

University of New Mexico

## UNM Digital Repository

---

Electrical and Computer Engineering ETDs

Engineering ETDs

---

Fall 11-10-2020

# Artificial Intelligence Enabled Distributed Edge Computing for Internet of Things Applications

Georgios Fragkos

*University of New Mexico*

Follow this and additional works at: [https://digitalrepository.unm.edu/ece\\_etds](https://digitalrepository.unm.edu/ece_etds)



Part of the [Electrical and Computer Engineering Commons](#)

---

### Recommended Citation

Fragkos, Georgios. "Artificial Intelligence Enabled Distributed Edge Computing for Internet of Things Applications." (2020). [https://digitalrepository.unm.edu/ece\\_etds/494](https://digitalrepository.unm.edu/ece_etds/494)

This Thesis is brought to you for free and open access by the Engineering ETDs at UNM Digital Repository. It has been accepted for inclusion in Electrical and Computer Engineering ETDs by an authorized administrator of UNM Digital Repository. For more information, please contact [disc@unm.edu](mailto:disc@unm.edu).

---

*Candidate*

---

*Department*

This thesis is approved, and it is acceptable in quality and form for publication:

*Approved by the Thesis Committee:*

*E. Tsiropoulou*     *Eirini Eleni Tsiropoulou* , Chairperson

---

*Xiang Sun*

---

*Marios Pattichis*

---

---

---

---

---

---

---

---

---

---

# Artificial Intelligence Enabled Distributed Edge Computing for Internet of Things Applications

by

**Georgios Fragkos**

M.S., National Technical University of Athens, 2018

THESIS

Submitted in Partial Fulfillment of the  
Requirements for the Degree of

Master of Science  
Computer Engineering

The University of New Mexico

Albuquerque, New Mexico

December, 2020

# Dedication

*To my parents, Kostas and Gioula, and brother John, who have constantly supported, loved, and encouraged me throughout my graduate degree.*

# Acknowledgments

I would like to thank my Ph.D. advisor, Dr. Eirini Eleni Tsiropoulou, for the opportunity that she provided me with to pursue my research interests in Reinforcement Learning and Game Theory, and being there for me when I needed guidance. I also want to acknowledge both Dr. Pattichis and Dr. Sun for being a part of my thesis committee.

I would also like to thank my friends from the PROTON lab for their help, especially when I was new to this type of research environment. I want to say thank you to all of my colleagues that helped me to overcome any obstacles that I came across during my research journey.

Finally, I want to express my gratitude towards my parents and brother for providing me with unfailing support and continuous encouragement throughout my years of study. This accomplishment would not have been possible without them. Thank you.

# Artificial Intelligence Enabled Distributed Edge Computing for Internet of Things Applications

by

**Georgios Fragkos**

M.S., National Technical University of Athens, 2018

M.S., Computer Engineering, University of New Mexico, 2020

## Abstract

Artificial Intelligence (AI) based techniques are typically used to model decision making in terms of strategies and mechanisms that can conclude to optimal payoffs for a number of interacting entities, often presenting competitive behaviors. In this thesis, an AI-enabled multi-access edge computing (MEC) framework is proposed, supported by computing-equipped Unmanned Aerial Vehicles (UAVs) to facilitate Internet of Things (IoT) applications. Initially, the problem of determining the IoT nodes optimal data offloading strategies to the UAV-mounted MEC servers, while accounting for the IoT nodes' communication and computation overhead, is formulated based on a game-theoretic model. The existence of at least one Pure Nash Equilibrium (PNE) point is shown by proving that the game is submodular. Furthermore, different operation points (i.e., offloading strategies) are obtained and studied, based either on the outcome of Best Response Dynamics (BRD) algorithm, or via alternative reinforcement learning approaches, such as gradient ascent, log-linear and Q-learning algorithms, which explore and learn the environment towards

determining the users' stable data offloading strategies. The respective outcomes and inherent features of these approaches are critically compared against each other, via modeling and simulation.

**This work has been published in:**

**G. Fragkos, E.E. Tsiropoulou, and S. Papavassiliou, "Artificial Intelligence Enabled Distributed Edge Computing for Internet of Things Applications," in IEEE International Conference on Distributed Computing in Sensor Systems, 2020**

# Contents

List of Figures	ix
List of Tables	xi
Glossary	xii
<b>1 Overview</b>	<b>1</b>
1.1 Introduction . . . . .	1
1.2 Related Work & Motivation . . . . .	4
1.3 Contributions . . . . .	10
1.4 Outline . . . . .	11
<b>2 AI-enabled Distributed Edge Computing System for IoT Applications</b>	<b>12</b>
2.1 Communication & Computation Overhead . . . . .	12
2.2 IoT Devices Utilities . . . . .	14
2.3 Game-Theoretic Edge Distributed Computing . . . . .	15



*Contents*

2.3.1	Problem Formulation . . . . .	15
2.3.2	Problem Solution . . . . .	17
2.3.3	Best Response Dynamics . . . . .	18
2.4	Reinforcement Learning-Enabled Distributed Edge Computing . . . . .	19
2.4.1	Gradient Ascent Learning . . . . .	20
2.4.2	Log-Linear Learning . . . . .	22
2.4.3	Q-Learning . . . . .	23
<b>3</b>	<b>Experiments</b>	<b>28</b>
3.1	Experiment Setup . . . . .	28
3.2	Pure Operation Performance . . . . .	29
3.3	Comparative Evaluation . . . . .	35
3.4	Discussion on Learning Methods Applicability . . . . .	38
<b>4</b>	<b>Conclusion and Future Works</b>	<b>41</b>
	<b>References</b>	<b>44</b>

# List of Figures

3.1	BRD Average Offloaded Data & Overhead . . . . .	29
3.2	BRD Social Welfare & Utility . . . . .	30
3.3	LRI Action Proabilities . . . . .	30
3.4	LRI Average Offloaded Data & Overhead . . . . .	31
3.5	LRI Social Welfare & Utility . . . . .	31
3.6	LRI Learning Parameter . . . . .	32
3.7	BLLL Social Welfare . . . . .	32
3.8	BLLL Average Utility . . . . .	33
3.9	BLLL Average Offloaded Data . . . . .	33
3.10	BLLL Average Total Overhead . . . . .	34
3.11	Q-Learning Social Welfare . . . . .	34
3.12	Q-Learning Average Utility . . . . .	35
3.13	Q-Learning Average Offloaded Data . . . . .	35
3.14	Q-Learning Average Total Overhead . . . . .	36

*List of Figures*

3.15	RL Social Welfare Comparison . . . . .	36
3.16	RL MSE Comparison . . . . .	37
3.17	RL Execution Time . . . . .	37
3.18	BLLL MSE for different number of actions . . . . .	38
3.19	BLLL Execution Time for different number of actions . . . . .	39

# List of Tables

3.1	RL Comparison Table . . . . .	38
-----	-------------------------------	----

# Glossary

$t$	Timeslot instance
$D$	Set of IoT devices where $D = \{1, \dots, d, \dots,  D \}$
$x, y$	Coordinates of area, $x[m] \times y[m]$
$\mathcal{T}_d^{(t)}$	Computation task of IoT device $d$
$I_d^{(t)}$	Total amount of data of IoT device's task's $\mathcal{T}_d^{(t)}$
$\phi_d^{(t)}$	Computation intensity of IoT device's task's $\mathcal{T}_d^{(t)}$
$a_{d,j}^{(t)}$	Percentage of the overall amount of the device's computation task's data
$A_d^{(t)}$	Set of offloading strategies where $A_d^{(t)} = \{a_{d,min}^{(t)}, \dots, a_{d,j}^{(t)}, \dots, a_{d,max}^{(t)}\}$
$R_d^{(t)}$	IoT device's $d$ uplink data rate to the UAV-mounted MEC server
$W$	System's bandwidth
$p_d^{(t)}$	Device's transmission power
$g_d^{(t)}$	Device's channel gain to communicate with the UAV
$\sigma_o^2$	Variance of the additive white gaussian noise

## Glossary

$F_{UAV}$	Computation capability of the UAV-mounted MEC server
$B_{UAV}$	Operational threshold of the UAV-mounted MEC server
$O_{time,d}^{(t)}$	Time overhead experienced by the IoT device $d$
$O_{energy,d}^{(t)}$	Energy overhead experienced by the IoT device $d$
$T$	Duration of timeslot $t$
$e_d^{(t)}$	Energy availability of the IoT device $d$
$O_d^{(t)}$	Total experienced normalized overhead of IoT device $d$
$O_{ij}^{tr,e}$	Transmission energy overhead
$U_d^{(t)}$	Utility Function
$b, c$	Configurable parameters in the utility function
$G$	Non-cooperative game of the IoT devices
$\mathbf{a}_{-d,j}^{(t)}$	Data offloading strategy vector of all IoT devices excluding $d$
$a_{d,j}^{(t)*}$	Optimal data offloading strategy of IoT device $d$
$\mathbf{a}^{(t)*}$	Data offloading vector of the IoT devices at PNE point
$\mathbf{a}_{-d,j}^{(t)*}$	Data offloading vector of the IoT devices at PNE point excluding $d$
$BR_d(\mathbf{a}_{-d,j}^{(t)*})$	IoT Device's $d$ best offloading response
$\mathbf{P}_d^{(ite)}$	Action probability vector of IoT device $d$
$P_{thres}$	Threshold value of the action probability
$\eta$	LRI Learning parameter

## Glossary

$[\hat{U}_d^{(t)}]^{(ite)}$	Normalized utility function
$\beta$	BLLL Learning parameter
$Q_d^{(ite)}(\mathbf{a})$	Action values vector of IoT device $d$
$Q_{a_{d,j}}^{(ite)}$	Q-value of an offloading strategy for IoT device $d$
$\theta$	Q-learning learning parameter
$\epsilon$	Q-learning exploration parameter

# Chapter 1

## Overview

### 1.1 Introduction

The rapid deployment of Internet of Things (IoT) devices [1, 2], such as sensors, smartphones, autonomous vehicles, wearable smart devices, along with the recent advances in the Artificial Intelligence (AI) and Reinforcement Learning (RL) techniques [3], have paved the way to a future of using distributed edge computing to assist humans' everyday activities, in several domains such as transportation, healthcare, public safety and others [4-6]. The ubiquity of the IoT devices with enhanced sensing capabilities creates increasingly large streams of data that need to be collected and processed in an energy and time efficient manner.

Traditionally, Cloud-based solutions were utilized to deal with the computational, storage, and networking challenges imposed by the large streams of data. However, Cloud computing faces great challenges related to energy consumption, latency, and security, all of them being critical aspects for sensor-driven applications [7,8]. On the other hand, the emerging edge computing paradigm proposes shifting the pendulum away from the traditional Cloud computing model, towards a distributed infrastruc-



## *Chapter 1. Overview*

ture model at the edge of the network, by offering computational resources closer to the physical location of data producers/consumers [9–11].

Nevertheless, in order to fully unleash the autonomous decision-making capabilities of the edge devices and users, while exploiting the distributed edge computing capabilities, there is an urgent need to push the AI frontiers to the system’s edge [12]. AI mechanisms allow 5G networks to be predictive and proactive, which is essential in making the 5G vision conceivable [13]. Nowadays, AI has extended its domain and strength, beyond the traditional machine learning approaches, by being founded on multi-disciplinary techniques, such as control theory, computationally light reinforcement learning techniques, game theory, optimization theory, and meta-heuristics [14].

Game Theory is a mathematical tool that helps us understand the phenomena that we observe in cases where multiple decision makers with competing interests interact among each other aiming at determining a stable mode of operation [15]. Specifically, game theory was first proposed for the economics domain [16] and was originally examining the ways in which interacting choices of economic entities produce outcomes with respect to the utilities of those entities [17]. However, it has also been deployed in multiple domains such as Psychology [18], Political Science [19], Sociology [20] and many others. In the recent years, game theory has also started to be extensively utilized in the field of Computer Science, where, in the majority of the cases, the respective entities act in a selfish way aiming at maximizing their own objective. Accordingly, there have been published many research works [21–25] which exploit the benefits of game theory to solve distributed decision-making problems, networks-related problems [26], resource management and orchestration challenges [27], security and privacy concerns [28], or even Big Data processing-related problems.

Except for Game Theory, another important feature that enables the research community to take advantage of the AI’s power is Reinforcement Learning (RL).

## *Chapter 1. Overview*

RL is a subset of Machine Learning, where the agents learn to achieve a goal, i.e., maximize the expected cumulative future reward, in an uncertain and potentially complex environment which demonstrates dynamic variability and stochasticity [29–32]. Since the agent’s actions have short and long term consequences, the agent needs to gain some understanding of the complex effects its actions have on the environment and it should find the perfect balance between exploration (exploring potential hypotheses in terms of choosing its actions) and exploitation (exploiting limited knowledge about what is already learned should work in a satisfactory way). The main difference between RL and the traditional Supervised Learning [33] is that there is no need for labeled input/output pairs and that RL focuses on finding a balance between exploration and exploitation, achieving thus near-optimal solutions. The latter observation reveals that the reinforcement learning techniques can be applied in a real-time decision-making problem, which is an important component within the dynamically changing networking and communications environments. The selected actions of the agents transition the current state of the environment to the next state and finally the agents experience a reward as a feedback from the environment.

As a result, Reinforcement Learning has paved the way towards a lightweight AI-enabled future and becomes a core component of the AI vision to support the distributed decision making and emulate the humans behavior through a machine type of representation and actions [34, 35]. Furthermore, the lower computational complexity of the reinforcement learning approaches, in terms of data classification and offline processing, compared to the supervised learning approaches, have provided the enhanced flexibility to the implementation of the decision-making problems in a real-time or close to real-time manner.

Motivated by the aforementioned observations and arguments, in this paper, we propose an artificial intelligence enabled multi-access edge computing (MEC)

framework, supported by computing-equipped Unmanned Aerial Vehicles (UAVs) to facilitate IoT applications. The key problem at hand is to properly explore and learn the environment and the interdependence among the IoT nodes actions, so that to determine their optimal data offloading strategies to an UAV-mounted MEC server, while accounting for the IoT nodes' communication and computation overhead.

