Based on [75], the total computational complexity for the fully

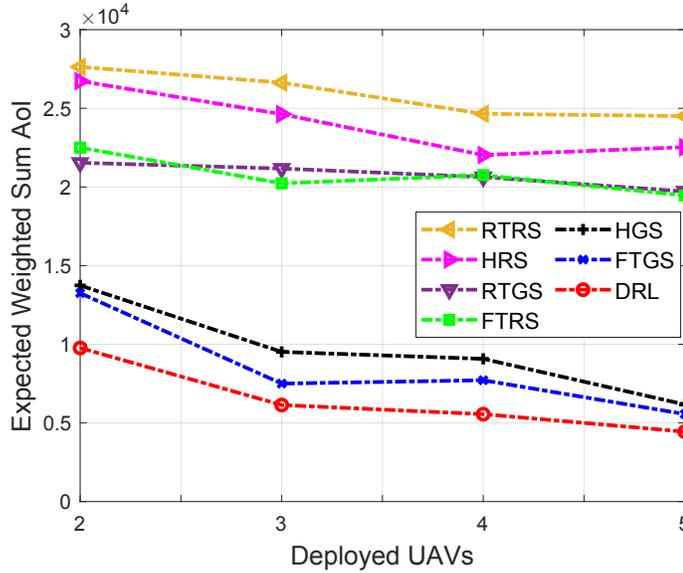


Fig. 6.4: Impact of number of UAVs and comparisons.

connected layers can be expressed as the number of multiplications: $O(\sum_{p=1}^P n_p \cdot n_{p-1})$, where n_p is the number of neural units in fully-connected layer p .

6.4 Simulation Results and Discussion

We use simulations to evaluate the performance of the proposed algorithm. Firstly, we describe the simulation setup and then present the benchmark schemes followed by results and analysis.

Simulation Setup

The simulation parameters which are used are outlined in Table 6.2. The simulation parameters should be carefully defined for an accurate representation of a real road scenario. A road segment of length 5km is considered in our simulation, on which multiple-UAVs are deployed to collect or process multiple traffic streams from vehicles where a Free-Flow traffic model is adopted. For sake of illustration, we assume that the application weight for all vehicles is identical and normalized to unity and all vehicles can communicate with the deployed UAV at different rates according to channel conditions within the given road segment. The results are collected after the training phase (500 000 samples), each sample corresponds to a snapshot of the vehicular network at a particular time-slot. All simulations are run for 2-layer fully connected neural network for each network

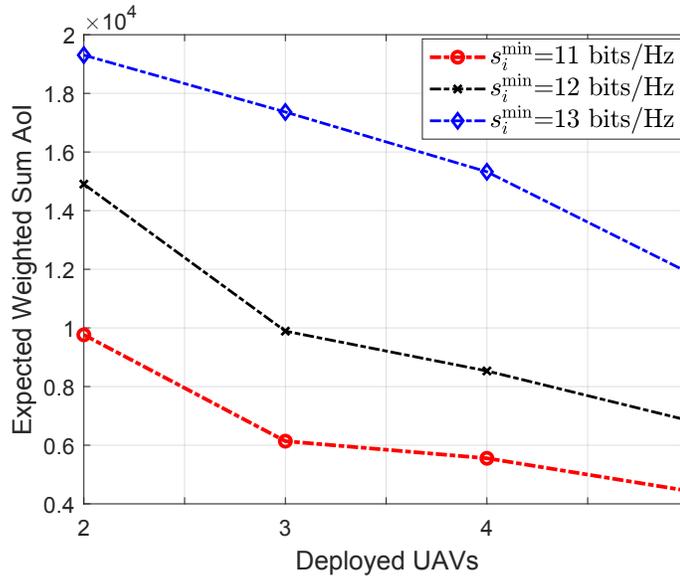


Fig. 6.5: Impact of status-update size.

(i.e., the actor and critic networks), which includes 20 and 80 neurons in the first and second layers respectively. Various combinations of Layers and Neurons are tested in the process of debugging, we found the best architecture is to use two-layers fully connected network while changing this architecture was often counterproductive. The deployment of the two-layer fully connected network is widely used in the literature while focusing on DRL Algorithm [75]. The rectified linear unit (ReLU) function is utilized for activation for both networks, while hyperbolic tangent (tanh) is used in the last layer for that actor-network to bound the traveling distance of the deployed UAVs according to the maximum traveling distance constraint. The generated samples are used to train the deep neural network by utilizing Tensor Processing Unit (TPU) to realize a policy for the deployed UAVs. After establishing the trajectory determined by the proposed algorithm, another set of mobility traces was used to test the performance of the proposed algorithm.

Benchmark Schemes

In fact, there is no existing approach that aims for solving a similar problem in vehicular networks, to the best of our knowledge; thus, for the sake of comparison, we consider six other baseline policies:

Table 6.2: Simulation Parameters in UAV-assisted Vehicular Networks

Parameter	Value
Minimum Vehicle Speed, v_{min}	22m/s
Maximum Vehicle Speed, v_{max}	33m/s
Velocity Mean	27.5m/s
Velocity Variance	4.5m/s
UAV Max Speed, ω_{max}	50m/s
Size of time frame,	240s
Total number of time slots per frame, N	240
Time slot duration, δ_t	1s
Road Segment of Length d	5Km
Buffer Size	10000
Patch Size	120
Activation Functions	ReLU and Tanh
Number of Layers	2
Learning Rate for Actor	0.001
Learning Rate for Critic	0.002
Reward Discount	0.8
Action Variation	50
Decay the Action Randomness	0.995
Soft Replacement Value	0.01
Optimizer Technique	Adam
UAV Altitude, H	100m
Channel Gain, γ_0	-50 dB
Noise Power, σ^2	-110dBm

- *Random Trajectory with Random Scheduling (RTRS)*: In the RTRS approach, at each time-slot, each deployed UAV randomly changes its direction within $[0, 2\pi]$ and a traveling distance within $[0, V_{max}]$. The UAV also randomly selects a vehicle on the road segment to upload its status-update packets. Meanwhile, if one or more UAV(s) violate the safety distance and fly beyond the road segment border (or distance) between UAVs, then all deployed UAVs abandon this action.
- *Random Trajectory with Greedy Scheduling (RTGS)*: In the RTGS approach, at each time-slot, each deployed UAV randomly changes its direction and traveling distance. It also selects a vehicle to upload its status-update packets according to Greedy transmission policy. Similar to

the RTRS approach, the selected trajectory action resulting in the violation of safety distance and/or road segment will be refused.

- *Fixed Trajectory with Greedy Scheduling (FTGS)*: In the FTGS approach, at each time-slot, the deployed UAV travels within a fixed traveling distance in the same direction as the vehicles (i.e. 1st UAV at 10m/s, 2nd UAV at 20m/s, and so on). It simultaneously collects the status-update packets according to the Greedy transmission policy. Meanwhile, for each UAV that reaches the end of the road segment, a new UAV with the same velocity will be dispatched.
- *Fixed Trajectory with Random Scheduling (FTRS)*: In the FTGS approach, at each time-slot, the deployed UAVs with fixed velocities randomly collect the status-update packets from vehicles on the road segment. Meanwhile, same as the FTGS approach, a new UAV with the same velocity will be dispatched for each UAV that reaches the end of the road segment.
- *Hovering with Random Scheduling (HRS)*: In the HRS, the deployed UAVs hover at a fixed position, every 1km, to randomly collect the status-update packets.
- *Hovering with Greedy Scheduling (HGS)*: In the HGS, the hovering UAVs collect the status-update packets according to the Greedy transmission policy.

Results and Analysis

We start by first investigating the impact of the number of UAVs on the EWSA, where the size of the status-update for all vehicles is set as $s_i^{min} = 11\text{bits/Hz}$ and the vehicular density is 6 Veh/km. It can be seen in Fig. 6.4, the DRL approach scheme achieved the lowest EWSA compared to other approaches. It is because the deployed UAVs have a more feasible trajectory to adapt as velocity and direction change; therefore, the UAVs can collect more status-updates. We also observe that optimizing the UAVs trajectories becomes more crucial with a low number of UAVs to attain better communication and increase the service amount. For instance, to achieve the total of EWSA of 7×10^3 time-slots, the required number of UAVs for our proposed solution is almost 3UAVs. The same performance can be achieved by increasing the number of UAVs for HGS and FTGS approaches, where 5 UAVs become necessary to fulfill the requirements. On the other hand, by

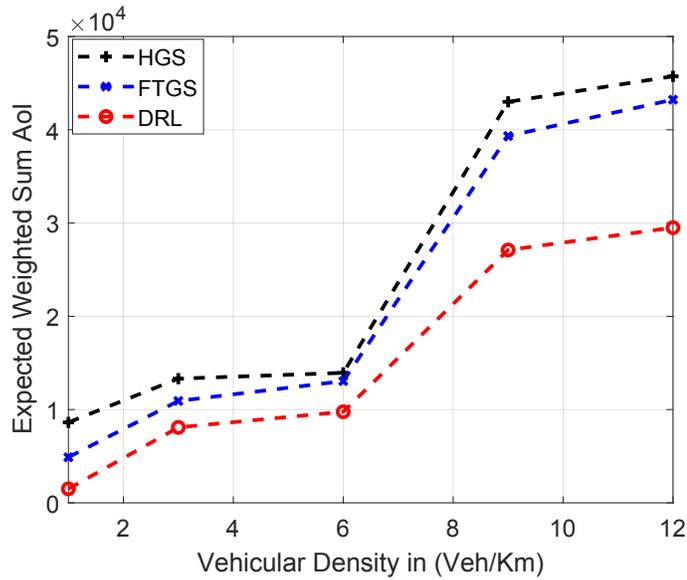


Fig. 6.6: Impact of vehicular density.

increasing the number of UAVs beyond 5 UAVs, the overall enhancement slightly decreases. This is because optimizing the trajectory becomes less significant to minimize the EWSA, where the trajectories of DRL, FTGS and HGS tend to be the same.