## **1.2 Related Work & Motivation**

Mutli-access Edge Computing (MEC) [36] is constantly gaining ground in distributed computing, since traditional Cloud computing suffers from high energy consumption and latency due to high volume data streams. In [37], the authors study the problem of users' data offloading along to a MEC server as well as the interference management problem in wireless cellular networks by solving the joint optimization of the computation offloading decision, physical resource block allocation, and MEC computation resource allocation. Moreover, the authors in [38] present the problem of data offloading to a MEC server as a non-cooperative game among vehicles aiming at minimizing the latency of data offloading and also the existence of a Nash Equilibrium is proven. In [39], the joint problem of MEC server selection by the end-users and their optimal data offloading, along with the optimal price setting by the MEC servers is examined in a multi-MEC servers and multi-users environment. For that reason they utilize game Theory and reinforcement learning, and more specifically the theory of Stochastic Learning Automata (SLA).

Distributed edge computing has been immensely supported by the adoption of UAV-mounted MEC servers [40], due to the UAVs' unique characteristics, i.e., fast, flexible, and effortless deployment, mobility, maneuverability, line-of-sight communication, etc. The problem of minimizing the IoT devices' communication and computation energy consumption and the UAVs' flying energy utilization is studied in [41],

## Chapter 1. Overview

by jointly optimizing the devices' data offloading, transmission power, and the UAVs' trajectory. In [42], the problem of partial data offloading from the IoT devices to ground or UAV-mounted MEC servers is studied in order the devices to satisfy their minimum Quality of Service (QoS) prerequisites, by adopting the novel concept of Satisfaction Equilibrium. In [43], the authors target at UAVs energy-efficiency, where the authors aim at extending the UAVs' battery lifetime by jointly optimizing their hovering time, and the devices' scheduling and data offloading, while considering the constraints of the UAVs' computation capability and the devices' QoS constraints. A similar problem is studied in [44] by exploiting the uplink and downlink communication among the devices and the UAVs in terms of data offloading/receiving data respectively, while guaranteeing the energy efficient operation of the overall system.

In [45], the problem of jointly optimizing the devices' association, transmission power, and data offloading to the UAVs, as well as the UAVs' trajectory is studied, aiming at minimizing the overall power consumption in the system. In [46], the authors introduce artificial intelligence into the UAVs data offloading process in a multi-server MEC environment, by utilizing concepts from game theory and reinforcement learning. They formulate a non-cooperative game among the UAVs, which is proven to be submodular and as a result a Pure Nash Equilibrium (PNE) exists. In order to approach the PNE they utilize a Best Response Dynamics approach as well as two different reinforcement learning algorithms.

A techno-economics approach is presented in [47], where the UAVs charge fees the users for the computation services that they offer to them. Also, the UAVs charge their battery over a microwave station and the authors target at maximizing the UAVs' utility by optimizing their trajectories and the data offloading process. Following the notion of the techno-economic study of the UAV-assisted MEC system, the authors in [48] study the end-users behavioral characteristics in terms of their risks in the task offloading process and they propose a novel pricing mechanism in

## Chapter 1. Overview

order to introduce a more social behavior to the users with respect to competing for the UAV-mounted MEC servers' computation resources.

In [49], the UAVs act as cache and edge computing nodes, and two sequentially solved optimization problems are considered, to minimize the communication and computation delay and maximize the energy efficiency of the system. In [50], the UAVs act both as MEC servers and as wireless power transfer nodes charging the IoT devices. The problem of maximizing the UAVs' computation rate is examined under the UAVs' energy provision and speed constraints. This problem has been extended in [51] by studying the minimization of the overall system's energy consumption by jointly optimizing the devices' association to the UAVs, the UAVs' flying time, and their wireless powering duration.

The authors in [52] study a MEC environment, where a UAV is served by cellular ground base stations (GBSs) for computation offloading. Since they aim at minimizing the UAV's computation offloading scheduling time by optimizing its trajectory subject to the maximum speed constraint of the UAV and the computation capacity constraints at the GBSs, they propose an iterative algorithm based on Successive Convex Approximation (SCA) which obtains near-optimal solutions. In [53] and [54] the traditional problem where a set of ground users offload tasks to a UAV has been extended, since the authors examine an UAV-assisted MEC architecture where the UAV has a twofold role, i.e., contributing in users' task execution and acting as a relay node for offloading the users' received computation tasks to an access point (AP). The non-convex minimization of the weighted sum energy of both the users and the UAV is achieved using a centralized iterative algorithm. Similarly, in [55] a two-hop uplink communication for Space-Air-Ground Internet of Remote Things (SAG-IoRT) networks is studied, which is assisted with UAV relays in order to facilitate complete offloading of the terrestrial smart devices to satellites. They target at maximizing the whole system's energy efficiency by jointly optimizing the sub-

## *Chapter 1. Overview*

channel selection, uplink transmission power control and the UAV relay deployment. The authors in [56] introduce a UAV-enabled MEC system, where the UAVs act jointly as relay and data processing nodes to support the communication and computing demands of the ground devices. A joint optimization problem is formulated to minimize the service delay of the ground devices and the UAVs by determining the UAVs optimal position, the communication and computing resource allocation, and the devices' task splitting.

A centralized task offloading approach to the UAV-mounted and ground MEC servers is proposed in [57], where an intelligent centralized agent makes optimal decisions about the users' task offloading strategies via sensing the communication and computing conditions of the environment towards optimizing the users' Quality of Experience (QoE). An air-ground integrated MEC architecture is proposed in [58] consisting of both ground and UAV-mounted MEC servers. The authors highlight the benefits of the UAV-mounted MEC servers and the problem of opportunistic computational offloading is studied in order to determine the tasks that should be offloaded to the neighboring UAV clusters with sufficient computing resources, in order to increase the UAVs' lifetime and decrease the overall computation time.

Additionally, many research papers deal with the problem of data offloading among a cluster of UAVs. More specifically, in [59] the authors propose the Fog Computing aided Swarm of Drones (FCSD), where a drone will have a computation task to execute and will partially offload its data to nearby drones in order to perform the computations, thus acting as fog nodes. The scope of this research work is to minimize the energy consumption of the FCSD system subject to the reliability and latency constraints by introducing and utilizing an iterative distributed algorithm based on the Proximal Jacobi method. As far as [60] is concerned, a network of capacitated UAV-mounted cloudlets (NUMC) covering a region is considered, where each UAV is endowed with limited computational resources and a restricted capacity

## *Chapter 1. Overview*

providing edge computing services to IoT users in that region. The UAVs perform binary offloading to other UAVs and as a consequence the authors formulate an exact potential game in order to capture the UAVs' competitive behavior in terms of minimizing their energy consumption with respect to the QoS satisfaction of the IoT users' requirements. Moreover, the research work [61] proposes a task-scheduling algorithm based on reinforcement learning targeting at the collaboration of multiple UAV tasks within a UAV cluster. Specifically, the proposed algorithm enables the UAV to adjust its task strategy automatically using its calculation of task performance efficiency, while reinforcement learning has been deployed in order the UAVs to learn tasks according to real-time data and as a consequence to perform decision making regarding the channel allocation problem in a distributed manner.

Another important aspect of distributed edge computing in the recent bibliography is the ability of the ground users and the UAVs to harvest energy in order to boost the overall system's energy efficiency. In [62], the authors consider a multi-drone enabled data collection system for smart cities, where there are two kinds of UAVs, i.e., the Low Altitude Platforms (LAPs) which collect the data from the smart city and the High Altitude Platform (HAP) which provides energy to the LAPs using wireless energy beams. The scope of this paper is to minimize the total laser charging energy of the HAP using a novel search algorithm named Drones Travelling Algorithm (DTA). Furthermore, [63], [64] and [65] all introduce harvesting models in order the ground users to be able to perform computations locally and/or to transmit information to the UAVs using the harvested energy from the latter ones, aiming thus at the minimization of the users' energy consumption. Additionally, in [66] the authors propose an efficient energy and radio resource management framework based on intelligent power cognition of the Solar-powered UAVs (SUAVs), which can learn the environment including the spatial distributions of solar energy density, the channel state evolution, and the traffic patters of wireless communication applications in adaption to the environment changes. Thus, they utilize reinforcement learning in

## *Chapter 1. Overview*

order to maximize the total system throughput within the lifetime of the SUAVs, by optimizing the energy harvesting and resource allocation of the power cognitive SUAVs.

In [67] the authors focus on the UAVs' data offloading to a MEC server and thus, they formulate a multi-nature strategy non-cooperative game among the UAVs taking into consideration the energy consumption, time delay and computation cost. As a result, they prove the existence of a Pure Nash Equilibrium and propose a distributed algorithm to determine the UAVs' strategies at the PNE point. This research work is sufficiently extended in [68], where the authors also aim at minimizing a combination of energy overhead and delay for each UAV concurrently. Additionally, in [69] the authors examine a UAV-assisted crowd surveillance use case, where the UAVs acquire videos from cameras on the ground and they perform computation either on board or at the ground servers. The research work in [70] studies the joint optimization problem of the UAV's trajectory and radio resource allocation via a Successive Convex Approximation (SCA) technique, in order to maximize the number of served devices in terms of achievable uplink data rate. In [71], the UAV's time flight is minimized by optimizing its altitude, while jointly maximizing the number of offloaded bits by the ground devices.

However, despite the significant research work and advances achieved by the aforementioned research efforts, the problem of the IoT devices' distributed and autonomous decision-making with respect to their data offloading strategies, towards jointly optimizing their communication and computation overhead has not yet been fully exploited, especially under the the prism of artificial intelligence. In this thesis, a field of IoT devices is considered supporting latency and energy sensitive IoT applications. Accordingly each IoT device has the option to execute its computation task either locally or offload part of it to a UAV-mounted MEC server, by considering the joint optimization of the involved communication and computation overhead.



The focus of this paper is placed on the design of an artificial intelligence-enabled framework that drives the strategic decision of optimal data offloading to the available UAV-mounted MEC server, founded on the power and principles of *Game Theory* and *Reinforcement Learning*.

### 1.3 Contributions

The key technical contributions of this thesis are summarized as follows. First of all, we model and formulate the IoT devices' communication, computation and energy overhead due to data offloading, while based on this, the utility of each device by offloading and processing its computation task's data to the UAV-mounted MEC server is reflected in representative functions.

Moreover, in order to capture the competitive behavior of the IoT devices, we formulate a non-cooperative game among them aiming at maximizing their own utility at every timeslot, while considering at the same time the experienced communication and computation time overhead, from offloading and processing their data at the UAV. As a consequence, this process enables the devices to learn from history, scrutinize the performance of other nodes, and adjust their behavior accordingly. We also show the existence of at least one Pure Nash Equilibrium (PNE) point, by proving that the game is submodular. Thus, we introduce a best response dynamics approach which converges to a PNE.

Additionally, the proposed distributed edge computing decision making is enhanced by an artificial intelligent element, realized by various reinforcement learning algorithms. The latter enable the IoT devices to learn their environment and make stable decisions regarding their data offloading strategies. A set of reinforcement learning algorithms is examined, including gradient ascent, log-linear, and Q-learning algorithms.

Finally, we provide detailed numerical results in order to evaluate the effectiveness of the proposed artificial intelligence-enabled distributed edge computing framework, while at the same time a comparative study highlights the drawbacks and benefits of the examined reinforcement learning algorithms.

## **1.4 Outline**

The rest of this thesis is organized as follows. In Section 2.1 we present the formulated IoT devices' communication and computation overhead, while in Section 2.2 we model the experienced utility of each device. Furthermore, in Section 2.3 we adduce out proposed game-theoretic edge distributed computing framework, by first formulating a non-cooperative game among the IoT devices and afterwards proving that there is at least one Pure Nash Equilibrium (PNE) in Sections 2.3.1 and 2.3.2 respectively. Thus, in Section 2.3.3 we introduce a best response dynamics method in order the IoT devices to converge to the aforementioned PNE. In Section 2.4 we introduce three different families of Reinforcement Learning algorithms, aiming at the IoT devices converging to a PNE in an autonomous and distributed manner. Specifically, in Section 2.4.1 we present the Linear Reward Inaction (LRI) algorithm, while in Sections 2.4.2 and 2.4.3 the Binary Log Linear (BLLL) and the stateless Q-Learning correspondingly. Finally, a detailed numerical and comparative performance evaluation results between the different proposed approaches are provided in Chapter 3, while Chapter 4 concludes this master's thesis.

# Chapter 2

## AI-enabled Distributed Edge Computing System for IoT Applications

### 2.1 Communication & Computation Overhead

A distributed edge computing system is considered consisting of a set of IoT devices  $D = \{1, \dots, d, \dots, |D|\}$  spread in an area  $x[m] \times y[m]$  and a UAV-mounted MEC server hovering above the area. Each IoT device has a computation task  $\mathcal{T}_d^{(t)}$  to be completed at timeslot  $t$ , which is defined as  $\mathcal{T}_d^{(t)} = (I_d^{(t)}, \phi_d^{(t)})$ , where  $I_d^{(t)}$  [bits] denotes the total amount of data of the IoT device's computation task, and the parameter  $\phi_d^{(t)} [\frac{CPU-Cycles}{bit}]$  represents the computation intensity of the device's task (i.e., a higher value of  $\phi_d^{(t)}$  expresses a more computing demanding application). At each timeslot  $t$ , each IoT device offloads part of its computation task's data to the UAV-mounted MEC server for further processing, while aiming at minimizing its experienced communication and computation latency and energy

cost. The IoT device's  $d$  set of data offloading strategies at timeslot  $t$  is denoted as  $A_d^{(t)} = \{a_{d,min}^{(t)}, \dots, a_{d,j}^{(t)}, \dots, a_{d,max}^{(t)}\}$ , where  $a_{d,j}^{(t)} \in [0, 1]$  is a percentage of the overall amount of the device's computation task's data.