Clearly, 2 UAVs may not be able to meet the vehicles's application requirements (i.e., service demand) for all status-update sizes. As a result, we study the impact of status-update size, s_i^{min} , on the proposed DRL approach over a different number of UAVs. As shown in Fig. 6.5, at lower s_i^{min} (e.g., 11bits/Hz), 2 UAVs are sufficient to attain a total EWSA of 1×10^4 time-slots. Slightly increasing the status-update size (e.g., 12bits/Hz), 2 UAVs are no longer sufficient, and as the figure shows 3 UAVs will need to be deployed to attain the same performance. Further increasing the update size, more UAVs will need to be deployed. The findings here show that based on the target application running on the vehicles which is generating time sensitive data, attaining certain target AoI performance may require less or more UAVs by the operator.

Next, we study the impact of vehicular density on the proposed DRL approach compared to FTGS and HGS approaches, where 2 UAVs are deployed to collect the status-update of size $s_i^{min} = 11$ bits/Hz from all vehicles. As shown in Fig. 6.6, at lower vehicular density, our proposed algorithm is able to minimize the EWSA, since a low vehicular density implies that the vehicular traffic is very light or, alternatively, the vehicle inter arrival time is large and thus each vehicle enjoys more

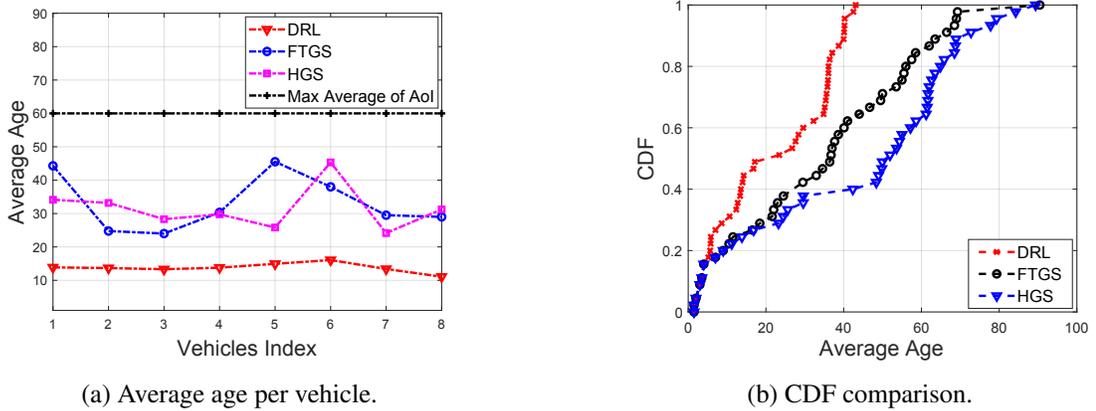


Fig. 6.7: Average age comparison with different policies.

service. In contrast, as the vehicular density increases, the vehicular traffic flow increases, the EWSA increases as expected since more UAVs are required to fulfill the vehicles' requirements. Also as seen in the figure, the performance gap between the policies increases as the number of vehicular density increases, and reduces for low density which demonstrates the importance of optimizing the trajectories of UAVs with a high number of vehicles. We can thus observe from the figure that DRL significantly reduces the EWSA compared to FTGS and HGS approaches.

The average age is another performance metric we study. Fig. 6.7(a), depicts this metric for a set of vehicles, where 2 UAVs are deployed to collect the status-update of size $s_i^{min} = 11 \text{ bits/Hz}$ from all vehicles and the vehicular density is set as 6 Veh/Km in a 2 minute period. The average age of vehicle i within the mission time N is captured by $\frac{1}{d_i - a_i} \left[\sum_{n=1}^N A_i(n) \right]$. Clearly, the proposed policy minimizes the average AoI in the system among the considered policies. In Fig. 6.7(b), the cumulative distribution function (CDF) of the average age is compared for different policies. The results are obtained through 10^3 independent Monte-Carlo trials on the road segment with the vehicular density is 12 Veh/km and a mission time of 4mins. It can be seen in the figure, the proposed algorithm consistently outperforms other approaches such as FTGS and HGS. For instance, the median AoI achieved by DRL is about 18 time-slots while this value can reach up to 38 time-slots and 50 time-slots in the FTGS and HGS approaches, respectively. This is because the former provides more flexibility for the deployed UAVs to learn the vehicular environment and its dynamics as well as to adapt their trajectories while scheduling transmission. In contrast, in

fixed trajectories and hovering approaches, the fixed velocity and hovering do not have a significant impact on the upload schedule. This justifies the robustness of the proposed algorithm in terms of minimizing average AoI.

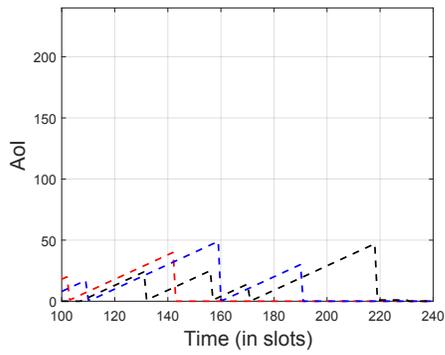
To better understand the impact of trajectories and scheduling transmission policy on AoI evolution over time in the environment of the vehicular network, we next show the AoI evolution over time for all approaches. For sake of fair comparison, we simulate a scenario that takes into account a snapshot in a 4 minute period with 2 UAVs for the same three randomly selected vehicles. As shown in Fig. 6.8, the AoI evolution can be drastically different for the different policies. It is observed that by utilizing our DRL approach, the AoI of the three vehicles is much smaller than that of the benchmark policies. This is due to the fact that, as explained above, the DRL agent learns the dynamics of the vehicular network and optimizes the deployed UAVs' trajectories by traveling back-and-forth to fly closer to a vehicle with the highest value of AoI in order to collect its status-update packets. Obviously, the number of lost status-update packets increases among the trajectory baselines of the UAVs and this is due to the fact that the deployed UAVs are unable to understand the vehicular network dynamics. Accordingly, the contact time interval that meets the requirement of the vehicles and the frequency of collection is less than that of the proposed algorithm, resulting in larger AoI in comparison with that in our algorithm. We can also observe from the figure that the Greedy scheduling policy is more effective than the random scheduling policy since at each time-slot, the Greedy policy always selects the vehicle with the highest value of AoI.

Finally, the convergence performance of the proposed DRL algorithm is studied. The total reward is calculated as the sum of the cost of each action of the UAVs, which are the EWSA and the deployment violation of UAVs. As shown in Fig. 6.9, the cumulative reward increases relatively quickly at the beginning of learning and when the number of episodes reaches a certain number (1000), the increase becomes relatively slow. This is because at the beginning of the training phase, the DRL agent starts to learn the deployment violation of UAVs such as road segment borders and safety distance between the deployed UAVs to avoid penalties. In addition, many vehicles were not yet properly scheduled to transmit their status-update packets since the UAVs had not yet learnt the suitable trajectories in the dynamic environment in order to achieve the required rate that minimizes the AoI. The trained DRL agent can significantly enhance the defined reward. This improvement

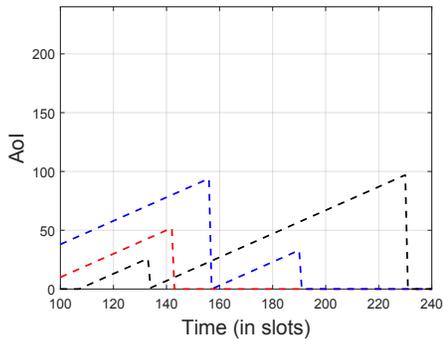
begins to diminish when the DRL agent is well trained about the road segment's borders and safety distance between the deployed UAVs and it starts to effectively adapt the UAVs' trajectories. It is worth mentioning that due to the non-stationarity (i.e. stable dynamics) of the environment, the cumulative reward is varying around its average while overall increasing with more trials (i.e., training). A similar observation has been reported in [90].

6.5 Summary

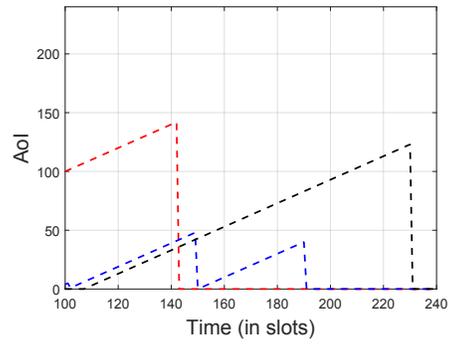
We investigated the use of UAVs to support Intelligent Transportation System applications. We studied the problem of optimizing the trajectory of multiple UAVs and scheduling policies to minimize the Expected Weighted Sum AoI of the vehicles in a vehicular network. More specifically, we formulated the design problem as a mixed integer non-linear program and proposed the DRL approach to learn the vehicular environment and its dynamics in order to control the UAVs' trajectories in the dynamic environment. Simulation results showed that the DRL approach achieved the lowest Expected Weighted Sum AoI and average age compared to other benchmark approaches, including fixed and random trajectories approaches, and static UAV placement.



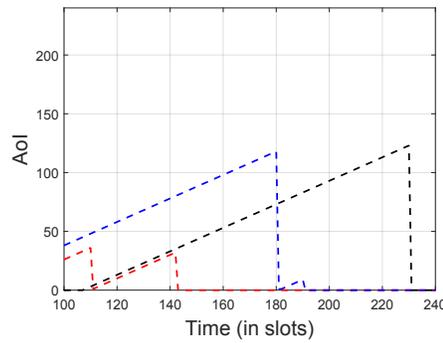
(a) DRI



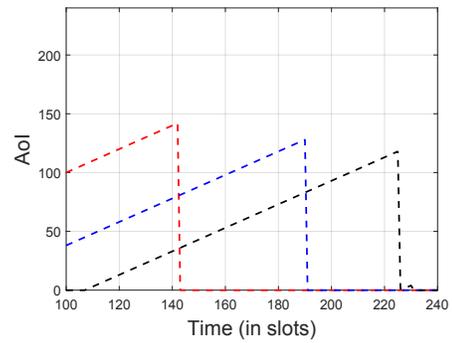
(b) RTGS



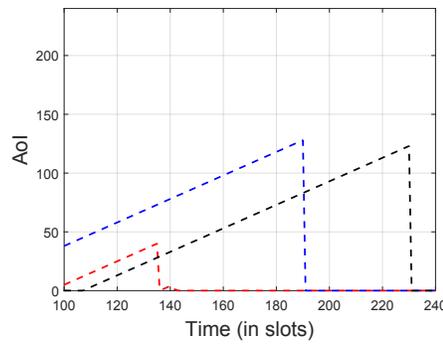
(c) RTRS



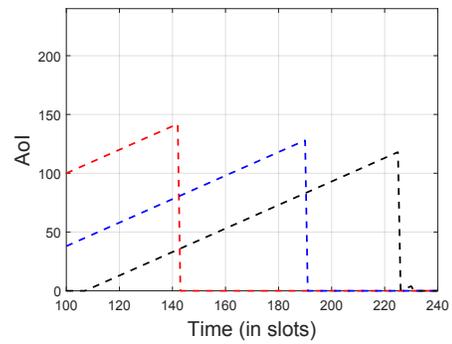
(d) RTGS



(e) RTRS



(f) RTGS.



(g) RTRS.

Fig. 6.8: The performance comparison of different policies for a sample of three vehicles.

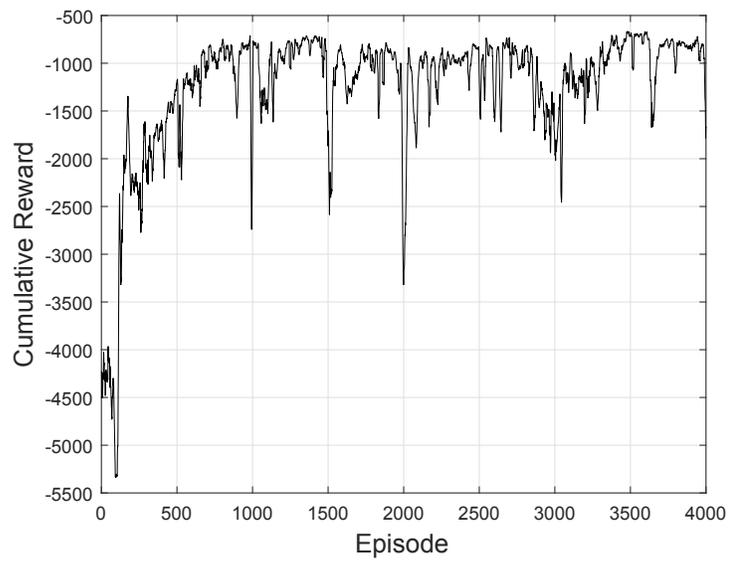


Fig. 6.9: Accumulated reward over time.

Chapter 7

On-Demand Content Delivery in Vehicular Networks with Optimized Multi-UAV Trajectories

7.1 Background, Related Works, and Contributions

As discussed in Chapter 5, Infrastructure-based communication networks tend to be susceptible to major damage arising from either natural disasters (e.g., hurricanes, etc.) or human-made ones (e.g., wars, explosions, etc.). Each of such events has the potential to damage or even destroy a country's communication infrastructure [3]. Such incidents have therefore demonstrated the need to have a quick, efficient, self-configuring, and infrastructure-less wireless network for emergency cases. Owing to their agility and mobility, UAVs are being promoted as a promising solution to provide fast network recovery when the infrastructure is temporarily unavailable. They can be deployed to enhance the coverage of cellular networks during an unplanned surge in traffic demand [67].

In this chapter, we propose dispatching multiple UAVs that cooperatively serve vehicles on a highway with limited or no communication infrastructure. Multiple existing works demonstrated

The work done in this chapter leads an IEEE published journal [72]

the benefits that a single UAV can deliver in emergency situations [67]. One UAV alone, however, may not likely meet the requirements of all vehicles moving at different speeds on a given road segment in a timely manner. Motivated by this, deployment of a swarm of UAVs is required to deliver critical information in vehicular networks.

In this work, we aim to minimize the number of deployed UAVs by jointly optimizing the UAVs trajectories and radio resource allocation in a given period, to guarantee the vehicles' requirements in terms of downloading all needed data subject to UAVs' and vehicles' mobility constraints, and before the vehicles depart a given road segment.

7.2 System Model

We consider a highway segment with damaged or unavailable communication infrastructure. Further, this segment has unidirectional free traffic flow of vehicles that depart the coverage of a Road Side Unit (RSU) as illustrated in (Fig. 7.1), where this RSU is assumed to be equipped with M UAVs that are intended to deliver critical data to vehicles crossing the given highway segment. The UAVs have their data cached from a centralized content server before they leave the RSU.

We consider multiple time frames with duration T where each frame (few minutes) is divided into N equal-time slots, each with length δ_t (few seconds), indexed by $n = 1, \dots, N$. We use \mathcal{V}^n to denote the subset of vehicles to be served, in time slot n , where $\mathcal{V} = \mathcal{V}^1 \cup \mathcal{V}^2 \cup \dots \cup \mathcal{V}^N$. We consider one time frame where the arrival and requirement for all vehicles within T can be accurately estimated. Examples of content to be delivered to vehicles include critical safety information, streaming service, etc. Each UAV has an onboard unit through which it receives and likely processes the content during its residence on the highway segment.

The UAVs are assumed to have high capacity fronthaul links (such as free space optics (FSO) or millimeter-wave (mmWave) links) with ingress RSU, where a central unit updates the content of the deployed UAVs and manages the cooperation between them. Therefore, data that cannot be completely delivered to one vehicle while being within the coverage of one UAV will resume its download once the vehicle gets connected with other deployed UAVs. By considering vehicle mobility and data requirement, this work aims is to dispatch just enough UAVs from the ingress

RSU to serve all vehicles before exiting the highway segment.

For simplicity, we assume the vehicle declares its required content to the ingress RSU before it enters the given highway segment. The content requested by each vehicle will be fully delivered by the UAVs within the vehicle's resident time on the considered segment. We adopt a widely used traffic model on the highway [79], where vehicles in each direction travel with different speeds generated according to a truncated Gaussian distribution in the range $[v_{min}, v_{max}]$ [81]. We assume that vehicles keep the same speed during the entire navigation period along the segment [101]. The flow of vehicles entering the desired highway segment follows Poisson distribution with arrival rate λ veh/s. Moreover, the initial positions and speeds of vehicles are assumed to be known through Differential Global Positioning System provided by the ingress RSU, and communicated to the UAVs through the fronthaul links. Therefore, the instantaneous position w_i^n of each vehicle $i \in \mathcal{V}$, at any time slot n can be calculated. According to federal aviation regulations, all UAVs are assumed to fly at a constant altitude H above ground level and each UAV m is located at $(x_m^n, 0, H)$, at time slot n , where the width of the lane is ignored as compared to the transmission range of vehicles and UAVs [88]. During the considered time frame, vehicles enter and leave the highway segment resulting in a change in the number of vehicles in \mathcal{V}^n . We are interested in the arrival and departure times of vehicles causing that change. Let a_i and d_i be the arrival and departure times of vehicle i to the highway segment, respectively. For each vehicle i , a_i and d_i can be calculated independently using the vehicle speed and highway distance. In our model, vehicles in set \mathcal{V} may request different content sizes from the UAVs, and UAVs can simultaneously communicate with multiples vehicles on different spectrums by allocating appropriate resources.

In practice, the following equations govern the UAV trajectories

$$|x_m^{n+1} - x_m^n| \leq V_{max} \delta_t, n = 1, \dots, N-1, \forall m, \quad (7.1)$$

$$x_m^0 = x_s, x_m^N = x_c, \forall m, \quad (7.2)$$

$$|x_m^n - x_j^n| \geq d_{min}, \forall m, m \neq j. \quad (7.3)$$

Eq. 7.1, limits the distance travelled by one UAV in every time slot based on the maximum

UAV speed V_{max} in m/s. Eq. 7.2, specifies the initial position of each UAV to be the beginning of the high segment at x_s and the final position to be the end of segment at x_c . In fact, the operator may decide on those positions based on multiple factors such as the location of their managed property, legislation and/or UAVs' charging stations. Eq. 7.3, ensures a safety distance d_{min} between UAVs to maintain collision-free trajectories.

In typical UAV assisted communication, the channel is generally modeled using large-scale fading and small scale fading. However, in highway scenarios, such the one considered in this chapter, the UAV-to-vehicle channel can be characterized with strong Line-of-Sight (LoS) and therefore the small scale fading can be neglected. All UAVs are assumed to transmit with constant power P leading to a received power $P_{i,m}^n = h_{i,m}^n P$ in slot n , where $h_{i,m}^n$ is the channel gain from UAV m to vehicle i in time slot n . This channel gain can be written as:

$$h_{i,m}^n = h_o \left(\sqrt{(w_i^n - x_m^n)^2 + H^2} \right)^{-2}, \forall n, m, \quad (7.4)$$

where h_o is the median of the mean path gain at reference distance $d_0 = 1$ m.

We define the service amount as the amount of cached data that the UAVs deliver to each vehicle within their residence on the highway segment. The service amount concept has been proposed in multiple previous papers especially in scenarios with vehicle mobility [25], where the rate is time-variant and does not exhibit the achievable service quality. Similarly, in our system model, the achievable rate at each vehicle varies according to multiple factors including UAV position and speed, vehicle speed, highway distance, etc. Consequently, we utilize the service amount concept to represent the service quality between UAVs and vehicles. The service amount $S_{i,m}$ provided between UAV m and vehicle i over the mission time N can be computed based on the summation of the achievable rates throughout the residence time on the defined highway segment, where the rate experienced by a given vehicle i is set to 0 as soon as it reaches the end of the highway segment at d_i . The service amount can be written as

$$S_{i,m} = \delta_t \sum_{n=0}^N s_{i,m}^n, \forall i \in \mathcal{V}, \forall m, \quad (7.5)$$

$$\text{where: } s_{i,m}^n = \begin{cases} r_{i,m}^n, & \text{if } a_i \leq n \leq d_i, \\ 0, & \text{otherwise.} \end{cases} \quad (7.6)$$

During its residence on the highway segment, vehicle i served by UAV m in time slot n receives rate $r_{i,m}^n = b_{i,m}^n \log_2(1 + P_{i,m}^n/\sigma^2)$, where σ^2 is the thermal noise power which is linearly proportional to the allocated bandwidth [24], and $b_{i,m}^n$ is the fraction of the spectrum resource allocated to vehicle i in time slot n from UAV m and it is equivalent to a number of resource blocks. In practice, we can allocate part of the spectrum for each vehicle, and hence $b_{i,m}^n$ is approximately continuous between 0 and 1.