Moreover, a non-orthogonal multiple access (NOMA)-based wireless communication environment is considered to enable each IoT device to offload its computation task's data  $a_{d,j}^{(t)} \cdot I_d^{(t)}$  [bits] to the UAV at each timeslot  $t$ . Also, the Successive Interference Cancellation (SIC) technique is implemented at the UAV to improve the interference management in the congested IoT environment [72]. Each IoT device's  $d$  uplink data rate to the UAV-mounted MEC server at each timeslot  $t$  is calculated through the Shannon's formula, as follows.

$$R_d^{(t)} = W \cdot \log\left(1 + \frac{p_d^{(t)} \cdot g_d^{(t)}}{\sigma_o^2 + \sum_{d' \geq d+1}^{|D|} p_{d'}^{(t)} \cdot g_{d'}^{(t)}}\right) \quad (2.1)$$

where  $W$  [Hz] is the system's bandwidth,  $p_d^{(t)}$  is the device's transmission power, and  $g_d^{(t)}$  is the device's channel gain to communicate with the UAV at the timeslot  $t$ . Each device's transmission power is considered fixed in the following analysis and its absolute value depends on its hardware characteristics. Also, following the NOMA and SIC principles [73], without loss of generality, we consider  $g_{|D|}^{(t)} \leq \dots \leq g_d^{(t)} \leq \dots \leq g_1^{(t)}$ , thus, the interference that the IoT device  $d$  experiences is  $\sigma_o^2 + \sum_{d' \geq d+1}^{|D|} p_{d'}^{(t)} \cdot g_{d'}^{(t)}$ , where  $\sigma_o^2$  is the variance of the Additive White Gaussian Noise [74].

The UAV-mounted MEC server is assumed to have a computation capability  $F_{UAV}[\frac{CPU-Cycles}{sec}]$  that is shared among the IoT devices to process their offloaded data. Also, the UAV can process in parallel a total amount of data  $B_{UAV}$  [bits] at each timeslot. Based on the above, the time overhead that the IoT device  $d$  experiences at timeslot  $t$  by offloading  $a_{d,j}^{(t)} \cdot I_d^{(t)}$  is given as follows [75]:

$$O_{time,d}^{(t)} = \frac{a_{d,j}^{(t)} \cdot I_d^{(t)}}{R_d^{(t)}} + \frac{\phi_d^{(t)} \cdot a_{d,j}^{(t)} \cdot I_d^{(t)}}{\sum_{d' \neq d} a_{d',j'}^{(t)} \cdot I_{d'}^{(t)}} \cdot F_{UAV} \quad (2.2)$$

The first term of Eq.2.2 represents the communication time overhead that the IoT device experiences to offload its data to the UAV, while the second term captures the experienced computation time overhead. Also, as it is observed by the denominator of the second term in Eq.2.2, each IoT device exploits only a portion of the UAV's computation capability, as the latter is shared in a fair manner among the IoT devices with respect to how many bits they offloaded to the UAV.

Furthermore, the energy overhead that each IoT device experiences by offloading its computation task's data to the UAV at timeslot  $t$  is given as follows.

$$O_{energy,d}^{(t)} = \frac{a_{d,j}^{(t)} \cdot I_d^{(t)}}{R_d^{(t)}} \cdot p_d^{(t)} \quad (2.3)$$

The duration of a timeslot  $t$  is assumed  $T[sec]$  and the energy availability of an IoT device  $d$  at timeslot  $t$  is  $e_d^{(t)}[J]$ . Based on Eq.2.2, 2.3, the total normalized overhead that the IoT device  $d$  experiences at timeslot  $t$  is given as follows, which is the summation of the corresponding time and energy experienced overheads.

$$O_d^{(t)} = \frac{O_{time,d}^{(t)}}{T} + \frac{O_{energy,d}^{(t)}}{e_d^{(t)}} \quad (2.4)$$

## 2.2 IoT Devices Utilities

In the introduced artificial intelligence-enabled distributed edge computing framework each IoT device perceives a satisfaction by processing its data to the UAV-mounted MEC server, as well as a cost due to the time and energy overhead that it experiences. Moreover, each IoT device's experienced satisfaction and cost are dynamically interdependent with the data offloading strategies of the rest of the devices in the examined system. Thus, a holistic utility function is introduced for each IoT device to capture its perceived satisfaction and cost of processing its computation

task in the considered distributed edge computing system, as follows.

$$U_d^{(t)}(a_{d,j}^{(t)}, \mathbf{a}_{-d,j}^{(t)}) = b \cdot e^{\frac{a_{d,j}^{(t)}}{\sum_{\forall d' \neq d, d' \in D} a_{d',j'}^{(t)}}} - c \cdot e^{O_d^{(t)}} \quad (2.5)$$

where  $\mathbf{a}_{-d,j}^{(t)}$  is the data offloading strategy vector of all the devices residing in the examined system except for the IoT device  $d$ . Also, the weights  $b, c \in [0, 1]$  are configurable parameters representing how much the IoT device weighs the satisfaction that it receives by processing its data at the UAV (first term of Eq.2.5), as compared to the corresponding cost to perform this action (second term of Eq.2.5). Moreover, given that small changes in the devices' data offloading strategies can dramatically influence the stable operation of the distributed edge computing system due to the large number of devices, we have adopted the exponential form to capture the devices' satisfaction and cost tradeoffs and trends in Eq.2.5.

## 2.3 Game-Theoretic Edge Distributed Computing

In this section, we cast the IoT devices' distributed data offloading problem into the analytical framework of non-cooperative game theory. Initially, the non-cooperative data offloading game among the IoT devices is formulated, while subsequently an analytical solution is provided to determine a Pure Nash Equilibrium point of the game.

### 2.3.1 Problem Formulation

Each IoT device aims at maximizing its perceived utility, as expressed in Eq.2.5, at each timeslot in order to improve its perceived benefit from offloading and processing its data at the UAV-mounted MEC server, while mitigating its personal cost, as expressed by its experienced overhead (Eq.2.4). Thus, the corresponding optimization

problem for each IoT device, is expressed as the maximization of each IoT device's utility, as follows.

$$\begin{aligned} \max U_d^{(t)}(a_{d,j}^{(t)}, \mathbf{a}_{-d,j}^{(t)}) &= b \cdot e^{\forall d' \neq d, d' \in D} \frac{a_{d,j}^{(t)}}{\sum_{d',j'} a_{d',j'}^{(t)}} - c \cdot e^{O_d^{(t)}} \\ \text{s.t. } a_{d,j}^{(t)} &\in A_d^{(t)} \end{aligned} \quad (2.6)$$

Based on the maximization problem in Eq.2.6, we observe that the IoT devices' data offloading strategies are interdependent, and the devices demonstrate competitive behavior in terms of exploiting the UAV's computing capabilities. Thus, the utility maximization problem in Eq.2.6 is confronted as a non-cooperative game among the IoT devices. Let  $G = [D, \{A_d^{(t)}\}_{d \in D}, \{U_d^{(t)}\}_{d \in D}]$  denote the Distributed Data Offloading (DDO) game played among the IoT device's at each timeslot  $t$ , where as mentioned before  $D$  is the set of IoT devices,  $A_d^{(t)}$  is the data offloading strategy set of each device  $d \in D$ , and  $U_d^{(t)}$  denotes the device's utility.

The solution of the DDO game should determine an equilibrium point, where the IoT devices have maximized their perceived utility by selecting their optimal data offloading strategy  $a_{d,j}^{(t)*}$ . If the DDO game has a feasible PNE point, then at that point, no device has the incentive to unilaterally change its equilibrium data offloading strategy  $a_{d,j}^{(t)*}$ , given the strategies of the rest of the devices, as it cannot further improve its perceived utility. More precisely, the PNE of the non-cooperative DDO game is defined as follows.

**Definition 1. (Pure Nash Equilibrium)** *The data offloading vector  $\mathbf{a}^{(t)*} = (a_{1,j'}^{(t)*}, \dots, a_{|D|,j'}^{(t)*}), a_{d,j}^{(t)*} \in A_d^{(t)}$ , is a PNE of the DDO game if for every IoT device  $d$  the following condition holds true:  $U_d^{(t)}(a_{d,j}^{(t)*}, \mathbf{a}_{-d,j}^{(t)*}) \geq U_d^{(t)}(a_{d,j}^{(t)}, \mathbf{a}_{-d,j}^{(t)*})$  for all  $a_{d,j}^{(t)} \in A_d^{(t)}$ .*

Based on Definition 1, we conclude that the existence of a PNE in the DDO game guarantees the stable operation of the distributed edge computing system, while the

IoT devices maximize their perceived utility. On the other hand, if the DDO game does not have at least one PNE, that is translated to an unsteady and unstable state of the examined system.

### 2.3.2 Problem Solution

The theory of S-modular games is adopted in order to show the existence of at least one PNE for the DDO game [26, 76]. The basic intuition of the submodular games is that an increase in one's player's action for given actions of rivals, reinforces the desire of all players to decrease their actions because of strategic complementarity. S-modular games have gained great attention in resource allocation problems in wireless networks [77–80] due to: a) Pure Nash Equilibrium existence in S-modular games can be proved, b) if each player initially adopts his lowest strategy or his largest strategy, then he converges monotonically to an equilibrium, which depends on the initial state and finally c) if the S-modular game has a unique Nash Equilibrium, then it is dominance solvable and learning rules will converge to it, such as best response dynamics. Specifically, we show that the DDO game is submodular, which means that when an IoT device tends to offload a large amount of data to the UAV-mounted MEC server, the rest of the devices follow the exact opposite philosophy, i.e., they become more conservative in terms of their data offloading, as the MEC server is congested with tasks. Thus, in general a submodular game is characterized by strategic substitutes and has at least one PNE [26], [77]. Considering the DDO game with strategy space  $A_d^{(t)}$ , we can prove the following theorem.

**Theorem 1. (*Submodular Game*)** *The DDO game  $G = [D, \{A_d^{(t)}\}_{d \in D}, \{U_d^{(t)}\}_{d \in D}]$  is submodular of for all  $d \in D$  the following conditions hold true:*

- (i)  $\forall d \in D, A_d^{(t)}$  is a compact subset of the Euclidean space.



(ii)  $U_d^{(t)}$  is smooth in  $A_d^{(t)}$  and has non-increasing differences, i.e.,  $\frac{\partial^2 U_d^{(t)}}{\partial a_{d,j}^{(t)} \cdot \partial a_{d',j'}^{(t)}} \leq 0, \forall d, d' \in D, d \neq d', \forall j, j'$ .

*Proof.* Towards proving that the DDO game is submodular, we consider that the IoT device can partition its task in any feasible set of data and offload them to the UAV-mounted MEC server. Thus, the strategy space  $A_d^{(t)} = (0, 1]$  is continuous and a compact subset of the Euclidean space and  $U_d^{(t)}$  is a smooth function. Also we

have:  $\frac{\partial^2 U_d^{(t)}}{\partial a_{d,j}^{(t)} \cdot \partial a_{d',j'}^{(t)}} = b \cdot \lambda - c \cdot \mu$  where we set  $\lambda = e^{\frac{a_{d,j}^{(t)}}{\sum_{\forall d' \neq d, d' \in D} a_{d',j'}^{(t)}}} \cdot \left( \frac{-1}{\left( \sum_{\forall d' \neq d, d' \in D} a_{d',j'}^{(t)} \right)^2} + \frac{-1}{\left( \sum_{\forall d' \neq d, d' \in D} a_{d',j'}^{(t)} \right)^3} \cdot a_{d,j}^{(t)} \right)$  and  $\mu = e^{O_d^{(t)}} \cdot \left( \frac{\phi_d^{(t)} \cdot I_d^{(t)} \cdot \frac{I_{d'}^{(t)}}{B_{UAV}}}{\sum_{d' \neq d} a_{d',j'}^{(t)} \cdot I_{d'}^{(t)}} \right) \cdot (1 + O_d^{(t)})$ . Thus, we

observe that  $\lambda < 0$  and  $\mu > 0$ . Therefore, we conclude that  $\frac{\partial^2 U_d^{(t)}}{\partial a_{d,j}^{(t)} \cdot \partial a_{d',j'}^{(t)}} < 0$  and the DDO game is submodular.  $\blacksquare$

Consequently, taking into account that a submodular game has a non-empty set of Pure Nash Equilibrium points [26], [77], we conclude that the DDO game has at least one PNE  $\mathbf{a}^{(t)*} = (a_{1,j'}^{(t)*}, \dots, a_{|D|,j'}^{(t)*}, a_{d,j}^{(t)*})$ .

### 2.3.3 Best Response Dynamics

Towards determining the PNE of the DDO game, the Best Response Dynamics (BRD) method [81] is adopted. The BRD is a natural method by which the IoT devices proceed to a PNE via a local search method. However, it is noted that the quality of the PNE depends on the order that the IoT devices update their data offloading strategies. In this research work, we consider an asynchronous BRD algorithm, where all the IoT devices update simultaneously their data offloading strategies.

The best response strategy of each IoT device to the other devices' data offloading strategies is defined as follows.

$$\begin{aligned}
 BR_d(\mathbf{a}_{-d,j}^{(t)*}) &= a_{d,j}^{(t)*} = \arg \max_{a_{d,j}^{(t)} \in A_d^{(t)}} U_d^{(t)}(a_{d,j}^{(t)}, \mathbf{a}_{-d,j}^{(t)}) = \\
 & \ln\left(\frac{c}{b} \cdot \left(\frac{I_d^{(t)}}{R_d^{(t)} \cdot T} + \frac{\phi_d^{(t)} \cdot I_d^{(t)}}{\sum_{a_{d',j'}^{(t)} \cdot I_{d'}^{(t)}}} + \frac{I_d^{(t)}}{R_d^{(t)} \cdot e_d^{(t)}} \cdot p_d^{(t)}\right) \cdot \sum_{\forall d' \neq d, d' \in D} a_{d',j'}^{(t)}\right) \\
 &= \frac{[1 - \frac{d' \neq d}{BU_{AV}}] \cdot F_{UAV} \cdot T}{\sum_{d' \neq d} \frac{1}{a_{d',j'}^{(t)} \cdot I_{d'}^{(t)}} - \left(\frac{I_d^{(t)}}{R_d^{(t)} \cdot T} + \frac{\phi_d^{(t)} \cdot I_d^{(t)}}{\sum_{a_{d',j'}^{(t)} \cdot I_{d'}^{(t)}}} + \frac{I_d^{(t)}}{R_d^{(t)} \cdot e_d^{(t)}} \cdot p_d^{(t)}\right) [1 - \frac{d' \neq d}{BU_{AV}}] \cdot F_{UAV} \cdot T}
 \end{aligned} \tag{2.7}$$

In a nutshell, the asynchronous BRD algorithm that determines a PNE of the DDO game is described in Algorithm 1. The complexity of the asynchronous BRD algorithm is  $\mathcal{O}(|D| \cdot Ite)$ ,  $|D| \gg Ite$ , where  $Ite$  is the total number of iterations in order the algorithm to converge to the PNE. In Section 3 indicative numerical results in terms of the required number of iterations (and actual time) required for convergence are presented.

## 2.4 Reinforcement Learning-Enabled Distributed Edge Computing

In this section, an artificial intelligence approach is introduced based on reinforcement learning algorithms to enable the IoT devices to determine their stable data offloading strategies, in order to process their computation tasks at the UAV-mounted MEC server, while mitigating their experienced overhead. The need for adopting these learning approaches versus the game-theoretic model (as expressed via the BRD framework), arises in several realistic cases including the ones where: a) the devices are not fully aware of the closed-form solution (Eq. 2.7), and/or b) the devices' data offloading strategy space  $A_d^{(t)}$  is discrete (rather than being continuous

---

**Algorithm 1** Asynchronous BRD Algorithm

---

1: **Input:**  $D, C_d^{(t)}, p_d^{(t)}, e_d^{(t)}, T, A_d^{(t)}, \forall d \in D$   
2: **Output:** Pure Nash Equilibrium:  $\mathbf{a}^{(t)*}$   
3: **Initialization:**  $ite = 0, Convergence = 0, \mathbf{a}^{(t)}|_{ite=0}$   
4: **while**  $Convergence == 0$  **do**  
5:    $ite = ite + 1;$   
6:   **for**  $d = 1$  to  $|D|$  **do**  
7:     Each IoT device  $d$  determines  $a_{d,j}^{(t)*}|_{ite}$  w.r.t.  $\mathbf{a}_{-d,j}^{(t)*}|_{ite}$  (Eq.2.7) and receives  
    $U_d^{(t,ite)}(a_{d,j}^{(t)*}|_{ite}, \mathbf{a}_{-d,j}^{(t)*}|_{ite})$   
8:   **end for**  
9:   **if**  $\mathbf{a}_{d,j}^{(t)*}|_{ite} = \mathbf{a}_{d,j}^{(t)*}|_{ite-1}$  **then**  
10:      $Convergence = 1$   
11:   **end if**  
12: **end while**

---

as assumed in the game-theoretic model). In particular, three different sets of reinforcement learning algorithms are examined, namely the gradient ascent, log-linear, and Q-learning, and their inherent properties are exploited. More importantly, their convergence to a data offloading strategy set for all the IoT devices, is critically compared against the corresponding ones at the PNE point, obtained through the BRD algorithm under the game-theoretic framework introduced in Section 2.3.

### 2.4.1 Gradient Ascent Learning

In the gradient ascent reinforcement learning approach, the IoT devices act as Learning Automata (LA) and they learn their environment by performing gradient updates of their perceived utility. Specifically, Learning Automata are policy iterators, that keep a vector action probabilities over the available action set and, as is common in Reinforcement Learning, these probabilities are updated based on feedback sig-

nals that are received from the environment. These learning schemes perform very well in game theoretic environments, even though they do not require any information exchange (actions, rewards, strategies) on the other players in the game. Each device's data offloading decisions are characterized by an action probability vector  $\mathbf{P}_d^{(ite)} = [P_{a_{d,min}^{(t)}}^{(ite)}, \dots, P_{a_{d,j}^{(t)}}^{(ite)}, \dots, P_{a_{d,max}^{(t)}}^{(ite)}]$ . At each iteration of the gradient ascent algorithm, each device probabilistically chooses its potential data offloading strategy. The IoT devices make their stable data offloading decision, if  $P_{a_{d,j}^{(t)}}^{(ite)} \geq P_{thres}, \forall d \in D$ , where  $P_{thres}$  is a threshold value of the action probability. The most commonly applied gradient ascent learning algorithm is called Linear Reward-Inaction (LRI) [82] and the corresponding action probability updating rule is given as follows [83].

$$P_{a_{d,j}^{(t)}}^{(ite+1)} = P_{a_{d,j}^{(t)}}^{(ite)} + \eta [\hat{U}_d^{(t)}]^{(ite)} (1 - P_{a_{d,j}^{(t)}}^{(ite)}), \text{ if } a_{d,j}^{(t)}|_{ite} = a_{d,j}^{(t)}|_{ite+1} \quad (2.8a)$$

$$P_{a_{d,j}^{(t)}}^{(ite+1)} = P_{a_{d,j}^{(t)}}^{(ite)} - \eta [\hat{U}_d^{(t)}]^{(ite)} P_{a_{d,j}^{(t)}}^{(ite)}, \text{ if } a_{d,j}^{(t)}|_{ite} \neq a_{d,j}^{(t)}|_{ite+1} \quad (2.8b)$$

where  $\eta \in (0, 1]$  is the learning rate of the IoT devices. For large values of the learning rate  $\eta$ , the IoT devices explore less thoroughly their available data offloading strategies, thus they converge fast to their stable decisions, however, they achieve lower utility. The exact opposite holds true for small values of the learning rate. The reward that each device receives by its data offloading decision at each iteration  $ite$  of the LRI algorithm is the normalized utility  $[U_d^{(t)}]^{(ite)} = \frac{[U_d^{(t)}]^{(ite)}}{\sum_{d \in D} [U_d^{(t)}]^{(ite)}}$ .

The proposed gradient ascent reinforcement learning algorithm which converges to a PNE of the DDO game is described in Algorithm 2. As far as the Linear Reward-Inaction (LRI) algorithm's complexity is concerned, we indicate as  $Ite$  the number of iterations that the reinforcement learning algorithm requires to converge at a specific timeslot  $t$ . The respective total complexity is  $\mathcal{O}(Ite \cdot (|D| + |D| \cdot |A_d^{(t)}|))$ , because for the IoT devices' action selections and the corresponding action probabilities' updates components at a single iteration, the complexities are  $\mathcal{O}(|D|)$  and  $\mathcal{O}(|D| \cdot |A_d^{(t)}|)$  respectively. Since, the rest of the LRI algorithm includes only algebraic calculations

of constant time complexity, i.e.,  $\mathcal{O}(1)$ , the aforementioned overall complexity holds true.

## 2.4.2 Log-Linear Learning

An alternative set of reinforcement learning algorithms, named log-linear learning algorithms, is examined in this subsection. The log-linear learning algorithms enable the IoT devices to converge to the best PNE with high probability compared to gradient ascent learning algorithms that simply allow the devices to explore their distributed edge computing environment. Furthermore, the log-linear learning algorithms allow the IoT devices to deviate from their probabilistically optimal decisions and make some suboptimal decisions in order to thoroughly explore their available data offloading action space. An indicative log-linear learning algorithm is the Binary Log-Linear Learning (BLLL) algorithm. In BLLL algorithm, each IoT device initially selects a data offloading strategy among the available ones, with equal probability for each one, i.e.,  $P_{a_{d,j}}^{(ite=0)} = \frac{1}{|A_d^{(t)}|}$ . Then, at each iteration  $ite$  of the BLLL algorithm, one IoT device is randomly selected to perform exploration and learning. At the exploration phase, the device selects an alternative data offloading strategy  $a_{d,j'}^{(t)}|_{ite}$  and receives the corresponding utility  $[U_d^{(t)}]^{(ite)}$ . At the learning phase, the IoT device updates its data offloading strategy based on the following probabilistic rule.

$$P_{a_{d,j}}^{(ite+1)} = \frac{e^{[U_d^{(t)}]^{(ite)} \cdot \beta}}{e^{[U_d^{(t)}]^{(ite)} \cdot \beta} + e^{[U_d^{(t)}]^{(ite)} \cdot \beta}}, \text{ if } a_{d,j}|_{ite+1} = a_{d,j'}|_{ite} \quad (2.9a)$$

$$P_{a_{d,j}}^{(ite+1)} = \frac{e^{[U_d^{(t)}]^{(ite)} \cdot \beta}}{e^{[U_d^{(t)}]^{(ite)} \cdot \beta} + e^{[U_d^{(t)}]^{(ite)} \cdot \beta}}, \text{ if } a_{d,j}|_{ite+1} \neq a_{d,j'}|_{ite} \quad (2.9b)$$

where  $\beta \in \mathbb{R}^+$  is the learning parameter and for large values of  $\beta$  the IoT devices explore more thoroughly their available data offloading strategies. The BLLL algorithm converges when the summation of the devices' perceived utilities remain

approximately the same for a very small number of  $K$  consecutive iterations (convergence criterion).

The proposed Binary Log-Linear Learning (BLLL) that converges to a PNE of the DDO game is described in Algorithm 3. Regarding its corresponding complexity analysis, if we denote  $Ite$  the total number of iterations in order the reinforcement learning algorithm to converge to the PNE, the overall BLLL algorithm's complexity is  $\mathcal{O}(Ite)$ . This holds true, since as we have already mentioned, at each iteration  $ite$  of the algorithm only one IoT device performs exploration and learning, which results to algebraic calculations of constant time  $\mathcal{O}(1)$ . As a consequence, the convergence of the proposed algorithm is affected only by the total number of iterations.

### 2.4.3 Q-Learning

An alternative reinforcement learning approach, known as stateless Q-Learning, is studied in this subsection. The stateless Q-Learning utilizes the stochastic approximation methods in order to allow the IoT devices to explore and learn their environment by following a Markov Decision Process (MDP) policy, thus converging eventually to their stable data offloading decisions. Specifically, each IoT device  $d$  preserves an action values vector  $Q_d^{(ite)}(\mathbf{a}) = [Q_{a_{d,min}^{(t)}}^{(ite)}, \dots, Q_{a_{d,j}^{(t)}}^{(ite)}, \dots, Q_{a_{d,max}^{(t)}}^{(ite)}]$ , where  $Q_{a_{d,j}^{(t)}}^{(ite)}$  denotes the estimated value of that action  $a_{d,j}^{(t)}$  up to the iteration  $ite$ , i.e., it depicts the expected utility  $U_d^{(t,ite)}$  given that  $a_{d,j}^{(t)}$  is selected:

$$Q_{a_{d,j}^{(t)}}^{(ite)} \cong \mathbb{E}[U_d^{(t,ite)} | a_{d,j}^{(t)} | ite] \quad (2.10)$$

An indicative way to estimate the aforementioned  $Q_{a_{d,j}^{(t)}}^{(ite)}$  value is based on the following standard Q-Learning update rule which is given as follows.

$$Q_{a_{d,j}^{(t)}}^{(ite)} = Q_{a_{d,j}^{(t)}}^{(ite)} + \theta \cdot (U_d^{(t,ite)} - Q_{a_{d,j}^{(t)}}^{(ite)}) \quad (2.11)$$

where  $\theta \in (0, 1]$  is the learning parameter. Since each IoT device selects an offloading strategy at each iteration  $ite$ , we introduce the widely used action selection rule known as the greedy approach. According to the greedy rule, the IoT devices select the offloading strategies with the highest expected utility (Eq.2.12), thus they only exploit the knowledge that is acquired up to the iteration  $ite$ .

$$a_{d,j}^{(t)}|_{ite+1} = \arg \max_{a_{d,j}^{(t)} \in A_d^{(t)}} Q_d^{(ite)}(\mathbf{a}) \quad (2.12)$$

Additionally, we also examine an alternative action selection approach named  $\epsilon$ -greedy. Under the  $\epsilon$ -greedy approach, the IoT devices perform exploration with probability  $\epsilon$  by selecting another data offloading strategy with equal probability  $\frac{1}{|A_d^{(t)}|-1}$  other than the one that maximizes their expected utility. For  $\epsilon = 0$ , the  $\epsilon$ -greedy approach is equivalent to the greedy approach.

The proposed Q-Learning algorithm that converges to a PNE of the DDO game is described extensively in Algorithm 4. We indicate as  $Ite$  the number of epochs that the reinforcement learning algorithm will execute in order to approach a potential Pure Nash Equilibrium at a specific timeslot  $t$ . The respective total complexity is  $\mathcal{O}(Ite \cdot |D|)$ , because all the IoT devices select actions with respect to the corresponding action values based on the  $\epsilon$ -greedy approach, experience a reward and afterwards they update this Q-value. All of these steps are performed in a sequential way and since the rest of the stateless Q-Learning algorithm contains only algebraic calculations of constant time complexity, i.e.,  $\mathcal{O}(1)$ , the aforementioned overall complexity holds true.

Detailed numerical evaluation of the examined reinforcement learning algorithms is provided in Section 3.