7.3 Problem Formulation

To mathematically formulate the problem¹, we introduce two binary decision variables: $\gamma_m \in \{0, 1\}$, $\forall m$, that takes the value of 1 if UAV m is deployed and 0 otherwise, $y_{i,m}^n \in \{0, 1\}$ indicates whether UAV m is serving vehicle i in time slot n . Thus, our optimization problem is formulated as:

$$\begin{aligned} \mathcal{O} \mathcal{P}_1: & \min_{\substack{\gamma_m, b_{i,m}^n, \\ x_m^n, y_{i,m}^n}} \sum_{m=1}^M \gamma_m \\ \text{s.t. } \mathcal{C}1: & \delta_i \sum_{m=1}^M \sum_{n=0}^N s_{i,m}^n \geq S_i^{\min}, \forall i \in \mathcal{V}, \\ \mathcal{C}2: & \gamma_m \in \{0, 1\}, \forall m, \\ \mathcal{C}3: & y_{i,m}^n \in \{0, 1\}, \forall m, i \in \mathcal{V}^n, \forall n, \\ \mathcal{C}4: & |x_m^n - w_i^n| \leq R_c + (1 - y_{i,m}^n)K, \forall m, i \in \mathcal{V}^n, \forall n, \\ \mathcal{C}5: & \sum_{i=1}^{\mathcal{V}^n} b_{i,m}^n \leq \gamma_m, \forall n, m, \\ \mathcal{C}6: & 0 \leq b_{i,m}^n \leq y_{i,m}^n, \forall m, i \in \mathcal{V}^n, \forall n, \\ \mathcal{C}7: & \sum_{m=1}^M y_{i,m}^n \leq 1, \forall i \in \mathcal{V}^n, \forall n, \end{aligned}$$

¹For simplicity, consider one time frame, however, the optimization can be run iteratively to account for subsequent time frames.

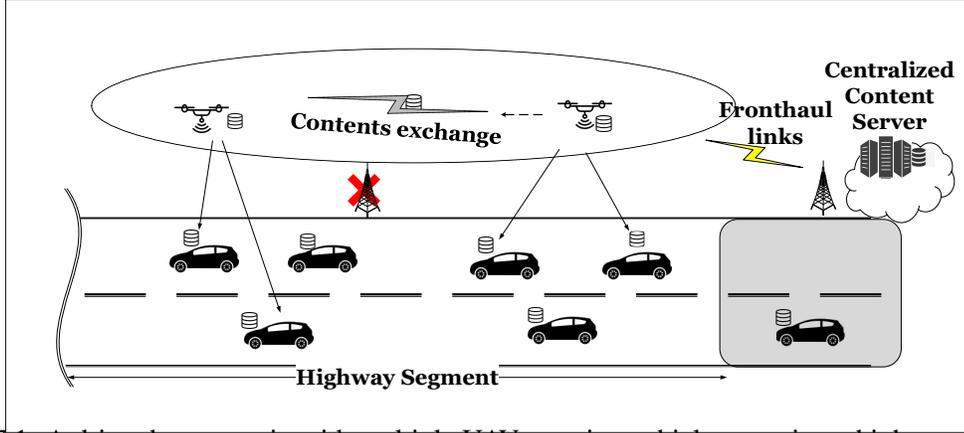


Fig. 7.1: A drive-thru scenario with multiple UAVs serving vehicles crossing a highway segment.

$$\mathcal{C}8 : x_m^0 = x_s, x_m^N = \gamma_m x_c + (1 - \gamma_m) x_s, \forall m,$$

$$\mathcal{C}9 : |x_m^n - x_j^n| \geq (\gamma_m + \gamma_j - 1) d_{min}, \forall m, m \neq j, n = 2, \dots, N - 1,$$

$$\mathcal{C}10 : |x_m^{n+1} - x_m^n| \leq \gamma_m V_{max} \delta_t, n = 1, \dots, N - 1, \forall m.$$

Constraint $\mathcal{C}1$ guarantees that each vehicle downloads its requested amount of data S_i^{min} in (bits/Hz) within their residence on the highway segment. $\mathcal{C}3$ and $\mathcal{C}4$ ensure that vehicle i lies within the UAV communication range R_c projected on the ground, if it is served by the deployed UAV m , where K is a large number that is used to ensure the validity of $\mathcal{C}4$. $\mathcal{C}5$ prevents wasting radio resources to UAVs that are not dispatched. $\mathcal{C}6$ ensures that the total allocated resources by one UAV is less than the available resource for every deployed UAV. $\mathcal{C}7$ ensures that one vehicle is served by at most one UAV at a time. $\mathcal{C}8$ indicates the initial and the final positions of the UAVs. $\mathcal{C}9$ guarantees that the deployed UAVs are sufficiently separated a minimum safety distance d_{min} . Finally, $\mathcal{C}10$ limits the distance traveled by one UAV in one time slot based on its maximum speed.

There are several challenges to solve \mathcal{OP}_1 including the nonconvexity of $\mathcal{C}1$ with respect to UAVs' trajectories and the binary variables. Therefore, \mathcal{OP}_1 constitutes mixed-integer non-convex problem, which is difficult to be optimally solved.

7.4 Proposed Solution

In this section, we attempt to efficiently solve our problem defined in $\mathcal{O}\mathcal{P}_1$ based on convex approximation methods and multiple equivalent transformations to generate a more efficient but sub-optimal solution. The nonconvex constraint in $\mathcal{C}1$ is transformed into another equivalent convex constraint form and successive convex approximation, SCA, optimization method is applied to solve it iteratively. As mentioned earlier, the Problem $\mathcal{O}\mathcal{P}_1$ is non-convex due to having $s_{i,m}^n$ as a function of the UAVs' trajectories and the resource allocation $b_{i,m}^n$ in $\mathcal{C}1$. To tackle the problem, we introduce slack variables $u_{i,m}^n \geq 0, \forall n, m, i \in \mathcal{V}$ and $t_{i,m}^n \geq 0, \forall n, m, i \in \mathcal{V}$, and rewrite $\mathcal{C}1$ as $\mathcal{C}1.1$, $\mathcal{C}1.2$, and $\mathcal{C}1.3$, where $u_{i,m}^n$ is lower bounded by a convex approximation approximation $\zeta_{i,m}^n$ with respect to $(w_i^n - x_m^n)^2$, where at each r^{th} iteration:

$$\begin{aligned} \zeta_{i,m}^n &= F_{i,m}^{r,n} - G_{i,m}^{r,n} \left((w_i^n - x_m^n)^2 - (w_i^{r,n} - x_m^{r,n})^2 \right), \\ F_{i,m}^{r,n} &= \log_2 \left(1 + \frac{Ph_0}{\sigma^2 (H^2 + (x_i^n - x_m^{r,n})^2)} \right), \forall i \in \mathcal{V}^n, n, \\ G_{i,m}^{r,n} &= \frac{(Ph_0/\sigma^2) \log_2 e}{\left(H^2 + (x_i^n - x_m^{r,n})^2 + (Ph_0/\sigma^2) \right) \left(H^2 + (x_i^n - x_m^{r,n})^2 \right)}. \end{aligned} \quad (7.8)$$

Next, we relax and rewrite the binary constraint in $\mathcal{C}3$ in the following equivalent form [102]:

$$y_{i,m}^n - (y_{i,m}^n)^2 \leq 0, \quad (7.9a)$$

$$0 \leq y_{i,m}^n \leq 1. \quad (7.9b)$$

Solving the approximated problem by applying the SCA method remains infeasible due to (7.9), which leads to a failed convergence of the SCA method. Inspired by the approach in [102], we overcome this issue by reformulating the objective function as presented in $\mathcal{O}\mathcal{P}_2$:

$$\begin{aligned} \mathcal{O}\mathcal{P}_2: \quad & \min_{\substack{\gamma_m, b_{i,m}^n, \\ x_m^n, u_{i,m}^n \geq 0, \\ t_{i,m}^n \geq 0, \theta_{i,m}^n}} \sum_{m=1}^M \gamma_m + A \sum_{m=1}^M \sum_{n=1}^N \sum_{i \in \mathcal{V}} \theta_{i,m}^n \\ & \text{s.t. } \mathcal{C}1.1: \delta_t \sum_{m=1}^M \sum_{n=1}^N t_{i,m}^n \geq S_i^{\min}, i \in \mathcal{V}, \end{aligned}$$

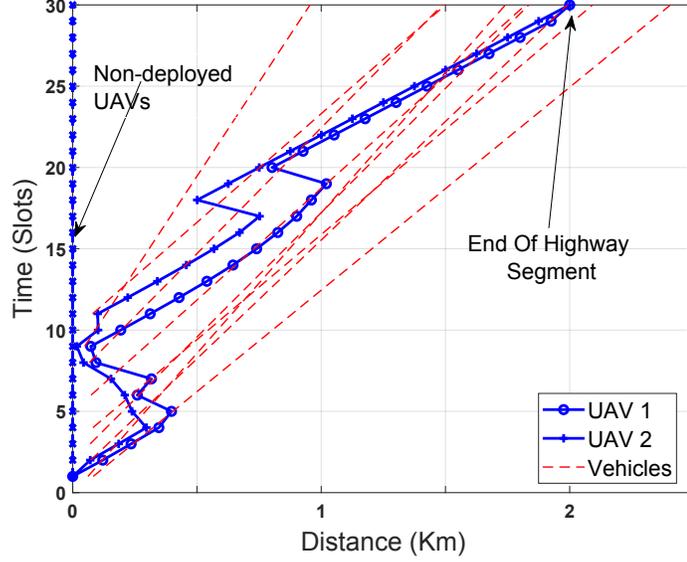


Fig. 7.2: UAVs trajectories.