---

**Algorithm 2** LRI Algorithm

---

1: **Input:**  $D, C_d^{(t)}, p_d^{(t)}, e_d^{(t)}, T, A_d^{(t)}, \forall d \in D$   
2: **Output:** Pure Nash Equilibrium:  $\mathbf{a}^{(t)*}$   
3: **Initialization:**  $ite = 0, Convergence = 0, P_{a_{d,j}^{(t)}}^{(ite)} = \frac{1}{|A_d^{(t)}|}, \forall d \in D, \forall a_{d,j}^{(t)} \in A_d^{(t)}, \eta$   
4: **while**  $Convergence == 0$  **do**  
5:   **for**  $d = 1$  to  $|D|$  **do**  
6:     Choose an offloading strategy  $a_{d,j}^{(t)}$  to offload the respective amount of data to the UAV-mounted MEC server, with respect to  $\mathbf{P}_d^{(ite)}$   
7:   **end for**  
8:   **for**  $d = 1$  to  $|D|$  **do**  
9:     Determines the respective reward  $[U_d^{(t)}]^{(ite)}$   
10:   **end for**  
11:   **for**  $d = 1$  to  $|D|$  **do**  
12:     **for all**  $a_{d,j}^{(t)} \in A_d^{(t)}$  **do**  
13:       Updates the action probabilities  $P_{a_{d,j}^{(t)}}^{(ite)}$  according to Eq.2.8a and Eq.2.8b  
14:     **end for**  
15:   **end for**  
16:   **if**  $\exists a_{d,j}^{(t)} \in A_d^{(t)} : P_{a_{d,j}^{(t)}}^{(ite)} \geq P_{thres}, \forall d \in D$  **then**  
17:      $Convergence = 1$   
18:   **else**  
19:      $ite = ite + 1$   
20:   **end if**  
21: **end while**

---



---

**Algorithm 3** BLLL Algorithm

---

- 1: **Input:**  $D, C_d^{(t)}, p_d^{(t)}, e_d^{(t)}, T, A_d^{(t)}, \forall d \in D$
  - 2: **Output:** Pure Nash Equilibrium:  $\mathbf{a}^{(t)*}$
  - 3: **Initialization:**  $ite = 0, Convergence = 0$ , Arbitrary Action Profile  $\mathbf{a}_{d,j}^{(t)}, \beta$
  - 4: **while**  $Convergence == 0$  **do**
  - 5:      $ite = ite + 1$
  - 6:     IoT device  $d \in D$  is randomly selected to perform exploration and learning
  - 7:     IoT device  $d$  selects an alternative data offloading strategy  $a_{d,j'}^{(t)}$  with equal probability  $P_{a_{d,j'}^{(t)}}^{(ite)} = \frac{1}{|A_d^{(t)}|}$
  - 8:     IoT device  $d$  receives the corresponding utility  $[U_d^{(t)}]^{(ite)}$
  - 9:     IoT device  $d$  Computes the probabilities based on the probabilistic rule presented in Eq.2.9a and Eq.2.9b
  - 10:     IoT device  $d$  updates the offloading strategy for the next iteration  $a_{d,j}^{(t,ite+1)}$  based on the above probabilities
  - 11:     **if** convergence criterion = True **then**
  - 12:          $Convergence = 1$
  - 13:     **else**
  - 14:          $ite = ite + 1$
  - 15:     **end if**
  - 16: **end while**
-

---

**Algorithm 4** Q-Learning Algorithm

---

- 1: **Input:**  $D, C_d^{(t)}, p_d^{(t)}, e_d^{(t)}, T, A_d^{(t)}, \forall d \in D$
  - 2: **Output:** Pure Nash Equilibrium:  $\mathbf{a}^{(t)*}$
  - 3: **Initialization:**  $ite = 0, ITERATIONS, Q_d^{(ite)}(\mathbf{a}), \forall d \in D, \theta$
  - 4: **while**  $ite \neq ITERATIONS$  **do**
  - 5:   **for**  $d = 1$  to  $|D|$  **do**
  - 6:     *Choose an offloading strategy  $a_{d,j}^{(t)}$  to offload the respective amount of data to the UAV-mounted MEC server, with respect to the action value  $Q_d^{(ite)}(\mathbf{a})$  based on the  $\epsilon$ -greedy approach*
  - 7:   **end for**
  - 8:   **for**  $d = 1$  to  $|D|$  **do**
  - 9:     *Determines the respective reward  $U_d^{(t,ite)}$*
  - 10:   **end for**
  - 11:   **for**  $d = 1$  to  $|D|$  **do**
  - 12:     *Updates the respective action value  $Q_{a_{d,j}^{(t)}}^{(ite)}$  which depicts the expected reward of the chosen action  $a_{d,j}^{(t)}$ , based on Eq.2.11*
  - 13:   **end for**
  - 14:    $ite = ite + 1$
  - 15: **end while**
-

# Chapter 3

## Experiments

### 3.1 Experiment Setup

In this section, indicative numerical results are presented to illustrate the performance of the proposed artificial intelligence-enabled distributed edge computing framework (Section 3.2). A detailed comparative analysis is performed to gain insight about the behavior of the different learning and exploitation approaches adopted in this paper, by highlighting the drawbacks and benefits of the BRD model versus the examined reinforcement learning approaches (Section 3.3). Additional discussions regarding the robustness and applicability of the proposed learning methods are provided in Section 3.4.

We consider an environment consisting of  $|D| = 250$  IoT devices, where each IoT device's distance from the UAV-mounted MEC server is randomly and uniformly distributed in the interval  $(10m, 400m)$ . The simulation parameters are as follows:  $I_d^{(t)} \in [20, 100]MBytes$ ,  $C_d^{(t)} \in [1, 5] \cdot 10^9 CPUcycles$ ,  $\phi_d^{(t)} = \frac{C_d^{(t)}}{I_d^{(t)}}$ ,  $p_d^{(t)} \in [1.2, 2]Watts$ ,  $W = 5MHz$ ,  $b = 0.74$ ,  $c = 0.0043$ ,  $B_{UAV} \geq \sum_{d \in D} I_d^{(t)}$  and  $F_{UAV} = 15 \cdot 10^9 \frac{CPUcycles}{sec}$ . Unless otherwise explicitly stated, we consider  $a_{d,min}^{(t)} \in (0, 0.2]$ ,  $a_{d,max}^{(t)} \in [0.8, 1.0]$

with an intermediate step of 0.05,  $\eta = 0.3$ ,  $\beta = 1000$  and  $\theta = 0.6$ . The proposed framework’s evaluation was conducted via modeling and simulation and was executed in a MacBook Pro Laptop, 2.5GHz Intel Core i7, with 16GB LPDDR3 available RAM.

### 3.2 Pure Operation Performance

In this subsection, we examine the operation performance of the proposed artificial intelligence-enabled distributed edge computing framework under the game-theoretic and the reinforcement learning models, in terms of: the IoT devices’ data offloading strategies to the UAV-mounted MEC server, the corresponding experienced overhead and utility, the overall system’s achieved social welfare, as well as the required iterations and time (execution time) for convergence of the different examined approaches.

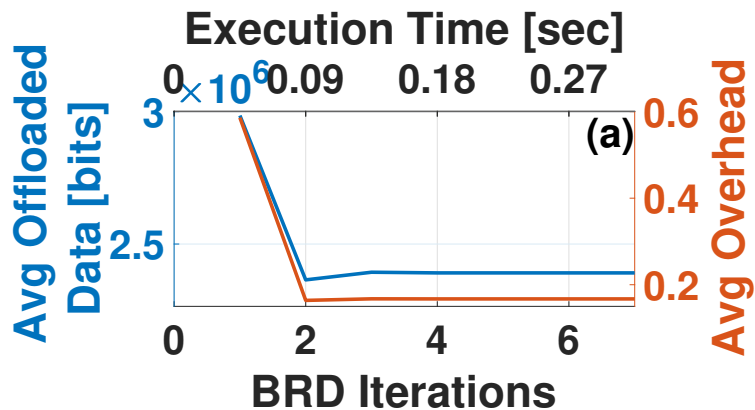


Figure 3.1: BRD Average Offloaded Data & Overhead

In particular, Fig.3.1 presents the IoT devices’ average offloaded data to the UAV and the corresponding experienced overhead as a function of the BRD algorithm’s iterations and real execution time (lower and upper horizontal axis respectively).

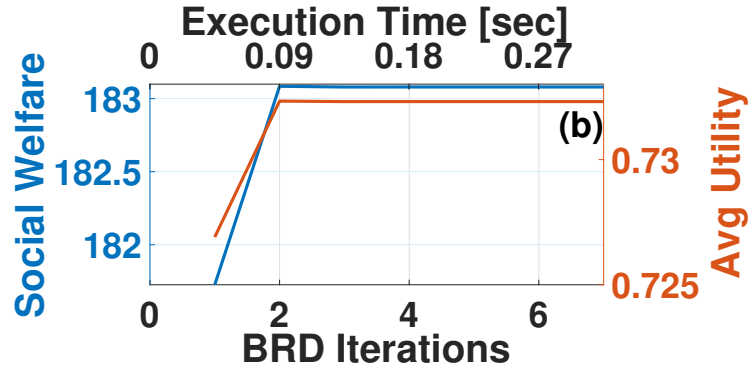


Figure 3.2: BRD Social Welfare & Utility

The results reveal that the BRD algorithm converges fast to a PNE (i.e., practically in less than 4 iterations, equivalent to 0.18 sec). Also, the IoT devices converge to a PNE, where they experience low average overhead (Fig.3.1) and high levels of utility (Fig. 3.2). Moreover, by studying the BRD framework from the system’s perspective, we observe that at the PNE high levels of social welfare are obtained (Fig.3.2).



Figure 3.3: LRI Action Probabilities

Fig.3.3 presents the convergence of the data offloading strategies of one indicative IoT device to a stable data offloading decision following the LRI algorithm. It is observed that the devices’ data offloading converge to a stable decision in less than 100 iterations i.e., 0.32 sec, following the learning procedure of the gradient ascent

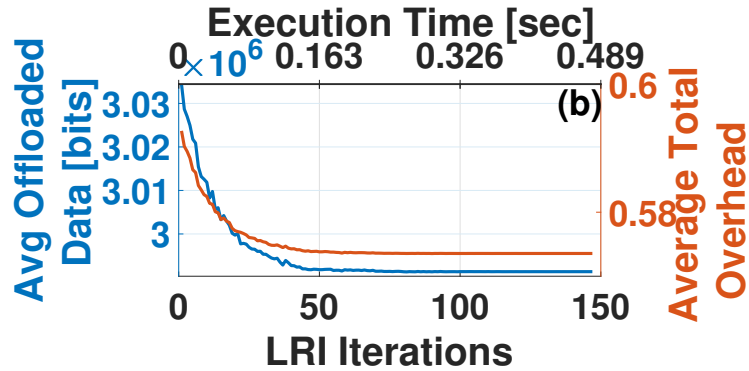


Figure 3.4: LRI Average Offloaded Data & Overhead

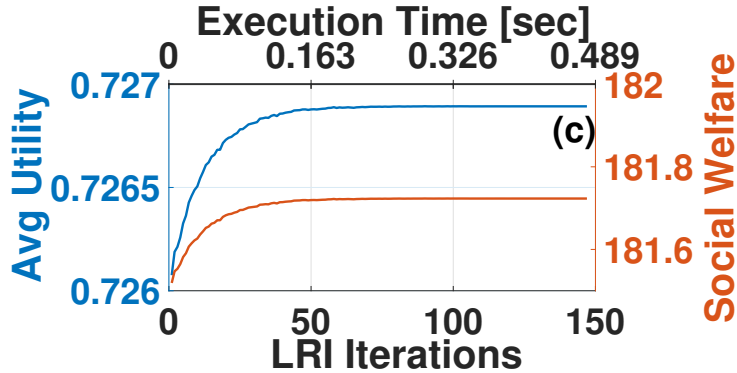


Figure 3.5: LRI Social Welfare & Utility

learning algorithm. Also, Fig. 3.4, 3.5 present the convergence of the IoT devices' average offloaded data, overhead, and utility, as well as the system's social welfare. The results show that the IoT devices learn in a distributed manner their surrounding environment and they strategically decide their data offloading strategies in order to achieve low overhead and high utility, while collectively enjoy high levels of social welfare. Furthermore, Fig.3.6 presents the trade-off among the achieved average utility of the IoT devices with the corresponding execution time of the LRI algorithm in order to converge to a stable data offloading decision as a function of the learning parameter  $\eta$ . The results reveal that for increasing values of the learning parameter  $\eta$ , the devices learn faster their environment and make a data offloading decision.

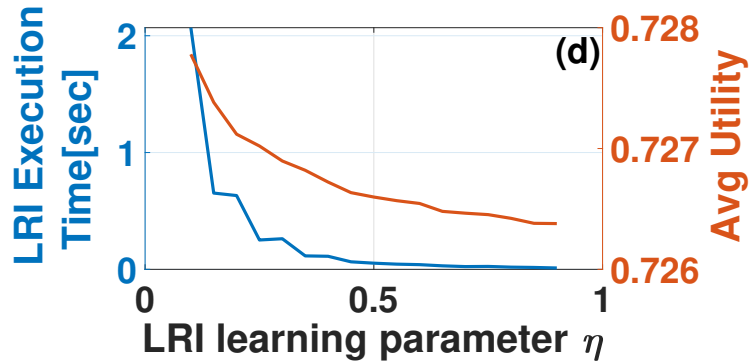


Figure 3.6: LRI Learning Parameter

However, this comes at the cost of lower achieved utility, as they under-explore their available data offloading decisions.