$$\mathcal{C}1.2 : t_{i,m}^n \leq b_{i,m}^n u_{i,m}^n, \forall n, m, i \in \mathcal{V},$$

$$\mathcal{C}1.3 : u_{i,m}^n \leq \zeta_{i,m}^n, \forall m, i \in \mathcal{V}^n, \forall n,$$

$$\mathcal{C}3.1 : y_{i,m}^n - (y_{i,m}^n)^2 \leq \theta_{i,m}^n, \forall m, i \in \mathcal{V}^n, \forall n,$$

$$\mathcal{C}3.2 : 0 \leq y_{i,m}^n \leq 1, \forall n, m, i \in \mathcal{V},$$

$$\mathcal{C}2, \mathcal{C}4, \mathcal{C}5, \mathcal{C}6, \mathcal{C}7, \mathcal{C}8, \mathcal{C}9, \mathcal{C}10,$$

where $\{\theta_{i,m}^n \geq 0, \forall n, m, i \in \mathcal{V}\}$ is a new slack variable and $A \geq 0$ is the penalty parameter. Examining $\mathcal{C}1.2$, the non-convexity factor $b_i^n u_i^n$ is on the greater side of the inequality. To deal with this constraint, we simply replace the right hand side of $\mathcal{C}1.2$ by an equivalent difference-of-convex (DC) function $b_i^n u_i^n = \frac{1}{4}[(b_i^n + u_i^n)^2 - (b_i^n - u_i^n)^2]$, and linearize the concave term $(b_i^n + u_i^n)^2$ at iteration r . Hence, $\mathcal{C}1.2$ is approximated as

$$-\frac{(b_i^{r,n} + u_i^{r,n})^2}{4} - \frac{(b_i^{r,n} - u_i^{r,n})(b_i^n - b_i^{r,n} + u_i^n - u_i^{r,n})}{2} + \frac{(b_i^n - u_i^n)^2}{4} + t_i^n \leq 0. \quad (7.11)$$

Similarly, we approximate the non-convex constraint $\mathcal{C}3.1$ as $y_{i,m}^n - 2y_{i,m}^{r,n}y_{i,m}^n + y_{i,m}^{r,n} \leq \theta_{i,m}^n$. Using the above approximation, $\mathcal{O} \mathcal{P}_2$ transforms into a Mixed Integer Quadratically Constrained Program (MIQCP) making several methods handy including CVX-MOSEK toolbox [103]. The algorithm

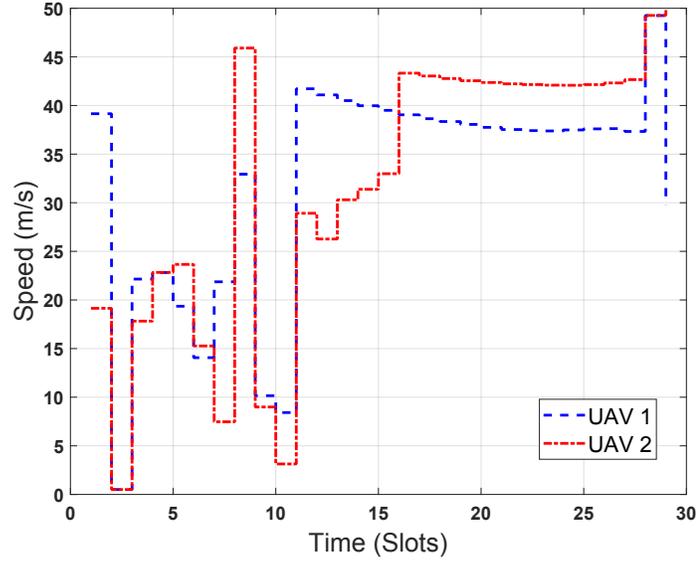


Fig. 7.3: UAVs speeds.

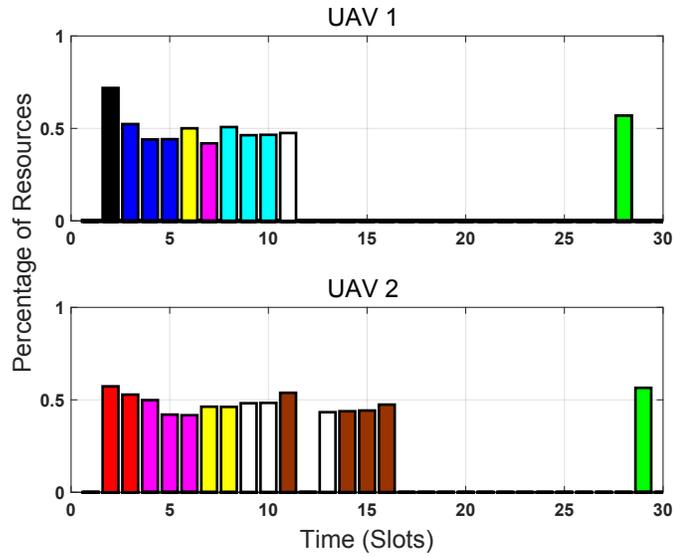


Fig. 7.4: Resource allocation.

proceeds until the number of UAVs converges. The overall complexity of solving \mathcal{OP}_2 depends on the solver that is employed to solve \mathcal{OP}_2 . In particular, \mathcal{OP}_2 is a MIQCP and, thus, several interior-point solvers can be employed to solve it. Therefore, we can employ the number of Newton steps, as a metric to measure its complexity. Therefore, the overall complexity of solving \mathcal{OP}_2 is approximately $I\sqrt{M(4NV + N + 1)}$ in the worst-case, where I is a finite number of iterations.

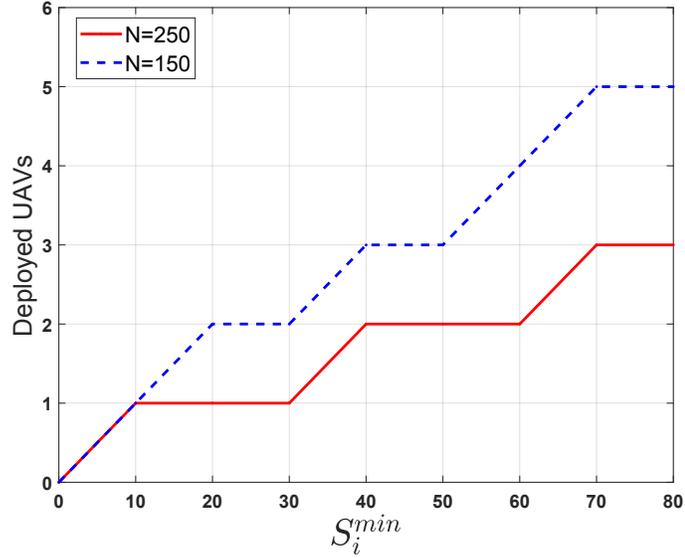


Fig. 7.5: Impact of S_i^{\min} .

7.5 Simulation Results and Discussion

In order to deliver realistic results, the simulation parameters should be an accurate representation of a real highway scenario. We assume a highway segment of length 4km in which multiple UAVs are dispatched with safety distance $d_{min} = 100\text{m}$ with communication range $R_c = 50\text{m}$ to provide streaming services to vehicles. The flow of vehicles entering the highway segment follows Poisson distribution with arrival rate 0.5veh/s . Vehicles velocities are randomly generated using a truncated Gaussian distribution with mean equal 90km/h , variance 16km/h , and velocities can be varied between $20\text{--}140\text{km/h}$. The channel power gain has been taken equal to $|h|^2 = -50\text{dB}$, noise power is $N_o = -110\text{dBm}$. We consider UAVs fly at a constant altitude $H = 100\text{m}$, with transmit power $P = 0.1\text{W}$, and maximum speed $V_{\max} = 50\text{m/s}$.

The CVX toolbox and numerical convex optimization solver MOSEK are used to solve our optimization. Without loss of generality, we assume that, at time $n = 0$, all UAVs are located at position $x_s = 0\text{m}$ and the final location of deployed UAVs is the end the highway segment $x_c = 4\text{km}$. Fig. 7.2 depicts the UAVs trajectories to provide services to vehicles (for $\mathcal{V} = 9$ vehicles with $S_i^{\min} = 30\text{bits/Hz}$) over a period $N = 30$ time slots. Each time slot is of length 5s . The vehicles enter the highway segment at different times as depicted in the figure. It also shows that only 2 UAVs are

needed to fulfill the requirements of all vehicles within the considered time period.

Due to the flexibility of UAVs (rotary-wing UAVs), Fig. 7.2 also shows that the UAVs start their trajectories by following the first batch of arriving vehicle(s) then move to follow the second subsequent batch and so on. Fig. 7.3 presents the change in speed of both UAVs to allow them follow the batch of vehicles they are serving then. Examining (a) and (b) of Fig. 2, one notes that both UAVs fly at a very high speed to reach the end of the highway segment and serve vehicles before they depart and before the mission time is over. We also observe that, while a UAV decreases its speed to follow a batch of vehicles, the second UAV dramatically drops its speed to maintain the safety distance d_{min} .

Fig. 7.4, demonstrates that at each time slot n the UAVs allocate the radio resources unequally among the vehicles depending on their arriving times and current locations. In this figure, resources allocated to different vehicles are marked in different colors. Fig. 7.4 also shows that the same vehicle may be served by both UAVs but each in a different time slot. It can be also seen that, due to dynamics of vehicles, the UAVs may not be able to allocate its resources continuously and have to serve some vehicles toward the end of the highway segment.

Clearly, 2 UAVs may not able to meet the vehicles' requirements for all service rates. Next, we study the impact of the minimum service amount S_i^{min} on the proposed solution over different mission time (in time slots). As shown in Fig. 7.5, with the lower service amount, optimizing the radio resources is sufficient to fulfill the vehicles' requirements with one or 2 UAVs. With increasing the minimum service amount, 2 UAVs cannot anymore fulfill to fully serve the vehicles through optimizing their radio resources. Increasing the number of deployed UAVs and optimizing their trajectories to fly closer to vehicles become more crucial for achieving better communication channels to increase the transmission rate and achieve larger service amount. As a result, the required number of UAVs increases by increasing the required service amount while keeping the other system parameters intact including fixed mission time and radio resources per each UAV. Fig. 7.5 also demonstrates that a larger mission time allows fewer number of UAVs to fully serve all vehicles. If the required service amount is 40 bits/Hz, only 2 UAVs are needed when the mission time is 250 time slots while 3 UAVs will be required if the mission time drops to 150 time slots.