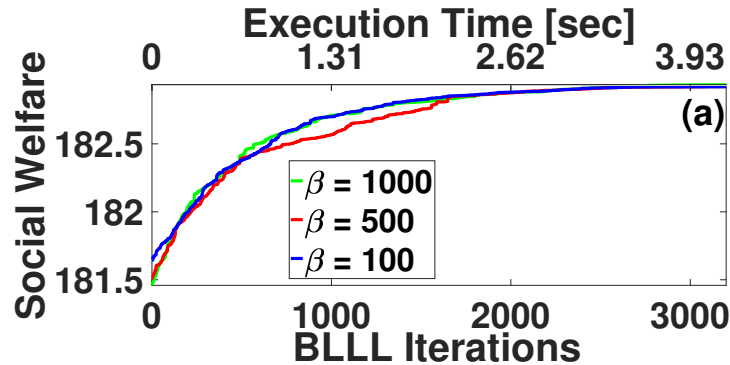


Figure 3.7: BLLL Social Welfare

Fig.3.7-3.10 examine the behavior of the BLLL algorithm, for different values of the learning parameter  $\beta$ , as a function of the iterations and the real execution time. The results show that the BLLL algorithm converges to the PNE with high probability, while the IoT devices follow a learning approach, bearing however the cost of longer convergence time. Thus, the IoT devices converge close to the PNE and they achieve high utility levels (Fig.3.8), and low overhead (Fig.3.10), while intelligently deciding their data offloading strategies (Fig.3.9). Furthermore, the system converges to high levels of social welfare (Fig.3.7). Moreover, it is observed

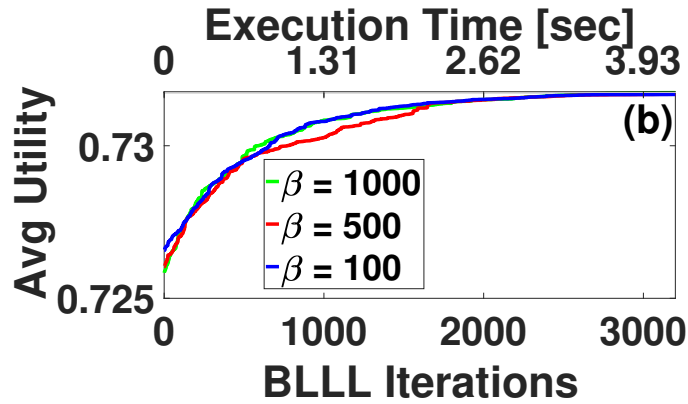


Figure 3.8: BLLL Average Utility

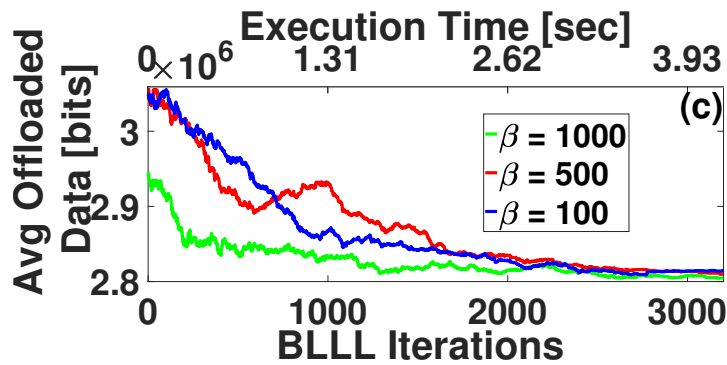


Figure 3.9: BLLL Average Offloaded Data

that better results are achieved for higher values of the learning parameter  $\beta$ .

Similarly, Fig.3.11-3.14 present the corresponding operation performance of the Q-learning approach, i.e., both the greedy and the  $\epsilon$ -greedy, in terms of the system's social welfare, the IoT devices' average utility, offloaded data, and overhead, respectively, as a function of the Q-learning algorithm's iterations and real execution time. The results reveal that the Q-learning algorithms converge to stable data offloading decisions for all the IoT devices (Fig.3.13) achieving high utilities (Fig.3.12), low overhead (Fig.3.14), and high social welfare values (Fig.3.11). It is also observed that the  $\epsilon$ -greedy algorithm by allowing with small probability ( $\epsilon = 0.01$ ) the IoT



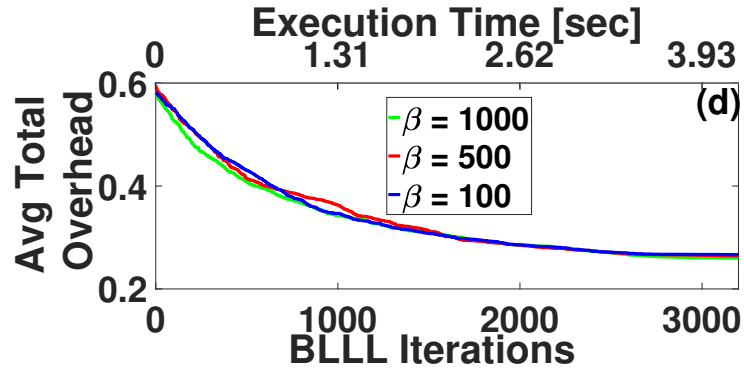


Figure 3.10: BLLL Average Total Overhead

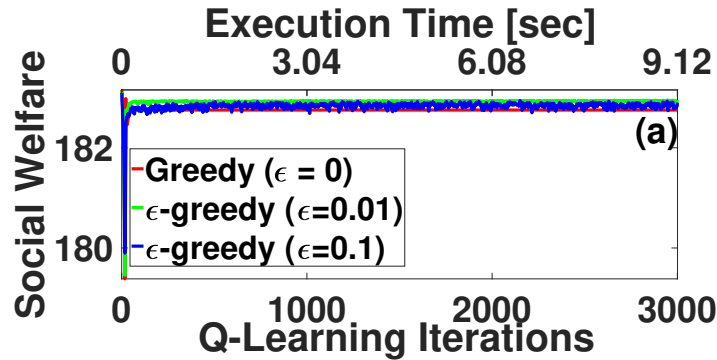


Figure 3.11: Q-Learning Social Welfare

devices to explore other data offloading strategies than the ones that maximize the expected utilities, achieve the best results among the different Q-learning implementations. This is due to the fact that the IoT devices can explore alternative actions compared to the greedy Q-learning algorithm ( $\epsilon = 0$ ) where they myopically choose the strategies that offer them the maximum expected utility. On the other hand, if the devices overexplore alternative strategies, i.e.,  $\epsilon = 0.1$ , they deviate from good outcomes, being "lost" in the exploration phase.

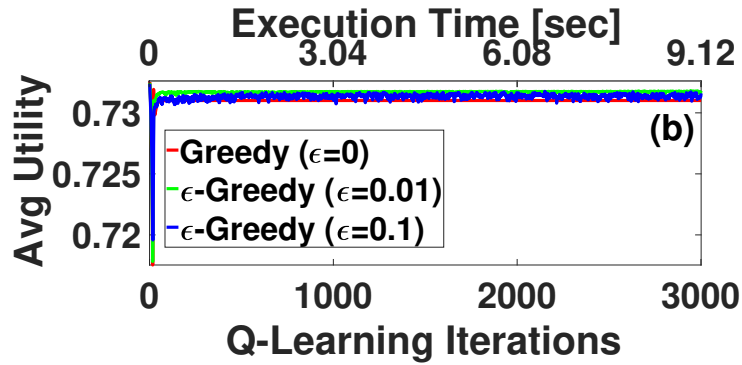


Figure 3.12: Q-Learning Average Utility

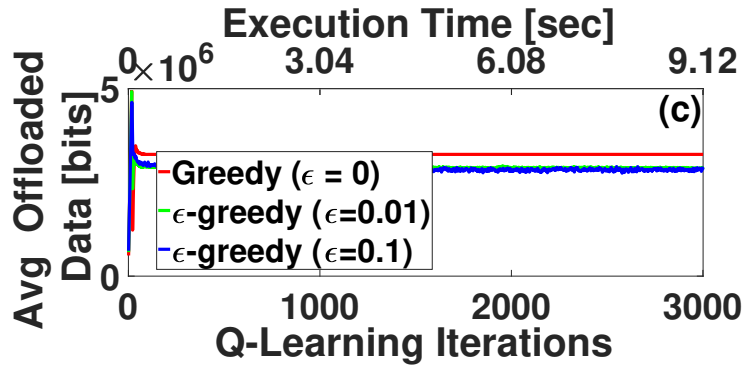


Figure 3.13: Q-Learning Average Offloaded Data

### 3.3 Comparative Evaluation

In this subsection, a comparative evaluation among the examined learning models (i.e. game theoretic model and reinforcement learning ones) utilized to determine the IoT devices' data offloading strategies in the distributed edge computing environment is performed.

Fig.3.15-3.17 present the system's social welfare, the social welfare's mean square error with respect to the BRD model, and the execution time of all the examined algorithms, respectively. The results reveal that the game-theoretic model - as reflected by the BRD algorithm - illustrates the best results, both in terms of achieved

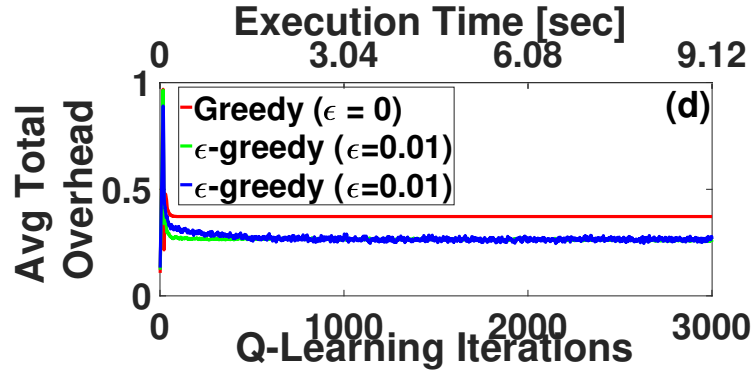


Figure 3.14: Q-Learning Average Total Overhead

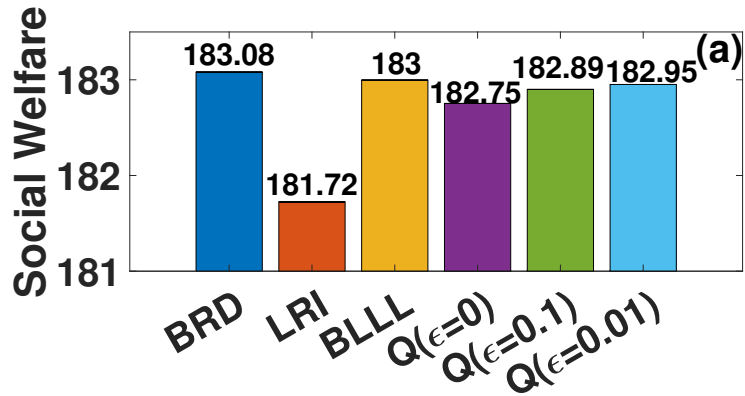


Figure 3.15: RL Social Welfare Comparison

social welfare and execution time. Then, the BLLL algorithm achieves the highest social welfare among all the reinforcement learning algorithms, given its inherent attribute to converge to a PNE with high probability as demonstrated in previous subsection. On the other hand, the LRI approach, given its simplistic action update rule (Eq.2.8a,2.8b) converges fast (Fig.3.17) to a stable data offloading vector for all the IoT devices, while sacrificing the achieved welfare (Fig.3.15). The Q-Learning approaches, i.e.,  $\epsilon = 0, 0.01, 0.1$  illustrate similar execution time (Fig.3.17) and high levels of social welfare (Fig.3.15) with the BRD algorithm’s PNE outcome. In a nutshell, based on the results in Fig.3.16, we observe that the smallest mean square error of the social welfare with respect to the BRD algorithm’s outcome is achieved by the

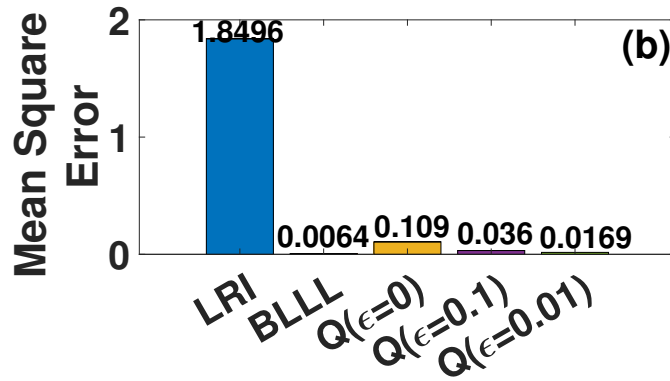


Figure 3.16: RL MSE Comparison

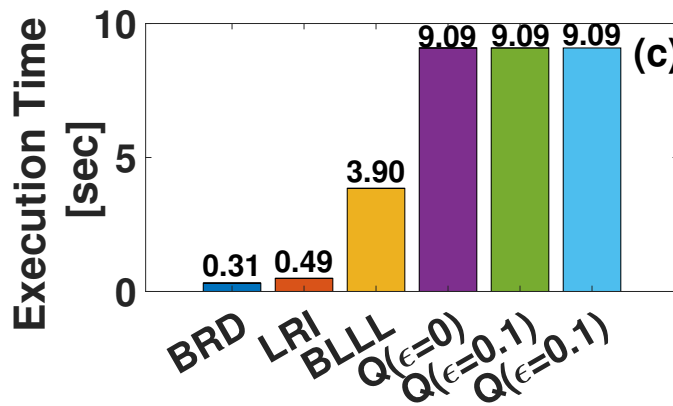


Figure 3.17: RL Execution Time

BLLL algorithm and then by the  $\epsilon$ -greedy Q-learning algorithms with  $\epsilon = 0.01$  and  $\epsilon = 0.1$ , respectively. Also, by allowing the IoT devices to slightly deviate from the strategies that maximize their expected utilities, they achieve better results than the other reinforcement learning approaches, as they thoroughly explore their alternative strategies.

The above discussion is summarized in Fig.3.16. We observe that the smallest mean square error of the social welfare with respect to the BRD algorithm's outcome is achieved by the BLLL algorithm and then by the  $\epsilon$ -greedy Q-learning algorithms with  $\epsilon = 0.01$  and  $\epsilon = 0.1$ , respectively. Following the previous algorithms, the

	<b>BRD</b>	<b>LRI</b>	<b>BLLL</b>	<b>Q(<math>\epsilon = 0</math>)</b>	<b>Q(<math>\epsilon = 0.1</math>)</b>	<b>Q(<math>\epsilon = 0.01</math>)</b>
<b>Social Welfare</b>	183.08	181.72	183	182.75	182.89	182.95
<b>MSE</b>	-	1.8496	0.0064	0.109	0.036	0.0169
<b>Execution Time</b>	0.31	0.49	3.90	9.09	9.09	9.09

Table 3.1: RL Comparison Table

greedy Q-learning algorithm still illustrates results close to the BRD algorithm’s ones, while the LRI algorithm achieves the worst outcome in terms of the system’s social welfare. Finally, the comparative results between the different reinforcement learning algorithms that were discussed above and presented in the graphs, are all included in Table 3.1.

### 3.4 Discussion on Learning Methods Applicability

In the following a detailed analysis of the BLLL learning approach operation is performed, with respect to the strategy space size available to the IoT devices (i.e., available number of actions). The BLLL approach is selected as it demonstrated the best results among all the examined reinforcement learning frameworks.