7.6 Summary

This chapter studied the trajectories of multiple UAVs to serve vehicles in a mobility environment. Since vehicles have different requirements within their residence on the highway segment, the UAVs trajectories and radio resource allocation are optimized to provide vehicles with a differentiated amount of data. We formulated our optimization problem to minimize the number of UAVs while guaranteeing service to all vehicles before exiting the highway segment. Resulting in a non-convex problem, we proposed a low-complexity solution and examined its behavior to fulfill the requirement of all vehicles.

Chapter 8

Conclusions and Future Research

Directions

8.1 Conclusion

This thesis addressed multiple challenges associated with the deployment of UAVs to assist wireless communication networks. Chapter 1 of this thesis provided a comprehensive overview of the key advantages, potential applications, and communication challenges associated with the exploitation of UAVs in order to assist wireless communication networks. Then, limitations of existing studies and a summary of the research contributions of this thesis were highlighted.

The first part of the thesis focused on the efficient deployment of UAV assisted static environments. In particular, in Chapter 2, the efficient online and offline trajectory planning and radio resource allocation of a UAV deployed to collect data from time constrained IoTDS were investigated. To enable reliable uplink communications for time-constrained IoTDS, a novel framework was proposed to jointly optimize UAV trajectory and communication. The optimization problem was to maximize the number of served IoTDS while guaranteeing minimal amount of data uploaded from each served device within the given deadline. Although the problem was non-convex, it was solved optimally using BRB algorithm. By convexifying the problem, a low-complexity solution was obtained, then we extended the solution to obtain an enhanced trajectory in order to minimize the distance traveled by the UAV while serving the IoTDS.

In Chapter 3, a novel model in UAV-assisted IoT networks was proposed to minimize age of information. It takes into account channel reliability between IoTs and UAVs and between UAVs and BS to minimize the AoI. A concrete analytical characterization of AoI for UAV-assisted IoT networks was derived when UAVs with virtual queues act as mobile active relays between IoTs and the BS. A Markov Decision Process problem was formulated to find the optimal altitude and scheduling policy that minimizes the AoI. Then, a PPO algorithm was developed to learn environment dynamics in order to control the altitude and scheduling policy of UAVs. In contrast to UAVs as mobile active relays, integrating RIS with UAVs does not require any radio frequency chain circuit to relay transmission from the IoTs to the BS. This provides a cost-effective solution with minimal energy consumption. In Chapter 4, the benefit of integrating RIS as a passive relay with UAVs in the performance of AoI was investigated. An optimization problem with the objective of minimizing the AoI was formulated to optimize the altitude of the UAV, communication schedule, and phase shifts of RIS elements. In the absence of prior knowledge of the activation pattern of the IoTs, a PPO algorithm was developed to learn the randomness of the IoTs' activation patterns and to solve this mixed-integer non-convex optimization problem.

The second part of this thesis focused on the efficient deployment of UAV-assisted vehicular networks. In Chapter 5, coverage problem was studied for UAV-assisted vehicular networks in the absence of a complete knowledge of the environment. The coverage problem was formulated as a Markov Decision Process. Then, the Actor Critic algorithm was proposed to explore the unknown environment and plan trajectories for a minimum number of UAVs. This was done in order to provide network connectivity for vehicles while minimizing the energy consumption of the deployed UAVs. In Chapter 6, the use of UAVs to support Intelligent Transportation System applications was explored. The problem of optimizing the trajectory and scheduling policies of multiple UAVs to keep the information fresh from vehicles in a vehicular network was studied. More specifically, the data collection problem was formulated as mixed-integer and non-convex problem. The DRL approach was proposed to learn the vehicular environment and its dynamics in order to control the UAVs' trajectories in the dynamic environment. DDPG was exploited to learn the trajectories of the deployed UAVs and to efficiently minimize the EWSA. In Chapter 7, a novel framework was developed for effective content delivery through cache enabled UAVs in vehicular networks. The content

delivery problem was formulated as an optimization problem with the objective of minimizing the number of UAVs while guaranteeing service to all vehicles before exiting the highway segment. The formulated problem was convexified and a low-complexity solution was obtained.

8.2 Future Work

8.2.1 Further Deployment of UAVs for 5G and Beyond Networks

Although this dissertation covered several research challenges related to the deployment of UAVs in assisting 5G and beyond, other problems that need to be investigated still exist. Results in Chapter 2 showed that a single UAV is not enough to meet the requirements of all IoTDs in a timely manner especially when the area or the number of devices increase. It would therefore be necessary to explore a joint optimization of trajectories and radio resource allocation of multiple UAVs. Since timely data collection was also evaluated in Chapter 2 for Orthogonal Multiple Access (OMA), Non-Orthogonal Multiple Access (NOMA) based UAVs should also be considered to accommodate a large number of IoTDs. The analysis of UAV and IoT energy consumption for a time constrained data collection scenario is another area that should be studied. In Chapter 3, a concrete analytical characterization of AoI for two-hop transmission was provided when UAVs with virtual queues act as mobile active relays between IoTDs and the BS. Future research should focus on extending this framework to consider multiple hops instead of only two hops (for example, multiple relays to the BS through multiple UAVs). Resource management is also another key future research area with respect to radio resources, UAV energy consumption, IoTDs' transmission power, UAVs' mission time, and number of UAVs, among others.

In Chapter 4, data freshness in aerial-RIS assisted static environments was investigated. However, destinations could be mobile with random velocities. In vehicular environments, vehicles may suffer from poor connections when navigating through a dark zone. In order to overcome this issue, RSUs can exploit the RIS technology to relay information to vehicles. However, high mobility of the vehicles leads to distinct network conditions and changes the network topology, which consequently complicates the design of efficient scheduling and the application of the RIS configuration approach. In terms of UAV assisted vehicular networks, there is still a need to study the effect of

backhaul stability on the overall system performance. Achieving a seamless handover among UAVs in such a dynamic environment also needs to be explored. Finally, a study on efficient scheduling techniques to mitigate interference between UAVs and ground base stations in a UAV assisted cellular network is another area of future research.

8.2.2 Safety and Security with UAVs

Safety and security are among the major challenges hindering the full exploitation of UAVs and their deployment in commercial applications. These challenges stem from the fact that UAVs are unmanned and they are controlled through wireless links. Also, in some cases, UAVs fly as low as 100-200 meters from the ground and at a high speed. As such, one might think of them as flying objects with a remote control. In the event that control is lost or compromised, such objects can become very harmful to humans on the ground. Transport Canada's Civil Aviation Daily Occurrence Reporting System (CADORS) reported 355 incidents caused by UAVs between 2005 and 2016 [104]. The number of such accidents will inevitably increase as the usage of UAVs increases. As for security, similar to any wireless environment, the fact that UAVs are controlled through wireless links poses serious threats as the control system can be hacked. An adversary might also use a number of controlled UAVs to interfere with existing operations of legitimate UAVs or to interrupt services by performing several malicious activities. Furthermore, UAVs equipped with different types of explosive materials might be used as a weapon for attack during critical operations such as offensive missions and terrorist attack. For instance, in September 2019, small UAVs were used to attack strategic facilities in Saudi Arabia leading to a major destabilization of global financial markets [105]. Based on the above mentioned challenges, a timely detection and tracking of unauthorized or hijacked UAVs is necessary in order to reduce or even eliminate the potential risks associated with a possible intrusion. Multiple detection and tracking techniques such as ground radar and computer vision have been extensively studied in literature. However, their detection/tracking accuracy is significantly affected by the environment, particularly in urban areas where obstacles such as buildings make these solutions infeasible. In fact, the detection and tracking of small mobile objects such as UAVs flying at moderate altitudes can be even more problematic due to the fact that it is extremely difficult to distinguish between small flying objects such as birds

and these controlled UAVs.

As a future direction, different scenarios for detecting and tracking UAVs will be investigated. More specifically, the following scenarios will be considered: (i) legitimate UAVs that lose their wireless connections with ground control station, and (ii) unauthorized UAVs flying around or within restricted areas. The overarching objective of this research direction is to accurately and timely detect and track uncontrolled UAVs by dispatching a swarm of UAVs whose trajectories have been optimized. Finding the optimal trajectories of multiple UAVs in order to provide an efficient and timely solution for detecting and tracking uncontrolled UAVs is an existing and unaddressed problem to the best of our knowledge. Specific objectives of this future direction include: (1) develop an artificial intelligence technique, especially deep reinforcement learning approach, to train a swarm of UAVs to detect and track uncontrolled UAVs and autonomously decide the proper trajectories of the deployed UAVs in order to continue tracking uncontrolled UAVs, (2) develop deep reinforcement learning technique to train deployed UAVs in multiple scenarios, i.e. first scenario involves passive uncontrolled UAVs (no active radio frequency signal is emitted from the uncontrolled one). In this case, deployed UAVs equipped with radar technology will be considered in this work. Second scenario involves active UAVs, where the swarm of UAVs will track the received signal strength, and 3) consider practical system configurations/designs according to US Federal aviation regulations that can be adopted in future wireless communication systems.

Bibliography

- [1] K. Letaief *et al.*, “The roadmap to 6G: AI empowered wireless networks,” *IEEE Commun. Mag.*, vol. 57, no. 8, pp. 84–90, Aug. 2019.
- [2] “Paving the path to 5G: Optimizing commercial LTE networks for drone communication,” [online] available at : <https://www.qualcomm.com/news/onq/2016/09/06/paving-path-5goptimizing-commercial-lte-networks-drone-communication>, Qualcomm, 2006.
- [3] P. Bupe *et al.*, “Relief and emergency communication network based on an autonomous decentralized UAV clustering network,” in *Proc. SoutheastCon*, April 2015.
- [4] The Wall Street Journal, “Wireless networks suffer in irma’s wake,” [online] available at : <https://www.wsj.com/livecoverage/hurricane-irma/card/1505235512>, 2017.
- [5] CNN, “Irma: A hurricane for the history books,” *CNN* [online] available at : <https://www.cnn.com/specials/hurricane-irma>, 2017.
- [6] M. Samir *et al.*, “UAV trajectory planning for data collection from time-constrained IoT devices,” *IEEE Trans. Wireless Commun.*, pp. 1–1, Sep. 2019.
- [7] Q. Wu *et al.*, “Joint trajectory and communication design for UAV-enabled multiple access,” in *Proc. IEEE GLOBECOM*, Dec 2017, pp. 1–6.
- [8] ———, “Capacity characterization of UAV-enabled two-user broadcast channel,” *IEEE J. Sel. Areas Commun.*, vol. 36, no. 9, pp. 1955–1971, Sep. 2018.
- [9] P. Li *et al.*, “Fundamental rate limits of UAV-enabled multiple access channel with trajectory optimization,” *IEEE Trans. Wireless Commun.*, vol. 19, no. 1, pp. 458–474, Jan. 2020.

- [10] Q. Wu *et al.*, “Joint trajectory and communication design for multi-UAV enabled wireless networks,” *IEEE Trans. Wireless Commun.*, vol. 17, no. 3, pp. 2109–2121, March 2018.
- [11] Y. Zeng *et al.*, “Trajectory design for completion time minimization in UAV-enabled multicasting,” *IEEE Trans. Wireless Commun.*, vol. 17, no. 4, pp. 2233–2246, April 2018.
- [12] J. Lyu *et al.*, “Spectrum sharing and cyclical multiple access in UAV-aided cellular offloading,” in *Proc. IEEE GLOBECOM*, Dec 2017, pp. 1–6.
- [13] D. Yang *et al.*, “Energy trade-off in ground-to-UAV communication via trajectory design,” *IEEE Trans. Veh. Technol.*, pp. 1–1, March 2018.
- [14] Y. Zeng *et al.*, “Throughput maximization for UAV-enabled mobile relaying systems,” *IEEE Trans. Commun.*, vol. 64, no. 12, pp. 4983–4996, Dec. 2016.
- [15] T. Nguyen *et al.*, “A novel cooperative NOMA for designing UAV-assisted wireless backhaul networks,” *IEEE J. Sel. Areas Commun.*, vol. 36, no. 11, pp. 2497–2507, Nov. 2018.
- [16] Q. Wu *et al.*, “Common throughput maximization in UAV-enabled OFDMA systems with delay consideration,” *IEEE Trans. Commun.*, vol. 66, no. 12, pp. 6614–6627, Dec. 2018.
- [17] S. Say *et al.*, “Priority-based data gathering framework in UAV-assisted wireless sensor networks,” *IEEE Sensors J.*, vol. 16, no. 14, pp. 5785–5794, July 2016.
- [18] C. Zhan *et al.*, “Energy-efficient data collection in UAV enabled wireless sensor network,” *IEEE Wireless Commun. Lett.*, Nov. 2017.
- [19] M. Mozaffari *et al.*, “Mobile unmanned aerial vehicles (UAVs) for energy-efficient internet of things communications,” *IEEE Trans. Wireless Commun.*, vol. 16, no. 11, pp. 7574–7589, Nov. 2017.
- [20] C. Zhan *et al.*, “Trajectory design for distributed estimation in UAV-enabled wireless sensor network,” *IEEE Trans. Veh. Technol.*, vol. 67, no. 10, pp. 10 155–10 159, Oct. 2018.
- [21] D. Ebrahimi *et al.*, “UAV-aided projection-based compressive data gathering in wireless sensor networks,” *IEEE Internet Things J.*, vol. 6, no. 2, pp. 1893–1905, April 2019.

- [22] J. Liu *et al.*, “Age-optimal trajectory planning for UAV-assisted data collection,” in *Proc. IEEE INFOCOM*, 2018, pp. 553–558.
- [23] M. Abd-Elmagid *et al.*, “Average peak age-of-information minimization in UAV-assisted IoT networks,” *IEEE Trans. Veh. Technol.*, vol. 68, no. 2, pp. 2003–2008, Feb. 2019.
- [24] M. Mozaffari *et al.*, “Optimal transport theory for cell association in UAV-enabled cellular networks,” *IEEE Commun. Lett.*, vol. 21, no. 9, pp. 2053–2056, Sep. 2017.
- [25] K. Xiong *et al.*, “Mobile service amount based link scheduling for high-mobility cooperative vehicular networks,” *IEEE Trans. Veh. Technol.*, vol. 66, no. 10, pp. 9521–9533, Oct. 2017.
- [26] K. Xiong *et al.*, “Towards 5G high mobility: A fairness-adjustable time-domain power allocation approach,” *IEEE Access*, vol. 5, pp. 11 817–11 831, June 2017.
- [27] X. Chen *et al.*, “Massive MIMO beamforming with transmit diversity for high mobility wireless communications,” *IEEE Access*, vol. 5, pp. 23 032–23 045, Oct. 2017.
- [28] H. Tuy *et al.*, *Monotonic Optimization: Branch and Cut Methods*. Boston, MA: Springer US, 2005, pp. 39–78.
- [29] T. Nguyen *et al.*, “Centralized and distributed energy efficiency designs in wireless backhaul hetnets,” *IEEE Trans. Wireless Commun.*, vol. 16, no. 7, pp. 4711–4726, July 2017.
- [30] F. Zhou *et al.*, “Computation rate maximization in UAV-enabled wireless-powered mobile-edge computing systems,” *IEEE J. Sel. Areas Commun.*, vol. 36, no. 9, pp. 1927–1941, Sep. 2018.
- [31] M. Arfaoui *et al.*, “Secrecy performance of multi-user MISO VLC broadcast channels with confidential messages,” *IEEE Trans. Wireless Commun.*, vol. 17, no. 11, pp. 7789–7800, Nov. 2018.
- [32] S. Boyd, *Advances in Convex Optimization: Interior-point Methods, Cone Programming, and Applications*, 2002. [Online]. Available: <http://www.dii.unisi.it/~control/seminars/boyd/notes/cdc02.pdf>

- [33] H. Sallouha *et al.*, “Energy-constrained UAV trajectory design for ground node localization,” in *Proc. IEEE GLOBECOM*, Dec. 2018, pp. 1–7.
- [34] M. Samir *et al.*, “Online altitude control and scheduling policy for minimizing AoI in UAV-assisted iot wireless networks,” *IEEE Trans. Mobile Comput.*, pp. 1–1, Dec. 2020.
- [35] BBC, “California wildfires: Death toll rises to 25,” 2018. [Online]. Available: <https://www.bbc.com/news/world-us-canada-46168107>
- [36] A. Kosta *et al.*, “Age of information: A new concept, metric, and tool,” *Foundations and Trends in Networking*, vol. 12, no. 3, pp. 162–259, 2017.
- [37] J. Hu *et al.*, “Cooperative internet of UAVs: Distributed trajectory design by multi-agent deep reinforcement learning,” *IEEE Trans. Commun.*, pp. 1–1, Nov. 2020.
- [38] —, “Distributed trajectory design for cooperative internet of UAVs using deep reinforcement learning,” in *Proc. IEEE GLOBECOM*, 2019, pp. 1–6.
- [39] M. Abd-Elmagid *et al.*, “Deep reinforcement learning for minimizing age-of-information in UAV-assisted networks,” in *Proc. IEEE GLOBECOM*, 2019, pp. 1–6.
- [40] W. Li *et al.*, “Minimizing packet expiration loss with path planning in UAV-assisted data sensing,” *IEEE Wireless Commun. Lett.*, vol. 8, no. 6, pp. 1520–1523, Dec. 2019.
- [41] C. Zhou *et al.*, “Deep RL-based trajectory planning for aoi minimization in UAV-assisted IoT,” in *Proc. IEEE WCSP*, 2019, pp. 1–6.
- [42] P. Tong *et al.*, “Deep reinforcement learning for efficient data collection in UAV-aided internet of things,” in *Proc. IEEE ICC*, 2020, pp. 1–6.
- [43] M. Yi *et al.*, “Deep reinforcement learning for fresh data collection in UAV-assisted IoT networks,” in *Proc. IEEE INFOCOM*, 2020, pp. 716–721.
- [44] S. Zhang *et al.*, “Age of information in a cellular internet of UAVs: sensing and communication trade-off design,” *IEEE Trans. Wireless Commun.*, pp. 1–1, Oct. 2020.

- [45] A. Cao *et al.*, “Peak age-of-information minimization of UAV-aided relay transmission,” in *Proc. IEEE ICC*, 2020, pp. 1–6.
- [46] Z. Jia *et al.*, “Age-based path planning and data acquisition in UAV-assisted IoT networks,” in *Proc. IEEE ICC*, May 2019, pp. 1–6.
- [47] P. Tong *et al.*, “UAV-enabled age-optimal data collection in wireless sensor networks,” in *Proc. IEEE ICC*, May 2019, pp. 1–6.
- [48] H. Hu *et al.*, “AoI-minimal trajectory planning and data collection in UAV-assisted wireless powered IoT networks,” *IEEE Internet Things J.*, pp. 1–1, Jan. 2020.
- [49] S. Zhang *et al.*, “Sensing and communication tradeoff design for AoI minimization in a cellular internet of UAVs,” in *Proc. IEEE ICC*, 2020, pp. 1–6.
- [50] A. Al-Hourani *et al.*, “Optimal LAP altitude for maximum coverage,” *IEEE Wireless Commun. Lett.*, vol. 3, no. 6, pp. 569–572, Dec. 2014.
- [51] J. Schulman *et al.*, “Proximal policy optimization algorithms,” 2017. [Online]. Available: <https://arxiv.org/abs/1707.06347>
- [52] ———, “Trust region policy optimization,” in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 1889–1897.
- [53] Y. Zhan *et al.*, “Free market of multi-leader multi-follower mobile crowdsensing: An incentive mechanism design by deep reinforcement learning,” *IEEE Trans. Mobile Comput.*, vol. 19, no. 10, pp. 2316–2329, Oct. 2020.
- [54] U. Challita *et al.*, “Interference management for cellular-connected UAVs: A deep reinforcement learning approach,” *IEEE Trans. Wireless Commun.*, vol. 18, no. 4, pp. 2125–2140, April 2019.
- [55] C. You *et al.*, “Hybrid offline-online design for UAV-enabled data harvesting in probabilistic los channel,” *IEEE Trans. Wireless Commun.*, pp. 1–1, June 2020.