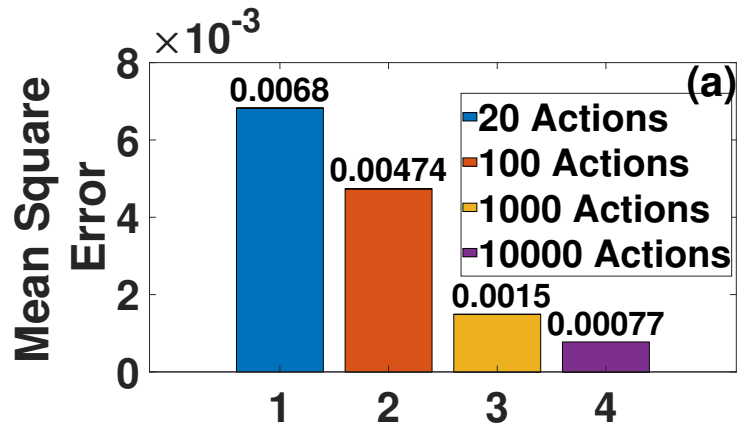


Figure 3.18: BLLL MSE for different number of actions

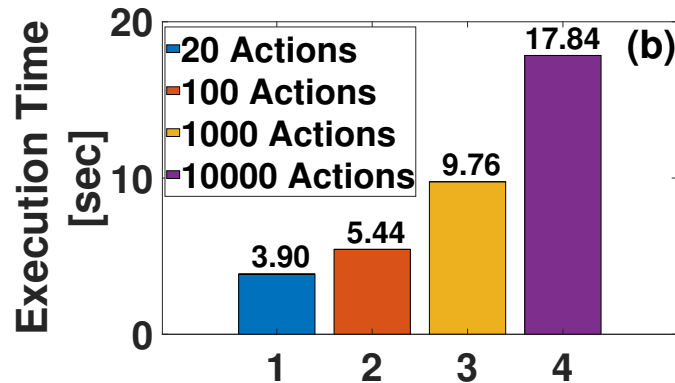


Figure 3.19: BLLL Execution Time for different number of actions

Fig.3.18 presents the mean square error of the BLLL algorithm's achieved social welfare compared to the outcome of the BRD algorithm for 20, 100, 1,000, 10,000 data offloading strategies, while Fig.3.19 shows the corresponding execution time of the BLLL algorithm. The results illustrate that as the devices' strategy space increases, the achieved social welfare by the BLLL algorithm approaches the corresponding one by the BRD algorithm, at the cost of increased execution time.

Based on the results provided in the latter two subsections, we observe that, the game-theoretic BRD algorithm converges to better results both from the devices' and the system's perspective, primarily due to the use of the closed-form used to determine the PNE (Eq. 2.7). Nevertheless, this requires that the devices are aware of the closed-form solution or can extrapolate it, which bears additional overhead. The reinforcement learning algorithms on the other hand, eliminate this assumption, by enabling the devices to learn their environment without having a priori knowledge of the optimal strategy rule. Last but not least, it should be noted that the reinforcement learning approaches can be better applied in realistic cases where the devices' strategy space is not continuous as considered in the game-theoretic model (i.e the devices may arbitrarily select any percentage of their data to offload), but instead the devices are allowed to select their data offloading strategies from a discrete

### *Chapter 3. Experiments*

predefined strategy space.

## Chapter 4

# Conclusion and Future Works

In this paper, an artificial intelligence-enabled distributed edge computing framework is proposed, to support IoT applications by exploiting the computing capabilities of a UAV-mounted MEC server. The communication and computation overhead experienced by the IoT devices is modeled, and appropriate utility functions are designed for the IoT devices to measure their satisfaction from offloading their computation tasks in the distributed edge computing environment. A non-cooperative game is formulated among the IoT devices and its PNE, i.e., devices' optimal data offloading strategies, is determined following the theory of submodular games. This game theoretic-model facilitates a process that enables the devices to learn, scrutinize the performance of other devices, and adjust their own behavior accordingly. Alternative reinforcement learning algorithms are adopted, i.e., gradient ascent, log-linear, and Q-learning, to determine the devices' stable data offloading strategies. Detailed numerical results are presented that demonstrate the operational characteristics and performance of the different models and algorithms, while they are compared against each other.

Part of our future work is to extend and evaluate the presented framework, while



## Chapter 4. Conclusion and Future Works

considering a multi-UAV-mounted servers setup, where the IoT devices can exploit the different computation choices of the environment. Moreover, another aspect of our future work is to examine the case where the actions of the IoT devices with respect to the UAV-mounted MEC server, i.e., offloaded bits, reside in a continuous space and also design a satisfactory UAV trajectory in the continuous two-dimensional area. In this case, it becomes impossible to represent the action values in a finite data structure such as a  $1D$  matrix and thus we will have to construct a non-linear function approximator via deep neural networks. As a consequence, we will utilize Deep Reinforcement Learning (DRL) where we will deploy several Temporal Difference (TD)-based Value-based methods such as Deep Q-Networks (DQN), Double Deep Q-Networks (DDQN), Dueling Networks, as well as Policy-based methods such as Advantage Actor Critic (A2C) and Deep Deterministic Policy Gradients (DDPG).

Additionally, we envision the integration of the blockchain data structure [84] and of a truth-inducing sybil resistant decentralized blockchain oracle [85] in order the IoT devices to vote regarding their satisfaction from the perceived Quality of Service (QoS) and Quality of Experience (QoE) from the UAV. Moreover, another important aspect which is interesting to examine in the future is the incentivization of the IoT devices to offer their data to the UAV following a labor economic approach [86] as well as the importance of the information that each IoT device wants to offload in a public safety scenario [72]. The security aspect regarding these use cases is also essential, since in a public safety scenario, i.e., terrorist attack, the IoT devices may have to mask their communication's information in a way that are not traceable by malicious users [87, 88].

We are also inclined to examine the case where there are multiple UAVs serving the IoT devices and the latter ones have to perform autonomous decision-making regarding to which UAV they will partially offload their data [46, 89]. At this case,

#### *Chapter 4. Conclusion and Future Works*

we should also consider the case of the incentivization and management of the UAVs in order to process the IoT devices' data [90,91], and also the resource orchestration in such a heterogeneous communication environment [92–94], where the UAVs may have different characteristics.

# References

- [1] L. Atzori, A. Iera, and G. Morabito, “The internet of things: A survey,” *Computer networks*, vol. 54, no. 15, pp. 2787–2805, 2010.
- [2] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, “Internet of things for smart cities,” *IEEE Internet of Things journal*, vol. 1, no. 1, pp. 22–32, 2014.
- [3] Y. Qian, J. Wu, R. Wang, F. Zhu, and W. Zhang, “Survey on reinforcement learning applications in communication networks,” 2019.
- [4] N. Hassan, S. Gillani, E. Ahmed, I. Yaqoob, and M. Imran, “The role of edge computing in internet of things,” *IEEE Communications Magazine*, vol. 56, no. 11, pp. 110–115, 2018.
- [5] P. J. Werbos, “The new ai: Basic concepts, and urgent risks and opportunities in the internet of things,” in *Artificial Intelligence in the Age of Neural Networks and Brain Computing*, pp. 161–190, Elsevier, 2019.
- [6] L. Lei, Y. Tan, K. Zheng, S. Liu, K. Zhang, and X. Shen, “Deep reinforcement learning for autonomous internet of things: Model, applications and challenges,” *IEEE Communications Surveys & Tutorials*, 2020.
- [7] P. A. Apostolopoulos, E. E. Tsiropoulou, and S. Papavassiliou, “Risk-aware social cloud computing based on serverless computing model,” in *2019 IEEE Global Communications Conference (GLOBECOM)*, pp. 1–6, IEEE, 2019.
- [8] A. Pérez, G. Moltó, M. Caballer, and A. Calatrava, “Serverless computing for container-based architectures,” *Future Generation Computer Systems*, vol. 83, pp. 50–59, 2018.
- [9] W. Z. Khan, E. Ahmed, S. Hakak, I. Yaqoob, and A. Ahmed, “Edge computing: A survey,” *Future Generation Computer Systems*, vol. 97, pp. 219–235, 2019.

## References

- [10] Y. C. Hu, M. Patel, D. Sabella, N. Sprecher, and V. Young, “Mobile edge computing—a key technology towards 5g,” *ETSI white paper*, vol. 11, no. 11, pp. 1–16, 2015.
- [11] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, “A survey on mobile edge computing: The communication perspective,” *IEEE Communications Surveys & Tutorials*, vol. 19, no. 4, pp. 2322–2358, 2017.
- [12] Y. Dai, D. Xu, S. Maharjan, G. Qiao, and Y. Zhang, “Artificial intelligence empowered edge computing and caching for internet of vehicles,” *IEEE Wireless Communications*, vol. 26, no. 3, pp. 12–18, 2019.
- [13] Q. Han, S. Liang, and H. Zhang, “Mobile cloud sensing, big data, and 5g networks make an intelligent and smart world,” *IEEE Network*, vol. 29, no. 2, pp. 40–45, 2015.
- [14] R. Li, Z. Zhao, X. Zhou, G. Ding, Y. Chen, Z. Wang, and H. Zhang, “Intelligent 5g: When cellular networks meet artificial intelligence,” *IEEE Wireless Communications*, vol. 24, no. 5, pp. 175–183, 2017.
- [15] M. J. Osborne and A. Rubinstein, *A course in game theory*. MIT press, 1994.
- [16] M. Rabin, “Incorporating fairness into game theory and economics,” *The American economic review*, pp. 1281–1302, 1993.
- [17] D. Ross, “Game theory in the stanford encyclopedia of philosophy,” *A cura di Zalta EN*, 2006.
- [18] P. Battigalli, M. Dufwenberg, *et al.*, “Psychological game theory,” tech. rep., 2019.
- [19] M. Le Breton and K. Van der Straeten, “Government formation and electoral alliances: The contribution of cooperative game theory to political science,” *Revue d’économie politique*, vol. 127, pp. 637–736, 2017.
- [20] T. R. Burns, E. Roszkowska, N. Machado Des Johansson, and U. Corte, “Paradigm shift in game theory: Sociological re-conceptualization of human agency, social structure, and agents’ cognitive-normative frameworks and action determination modalities,” *Social sciences*, vol. 7, no. 3, p. 40, 2018.
- [21] S. Ranadheera, S. Maghsudi, and E. Hossain, “Minority games with applications to distributed decision making and control in wireless networks,” *IEEE Wireless Communications*, vol. 24, no. 5, pp. 184–192, 2017.

## References

- [22] J. Chen, C. Hua, and C. Liu, “Considerations for better construction and demolition waste management: Identifying the decision behaviors of contractors and government departments through a game theory decision-making model,” *Journal of cleaner production*, vol. 212, pp. 190–199, 2019.
- [23] Z. Han, D. Niyato, W. Saad, and T. Başar, *Game Theory for Next Generation Wireless and Communication Networks: Modeling, Analysis, and Design*. Cambridge University Press, 2019.
- [24] R. Azad Gholami, L. K. Sandal, and J. Uboe, “Solution algorithms for optimal buy-back contracts in multi-period channel equilibria with stochastic demand and delayed information,” *NHH Dept. of Business and Management Science Discussion Paper*, no. 2019/10, 2019.
- [25] Z. Zheng, L. Song, Z. Han, G. Y. Li, and H. V. Poor, “Game theory for big data processing: multileader multifollower game-based admn,” *IEEE Transactions on Signal Processing*, vol. 66, no. 15, pp. 3933–3945, 2018.
- [26] Y. Zhang and M. Guizani, *Game theory for wireless communications and networking*. CRC press, 2011.
- [27] P. Vamvakas, E. E. Tsiropoulou, and S. Papavassiliou, “Dynamic provider selection & power resource management in competitive wireless communication markets,” *Mobile Networks and Applications*, vol. 23, no. 1, pp. 86–99, 2018.
- [28] A. Agah and S. K. Das, “Preventing dos attacks in wireless sensor networks: A repeated game theory approach,” *IJ Network Security*, vol. 5, no. 2, pp. 145–153, 2007.
- [29] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.
- [30] L. P. Kaelbling, M. L. Littman, and A. W. Moore, “Reinforcement learning: A survey,” *Journal of artificial intelligence research*, vol. 4, pp. 237–285, 1996.
- [31] J. Kober, J. A. Bagnell, and J. Peters, “Reinforcement learning in robotics: A survey,” *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1238–1274, 2013.
- [32] L. Busoniu, R. Babuska, and B. De Schutter, “A comprehensive survey of multiagent reinforcement learning,” *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 38, no. 2, pp. 156–172, 2008.