- [56] J. Harmer *et al.*, “Imitation learning with concurrent actions in 3D games,” in *Proc. IEEE CIG*, 2018, pp. 1–8.
- [57] M. Samir *et al.*, “Optimizing age of information through aerial reconfigurable intelligent surfaces: A deep reinforcement learning approach,” *IEEE Trans. Veh. Technol.*, under revision 2020.
- [58] M.Elhattab *et al.*, “Reconfigurable intelligent surface assisted coordinated multipoint in downlink NOMA networks,” *IEEE Commun. Lett.*, pp. 1–1, Oct. 2020.
- [59] Z.Abdullah *et al.*, “A hybrid relay and intelligent reflecting surface network and its ergodic performance analysis,” *IEEE Wireless Commun. Lett.*, vol. 9, no. 10, pp. 1653–1657, June 2020.
- [60] L. Yang *et al.*, “On the performance of RIS-assisted dual-hop UAV communication systems,” *IEEE Trans. Veh. Technol.*, vol. 69, no. 9, pp. 10 385–10 390, June 2020.
- [61] H. Yang *et al.*, “Age of information in random access networks: A spatiotemporal study,” in *Proc. IEEE GLOBECOM*, 2020, pp. 1–6.
- [62] M. Samir *et al.*, “Leveraging UAVs for coverage in cell-free vehicular networks: A deep reinforcement learning approach,” *IEEE Trans. Mobile Comput.*, pp. 1–1, April 2020.
- [63] ———, “Trajectory planning of multiple dronecells in vehicular networks: A reinforcement learning approach,” *IEEE Networking Lett.*, pp. 1–1, Jan. 2020.
- [64] ———, “Joint optimization of UAV trajectory and radio resource allocation for drive-thru vehicular networks,” in *Proc. IEEE WCNC*, April 2019, pp. 1–6.
- [65] AMR, “Global connected car market opportunities and forecasts, 2018 - 2025,” [online] available at : <https://www.alliedmarketresearch.com/connected-car-market>, 2018.
- [66] F. Tariq *et al.*, “A speculative study on 6G,” *IEEE Wireless Commun.*, vol. 27, no. 4, pp. 118–125, Aug. 2020.

- [67] M. Mozaffari *et al.*, “A tutorial on UAVs for wireless networks: Applications, challenges, and open problems,” *IEEE Commun. Surveys Tuts.*, pp. 1–1, March 2019.
- [68] I. Dalmaso *et al.*, “WiMAX networks for emergency management based on UAVs,” in *Proc. IEEE ESTEL*, Oct. 2012.
- [69] M. Mozaffari *et al.*, “Drone small cells in the clouds: Design, deployment and performance analysis,” in *Proc. IEEE GLOBECOM*, Dec. 2015.
- [70] J. Lyu *et al.*, “Placement optimization of UAV-mounted mobile base stations,” *IEEE Commun. Lett.*, vol. 21, no. 3, pp. 604–607, March 2017.
- [71] R. Bor-Yaliniz *et al.*, “Efficient 3-D placement of an aerial base station in next generation cellular networks,” in *Proc. IEEE ICC*, May 2016.
- [72] M. Samir *et al.*, “Trajectory planning and resource allocation of multiple UAVs for data delivery in vehicular networks,” *IEEE Networking Lett.*, pp. 1–1, May 2019.
- [73] X. Liu *et al.*, “Trajectory design and power control for multi-UAV assisted wireless networks: A machine learning approach,” *IEEE Trans. Veh. Technol.*, pp. 1–1, May 2019.
- [74] C. Liu *et al.*, “Energy-efficient UAV control for effective and fair communication coverage: A deep reinforcement learning approach,” *IEEE J. Sel. Areas Commun.*, vol. 36, no. 9, pp. 2059–2070, Sep. 2018.
- [75] ———, “Distributed energy-efficient multi-UAV navigation for long-term communication coverage by deep reinforcement learning,” *IEEE Trans. Mobile Comput.*, pp. 1–1, March 2019.
- [76] ———, “Energy-efficient distributed mobile crowd sensing: A deep learning approach,” *IEEE J. Sel. Areas Commun.*, pp. 1–1, March 2019.
- [77] W. Powell, “What you should know about approximate dynamic programming,” *Naval Res. Logistics*, vol. 56, no. 3, pp. 239–249, Apr. 2009.
- [78] Lillicrap *et al.*, “Continuous control with deep reinforcement learning,” *In Int. Conf. on Learning Representations*, 2016.

- [79] A. Reis *et al.*, “Deploying roadside units in sparse vehicular networks: What really works and what does not,” *IEEE Trans. Veh. Technol.*, vol. 63, no. 6, pp. 2794–2806, July 2014.
- [80] N. Wisitpongphan *et al.*, “Routing in sparse vehicular ad hoc wireless networks,” *IEEE J. Sel. Areas Commun.*, vol. 25, no. 8, pp. 1538–1556, Oct. 2007.
- [81] Z. Zhang *et al.*, “Stochastic characterization of information propagation process in vehicular ad hoc networks,” *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 1, pp. 122–135, Feb. 2014.
- [82] R. Atallah *et al.*, “Multihop V2I communications: A feasibility study, modeling, and performance analysis,” *IEEE Trans. Veh. Technol.*, vol. 66, no. 3, pp. 2801–2810, March 2017.
- [83] Y. Zeng *et al.*, “Energy-efficient UAV communication with trajectory optimization,” *IEEE Trans. Wireless Commun.*, vol. 16, no. 6, pp. 3747–3760, June 2017.
- [84] Y. Chen, B. Ai, Y. Niu, K. Guan, and Z. Han, “Resource allocation for device-to-device communications underlying heterogeneous cellular networks using coalitional games,” *IEEE Trans. Wireless Commun.*, vol. 17, no. 6, pp. 4163–4176, June 2018.
- [85] Y. Zeng *et al.*, “Energy minimization for wireless communication with rotary-wing UAV,” *IEEE Trans. Wireless Commun.*, vol. 18, no. 4, pp. 2329–2345, April 2019.
- [86] A. Khuwaja *et al.*, “A survey of channel modeling for UAV communications,” *IEEE Commun. Surveys Tuts.*, vol. 20, no. 4, pp. 2804–2821, July 2018.
- [87] V. Mnih *et al.*, “Playing atari with deep reinforcement learning,” 2013. [Online]. Available: <https://arxiv.org/abs/1312.5602>
- [88] K. Abboud *et al.*, “Stochastic analysis of a single-hop communication link in vehicular Ad-Hoc networks,” *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 5, pp. 2297–2307, Oct. 2014.
- [89] C. Liu *et al.*, “Distributed and energy-efficient mobile crowdsensing with charging stations by deep reinforcement learning,” *IEEE Trans. Mobile Comput.*, pp. 1–1, Aug. 2019.
- [90] P. Henderson *et al.*, “Deep reinforcement learning that matters,” *AAAI Conference on Artificial Intelligence*, 2018.

- [91] M. Samir *et al.*, “Age of information aware trajectory planning of UAVs in intelligent transportation systems: A deep learning approach,” *IEEE Trans. Veh. Technol.*, vol. 69, no. 11, pp. 12 382–12 395, Nov 2020.
- [92] T. Gandhi *et al.*, “Pedestrian protection systems: Issues, survey, and challenges,” *IEEE Trans. Veh. Technol.*, vol. 8, no. 3, pp. 413–430, Sep. 2007.
- [93] I. Kadota *et al.*, “Optimizing age of information in wireless networks with throughput constraints,” in *Proc. IEEE INFOCOM*, April 2018, pp. 1844–1852.
- [94] —, “Minimizing the age of information in broadcast wireless networks,” in *Proc. IEEE Allerton*, Sep. 2016, pp. 844–851.
- [95] Sutton *et al.*, “Reinforcement learning: An introduction,” in *MIT press*, 2018.
- [96] M. Khabazian *et al.*, “A performance modeling of connectivity in vehicular Ad Hoc networks,” *IEEE Trans. Veh. Technol.*, vol. 57, no. 4, pp. 2440–2450, July 2008.
- [97] J. Xiong *et al.*, “Parametrized deep qnetworks learning: Reinforcement learning with discretecontinuous hybrid action space,” 2018. [Online]. Available: <https://arxiv.org/pdf/1810.06394>
- [98] Zhen Liu, “Scheduling of random task graphs on parallel processors,” in *Proc. IEEE MAS-COTS*, 1995, pp. 143–147.
- [99] P. Bhattacharya *et al.*, “Optimal scheduling with strict deadlines,” *IEEE Trans. Autom. Control*, vol. 34, no. 7, pp. 721–728, July 1989.
- [100] A. Ganti *et al.*, “Optimal transmission scheduling in symmetric communication models with intermittent connectivity,” *IEEE Trans. Inf. Theory*, vol. 53, no. 3, pp. 998–1008, March 2007.
- [101] M. Khabazian *et al.*, “Performance modeling of message dissemination in vehicular ad hoc networks with priority,” *IEEE J. Sel. Areas Commun.*, vol. 29, no. 1, pp. 61–71, Jan. 2011.

- [102] T. Nguyen *et al.*, “A novel cooperative non-orthogonal multiple access (NOMA) in wireless backhaul two-tier hetnets,” *IEEE Trans. Wireless Commun.*, vol. 17, no. 7, pp. 4873–4887, July 2018.
- [103] M. Grant, S. Boyd, and Y. Ye, “CVX: matlab software for disciplined convex programming. version 2.1,” [online] available: <http://cvxr.com/cvx>, 2016.
- [104] P. Nesbit *et al.*, “Reported UAV incidents in canada: analysis and potential solutions,” *Journal of Unmanned Vehicle Systems*, pp. 51–61, March 2017.
- [105] BBC, “Saudi arabia oil facilities ablaze after drone strikes,” 2019. [Online]. Available: <https://www.bbc.com/news/world-middle-east-49699429>