## References

- [33] R. Caruana and A. Niculescu-Mizil, “An empirical comparison of supervised learning algorithms,” in *Proceedings of the 23rd international conference on Machine learning*, pp. 161–168, 2006.
- [34] C. Savaglio, P. Pace, G. Aloï, A. Liotta, and G. Fortino, “Lightweight reinforcement learning for energy efficient communications in wireless sensor networks,” *IEEE Access*, vol. 7, pp. 29355–29364, 2019.
- [35] H. Lu, C. Gu, F. Luo, W. Ding, and X. Liu, “Optimization of lightweight task offloading strategy for mobile edge computing based on deep reinforcement learning,” *Future Generation Computer Systems*, vol. 102, pp. 847–861, 2020.
- [36] T. Taleb, K. Samdanis, B. Mada, H. Flinck, S. Dutta, and D. Sabella, “On multi-access edge computing: A survey of the emerging 5g network edge cloud architecture and orchestration,” *IEEE Communications Surveys & Tutorials*, vol. 19, no. 3, pp. 1657–1681, 2017.
- [37] C. Wang, F. R. Yu, C. Liang, Q. Chen, and L. Tang, “Joint computation offloading and interference management in wireless cellular networks with mobile edge computing,” *IEEE Transactions on Vehicular Technology*, vol. 66, no. 8, pp. 7432–7445, 2017.
- [38] Y. Liu, S. Wang, J. Huang, and F. Yang, “A computation offloading algorithm based on game theory for vehicular edge networks,” in *2018 IEEE International Conference on Communications (ICC)*, pp. 1–6, IEEE, 2018.
- [39] G. Mitsis, P. A. Apostolopoulos, E. E. Tsiropoulou, and S. Papavassiliou, “Intelligent dynamic data offloading in a competitive mobile edge computing market,” *Future Internet*, vol. 11, no. 5, p. 118, 2019.
- [40] P. A. Apostolopoulos, E. E. Tsiropoulou, and S. Papavassiliou, “Cognitive data offloading in mobile edge computing for internet of things,” *IEEE Access*, vol. 8, pp. 55736–55749, 2020.
- [41] T. Zhang, Y. Xu, J. Loo, D. Yang, and L. Xiao, “Joint computation and communication design for uav-assisted mobile edge computing in iot,” *IEEE Trans. on Ind. Inform.*, pp. 1–1, 2019.
- [42] P. A. Apostolopoulos, M. Torres, and E. E. Tsiropoulou, “Satisfaction-aware data offloading in surveillance systems,” in *14th Workshop on Challenged Networks*, pp. 21–26, 2019.
- [43] Y. Du, K. Wang, K. Yang, and G. Zhang, “Energy-efficient resource allocation in uav based mec system for iot devices,” in *2018 IEEE GLOBECOM*, pp. 1–6, 2018.

## References

- [44] H. Guo and J. Liu, “Uav-enhanced intelligent offloading for internet of things at the edge,” *IEEE Transactions on Industrial Informatics*, vol. 16, no. 4, pp. 2737–2746, 2020.
- [45] Z. Yang, C. Pan, K. Wang, and M. Shikh-Bahaei, “Energy efficient resource allocation in uav-enabled mobile edge computing networks,” *IEEE Tran. on Wir. Com.*, vol. 18, no. 9, pp. 4576–4589, 2019.
- [46] K. Rael, G. Fragkos, J. Plusquellic, and E. E. Tsiropoulou, “Uav-enabled human internet of things,” in *2020 16th International Conference on Distributed Computing in Sensor Systems (DCOSS)*, pp. 312–319, 2020.
- [47] Y. Liu, M. Qiu, J. Hu, and H. Yu, “Incentive uav-enabled mobile edge computing based on microwave power transmission,” *IEEE Access*, vol. 8, pp. 28584–28593, 2020.
- [48] G. Mitsis, E. E. Tsiropoulou, and S. Papavassiliou, “Data offloading in uav-assisted multi-access edge computing systems: A resource-based pricing and user risk-awareness approach,” *Sensors*, vol. 20, no. 8, p. 2434, 2020.
- [49] Z. Tan, H. Qu, J. Zhao, S. Zhou, and W. Wang, “Uav-aided edge/fog computing in smart iot community for social augmented reality,” *IEEE Internet of Things Journal*, pp. 1–1, 2020.
- [50] F. Zhou, Y. Wu, R. Q. Hu, and Y. Qian, “Computation rate maximization in uav-enabled wireless-powered mobile-edge computing systems,” *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 9, pp. 1927–1941, 2018.
- [51] Y. Du, K. Yang, K. Wang, G. Zhang, Y. Zhao, and D. Chen, “Joint resources and workflow scheduling in uav-enabled wirelessly-powered mec for iot systems,” *IEEE Transactions on Vehicular Technology*, vol. 68, no. 10, pp. 10187–10200, 2019.
- [52] X. Cao, J. Xu, and R. Zhangt, “Mobile edge computing for cellular-connected uav: Computation offloading and trajectory optimization,” in *2018 IEEE 19th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, pp. 1–5, IEEE, 2018.
- [53] X. Hu, K.-K. Wong, K. Yang, and Z. Zheng, “Uav-assisted relaying and edge computing: Scheduling and trajectory optimization,” *IEEE Transactions on Wireless Communications*, vol. 18, no. 10, pp. 4738–4752, 2019.
- [54] T. Zhang, Y. Xu, J. Loo, D. Yang, and L. Xiao, “Joint computation and communication design for uav-assisted mobile edge computing in iot,” *IEEE Transactions on Industrial Informatics*, 2019.

## References

- [55] Z. Li, Y. Wang, M. Liu, R. Sun, Y. Chen, J. Yuan, and J. Li, “Energy efficient resource allocation for uav-assisted space-air-ground internet of remote things networks,” *IEEE Access*, vol. 7, pp. 145348–145362, 2019.
- [56] Z. Yu, Y. Gong, S. Gong, and Y. Guo, “Joint task offloading and resource allocation in uav-enabled mobile edge computing,” *IEEE Internet of Things Journal*, vol. 7, no. 4, pp. 3147–3159, 2020.
- [57] R. Wang, Y. Cao, A. Noor, T. A. Alamoudi, and R. Nour, “Agent-enabled task offloading in uav-aided mobile edge computing,” *Computer Communications*, vol. 149, pp. 324–331, 2020.
- [58] N. Cheng, W. Xu, W. Shi, Y. Zhou, N. Lu, H. Zhou, and X. Shen, “Air-ground integrated mobile edge networks: Architecture, challenges, and opportunities,” *IEEE Communications Magazine*, vol. 56, no. 8, pp. 26–32, 2018.
- [59] X. Hou, Z. Ren, J. Wang, S. Zheng, W. Cheng, and H. Zhang, “Distributed fog computing for latency and reliability guaranteed swarm of drones,” *IEEE Access*, vol. 8, pp. 7117–7130, 2020.
- [60] W. Ma, X. Liu, and L. Mashayekhy, “A strategic game for task offloading among capacitated uav-mounted cloudlets,” in *2019 IEEE International Congress on Internet of Things (ICIOT)*, pp. 61–68, IEEE, 2019.
- [61] J. Yang, X. You, G. Wu, M. M. Hassan, A. Almogren, and J. Guna, “Application of reinforcement learning in uav cluster task scheduling,” *Future Generation Computer Systems*, vol. 95, pp. 140–148, 2019.
- [62] Y. Du, K. Wang, K. Yang, and G. Zhang, “Trajectory design of laser-powered multi-drone enabled data collection system for smart cities,” in *2019 IEEE Global Communications Conference (GLOBECOM)*, pp. 1–6, IEEE, 2019.
- [63] F. Zhou, Y. Wu, H. Sun, and Z. Chu, “Uav-enabled mobile edge computing: Offloading optimization and trajectory design,” in *2018 IEEE International Conference on Communications (ICC)*, pp. 1–6, IEEE, 2018.
- [64] F. Zhou, Y. Wu, R. Q. Hu, and Y. Qian, “Computation rate maximization in uav-enabled wireless-powered mobile-edge computing systems,” *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 9, pp. 1927–1941, 2018.
- [65] Z. Na, M. Zhang, J. Wang, and Z. Gao, “Uav-assisted wireless powered internet of things: Joint trajectory optimization and resource allocation,” *Ad Hoc Networks*, vol. 98, p. 102052, 2020.



## References

- [66] J. Zhang, M. Lou, L. Xiang, and L. Hu, “Power cognition: Enabling intelligent energy harvesting and resource allocation for solar-powered uavs,” *Future Generation Computer Systems*, 2019.
- [67] M.-A. Messous, S.-M. Senouci, H. Sedjelmaci, and S. Cherkaoui, “A game theory based efficient computation offloading in an uav network,” *IEEE Transactions on Vehicular Technology*, vol. 68, no. 5, pp. 4964–4974, 2019.
- [68] M.-A. Messous, H. Sedjelmaci, N. Houari, and S.-M. Senouci, “Computation offloading game for an uav network in mobile edge computing,” in *2017 IEEE International Conference on Communications (ICC)*, pp. 1–6, IEEE, 2017.
- [69] N. H. Motlagh, M. Bagaa, and T. Taleb, “Uav-based iot platform: A crowd surveillance use case,” *IEEE Communications Magazine*, vol. 55, no. 2, pp. 128–134, 2017.
- [70] M. Samir, S. Sharafeddine, C. M. Assi, T. M. Nguyen, and A. Ghrayeb, “Uav trajectory planning for data collection from time-constrained iot devices,” *IEEE Transactions on Wireless Communications*, vol. 19, no. 1, pp. 34–46, 2019.
- [71] A. Farajzadeh, O. Ercetin, and H. Yanikomeroglu, “Uav data collection over noma backscatter networks: Uav altitude and trajectory optimization,” in *ICC 2019-2019 IEEE International Conference on Communications (ICC)*, pp. 1–7, IEEE, 2019.
- [72] G. Fragkos, E. E. Tsiropoulou, and S. Papavassiliou, “Disaster management and information transmission decision-making in public safety systems,” in *2019 IEEE Global Communications Conference (GLOBECOM)*, pp. 1–6, 2019.
- [73] M. Liu, T. Song, and G. Gui, “Deep cognitive perspective: Resource allocation for noma-based heterogeneous iot with imperfect sic,” *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 2885–2894, 2018.
- [74] C. Singhal and S. De, *Resource allocation in next-generation broadband wireless access networks*. IGI Global, 2017.
- [75] P. A. Apostolopoulos, E. E. Tsiropoulou, and S. Papavassiliou, “Risk-aware data offloading in multi-server multi-access edge computing environment,” *IEEE/ACM Transactions on Networking*, pp. 1–14, 2020.
- [76] X. Vives, “Complementarities and games: New developments,” *Journal of Economic Literature*, vol. 43, no. 2, pp. 437–479, 2005.

## References

- [77] E. E. Tsiropoulou, P. Vamvakas, and S. Papavassiliou, “Supermodular game-based distributed joint uplink power and rate allocation in two-tier femtocell networks,” *IEEE Transactions on Mobile Computing*, vol. 16, no. 9, pp. 2656–2667, 2016.
- [78] E. E. Tsiropoulou, P. Vamvakas, and S. Papavassiliou, “Joint customized price and power control for energy-efficient multi-service wireless networks via s-modular theory,” *IEEE Transactions on Green Communications and Networking*, vol. 1, no. 1, pp. 17–28, 2017.
- [79] E. Altman and Z. Altman, “S-modular games and power control in wireless networks,” *IEEE Transactions on Automatic Control*, vol. 48, no. 5, pp. 839–842, 2003.
- [80] S. Koulali, E. Sabir, T. Taleb, and M. Azizi, “A green strategic activity scheduling for uav networks: A sub-modular game perspective,” *IEEE Communications Magazine*, vol. 54, no. 5, pp. 58–64, 2016.
- [81] A. Matsui, “Best response dynamics and socially stable strategies,” *Journal of Economic Theory*, vol. 57, no. 2, pp. 343–362, 1992.
- [82] A. Nowé, P. Vrancx, and Y.-M. De Hauwere, “Game theory and multi-agent reinforcement learning,” in *Reinforcement Learning*, pp. 441–470, Springer, 2012.
- [83] G. Fragkos, P. A. Apostolopoulos, and E. E. Tsiropoulou, “Escape: Evacuation strategy through clustering and autonomous operation in public safety systems,” *Future Internet*, vol. 11, no. 1, p. 20, 2019.
- [84] K. Tsoulias, G. Palaiokrassas, G. Fragkos, A. Litke, and T. A. Varvarigou, “A graph model based blockchain implementation for increasing performance and security in decentralized ledger systems,” *IEEE Access*, vol. 8, pp. 130952–130965, 2020.
- [85] Y. Cai, G. Fragkos, E. E. Tsiropoulou, and A. Veneris, “A truth-inducing sybil resistant decentralized blockchain oracle,” in *2020 2nd Conference on Blockchain Research Applications for Innovative Networks and Services (BRAINS)*, pp. 128–135, 2020.
- [86] G. Fragkos, N. Patrizi, E. E. Tsiropoulou, and S. Papavassiliou, “Socio-aware public safety framework design: A contract theory based approach,” in *ICC 2020 - 2020 IEEE International Conference on Communications (ICC)*, pp. 1–7, 2020.

## References

- [87] G. Fragkos, C. Minwalla, J. Plusquellic, and E. E. Tsiropoulou, “Artificially intelligent electronic money,” *IEEE Consumer Electronics Magazine*, pp. 1–1, 2020.
- [88] G. Fragkos, C. Minwalla, J. Plusquellic, and E. E. Tsiropoulou, “Reinforcement learning toward decision-making for multiple trusted-third-parties in puf-cash,” in *2020 IEEE 6th World Forum on Internet of Things (WF-IoT)*, pp. 1–6, 2020.
- [89] G. Fragkos, N. Kemp, E. E. Tsiropoulou, and S. Papavassiliou, “Artificial intelligence empowered uavs data offloading in mobile edge computing,” in *ICC 2020 - 2020 IEEE International Conference on Communications (ICC)*, pp. 1–7, 2020.
- [90] G. Fragkos, S. Lebien, and E. E. Tsiropoulou, “Artificial intelligent multi-access edge computing servers management,” *IEEE Access*, vol. 8, pp. 171292–171304, 2020.
- [91] N. Patrizi, G. Fragkos, K. Ortiz, M. Oishi, and E. Tsiropoulou, “A uav-enabled dynamic multi-target tracking and sensing framework,” in *IEEE GLOBECOM*, 2020 (to appear).
- [92] M. Diamanti, G. Fragkos, E. E. Tsiropoulou, and S. Papavassiliou, “Unified user association and contract-theoretic resource orchestration in noma heterogeneous wireless networks,” *IEEE Open Journal of the Communications Society*, vol. 1, pp. 1485–1502, 2020.
- [93] M. Diamanti, G. Fragkos, E. Tsiropoulou, and S. Papavassiliou, “Resource orchestration in interference-limited small cell networks: A contract-theoretic approach,” in *International Conference on NETWORK Games, Control and Optimization (Netgcoop)*, 2020 (to appear).
- [94] N. Patrizi, G. Fragkos, E. Tsiropoulou, and S. Papavassiliou, “Contract-theoretic resource control in wireless powered communication public safety systems,” in *IEEE GLOBECOM*, 2020 (to appear).