

University of New Mexico

## UNM Digital Repository

---

Electrical and Computer Engineering ETDs

Engineering ETDs

---

Summer 6-10-2021

# Artificial Intelligent Risk-aware Autonomous Decision-Making in Resource-Constrained Computing Systems

Pavlos Athanasios Apostolopoulos  
*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

Apostolopoulos, Pavlos Athanasios. "Artificial Intelligent Risk-aware Autonomous Decision-Making in Resource-Constrained Computing Systems." (2021). [https://digitalrepository.unm.edu/ece\\_etds/509](https://digitalrepository.unm.edu/ece_etds/509)

This Dissertation 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).

Pavlos Athanasios Apostolopoulos

*Candidate*

Electrical and Computer Engineering

*Department*

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

*Approved by the Dissertation Committee:*

Eirini Eleni Tsiropoulou, Chairperson

Michael Devetsikiotis

Mark Gilmore

Symeon Papavassiliou

# Artificial Intelligent Risk-aware Autonomous Decision-Making in Resource-Constrained Computing Systems

by

**Pavlos Athanasios Apostolopoulos**

Diploma, Electrical and Computer Engineering, National Technical  
University of Athens, 2017

M.S., Computer Engineering, University of New Mexico, 2019

DISSERTATION

Submitted in Partial Fulfillment of the  
Requirements for the Degree of

**Doctor of Philosophy  
Engineering**

The University of New Mexico  
Albuquerque, New Mexico

**July, 2021**

# Dedication

*To my parents, Theodoros and Eirini, and my brothers Stelios and Lampros, for  
their unconditional support and love.*

*“Your limitation — it’s only your imagination”*



# Acknowledgments

I am grateful to everyone around me for making my Ph.D. meaningful and productive. Most importantly, I need to recognize my advisor, Dr. Eirini Eleni Tsiropoulou, for her help, guidance, and patience. She has been instrumental in shaping how I approach problems and giving me the freedom to follow my intuitions, and research that piques my interest, while at the same time she was an excellent mentor and friend. I also would like to thank Dr. Symeon Papavassiliou, who was my advisor at the National Technical University of Athens and motivated me to start my Ph.D. He never stopped advising, helping, and contributing to my research.

I also would like to thank Sandia National Laboratories, where I have worked for a long period during my Ph.D. as a research intern. These internships helped me learn how much I enjoy getting to apply many of the theoretical methods that I learned about to practical and challenging problems. Speaking of internships, I would also like to thank Facebook Inc. for providing me the opportunity to apply my research in a real-world problem that positively impacted the personalized experience of millions of users in the social network.

There also a number of professors that I have been lucky enough to take classes from, and I would like to thank them all. I need also to express my honor to the committee of this thesis, as I had the pleasure to present them my research work, and receive valuable comments, suggestions, and ideas.

I must thank many friends who have pushed me toward this goal. George, who has been with me almost from the start, and whose support, help, collaboration, and endless badgering have been a blessing. Dimitris and Marios, who were always there for me, and provided me tremendous help, especially when I started this journey. And to many more who have provided me support, inspiration, and drive over the years.

Above all, I would like to thank my family: My parents, Theodoros and Eirini, and my brothers Stelios, and Lampros, for all their support, love, and help. Without everything that you have given this would not be possible.

# Artificial Intelligent Risk-aware Autonomous Decision-Making in Resource-Constrained Computing Systems

by

**Pavlos Athanasios Apostolopoulos**

Diploma, Electrical and Computer Engineering, National Technical

University of Athens, 2017

M.S., Computer Engineering, University of New Mexico, 2019

Ph.D., Engineering, University of New Mexico, 2021

## **Abstract**

Artificial Intelligent autonomous systems are becoming increasingly ubiquitous in daily life. Mobile devices for example provide mechanical-generated intelligent support to humans, with various degrees of autonomy, and are a key part of the recent autonomous revolution. Autonomous intelligent systems aim to understand and interact with their users in a timely manner, while many of them are characterized by constrained resources. Despite that, the average person does not act in a formulaic and risk-neutral manner but instead exhibits risk-aware attitudes when performing a task that includes sources of uncertainties. When humans make decisions, they explore their surroundings, understand the emerging risks, perform actions, and evaluate their perceived outcomes. What a person characterizes as a satisfactory outcome is subjective to her own reasoning, behavior, and risk capacity. Therefore, an autonomous intelligent system should be enriched with human awareness, thus it should account for and sometimes mimic its owner's cognitive behavior and behavioral patterns, such that the latter's subjective satisfaction is optimized, and personalized service is provided.

Furthermore, the proliferation of autonomous systems, e.g., mobile or wearable devices, boosts the data volume and service demand. Each autonomous system aims to optimize its owner’s experience in a self-centric manner, and in several application domains, its actions impact the others’ experience and decision-making process generally. To this end, the users’ subjective goals generate conflicts, and the autonomous intelligent systems are expected to make decisions in non-cooperative environments. In this thesis, we investigate and introduce distributed autonomous decision-making frameworks by focusing on motivating application domains with the aforementioned challenges. We utilize Game Theory for studying the strategic interaction of the autonomous intelligent systems in non-cooperative environments and tackling the necessity of non-centralized and scalable solutions. We build autonomous intelligent decision-making agents through Reinforcement Learning, which is a popular statistical Artificial Intelligence (AI) technique for controlling unknown environments with partial, and incomplete information. Reinforcement Learning (RL) introduces the concept of an agent that learns to interact with an unknown environment by performing actions that are mainly driven by particular observations, and by evaluating the resulted feedback. We extend the regular RL setting through reward reshaping for considering the user’s risk-aware characteristics that are exhibited in real life. We incorporate Prospect Theory, which belongs to the behavioral economic subgroup, and describes how individuals make decisions between probabilistic alternatives, where risk is involved, and the probability of different outcomes is unknown. In the considered non-cooperative environments, we seek distributed solutions, thus Equilibrium points, where each autonomous intelligent agent does not have the incentive to change its own decision unilaterally.

Our investigation leads to autonomous intelligent decision-making frameworks that could serve as a step towards Artificial General Intelligence (AGI), where the computing systems learn to perform a task in a human-centric manner, thus in a similar way that the task would be completed by a person in real life.

# Contents

<b>List of Figures</b>	<b>xi</b>
<b>List of Tables</b>	<b>xiv</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Motivation . . . . .	1
1.2 Dissertation Contributions . . . . .	4
1.3 Publications . . . . .	6
1.4 Dissertation Outline . . . . .	10
<b>2 Preliminaries, and Introduced Theory</b>	<b>13</b>
2.1 Game Theory . . . . .	13
2.1.1 Mathematical Background . . . . .	14
2.1.2 Beyond Rationality: Introduced Equilibrium Points . . . . .	17
2.2 Reinforcement Learning . . . . .	21
2.2.1 Mathematical Background . . . . .	22
2.2.2 Deep Reinforcement Learning . . . . .	27
2.2.3 Learning Game Theory . . . . .	29
2.3 Prospect Theory . . . . .	33
2.3.1 Mathematical Background . . . . .	35
2.3.2 Risk-aware Decision-Making Framework . . . . .	36

<b>3</b>	<b>Mobile Edge Computing</b>	<b>39</b>
3.1	Related Work . . . . .	42
3.2	Contributions . . . . .	48
3.3	Intelligent Data Offloading in Competitive Mobile Edge Computing Market . . . . .	51
3.3.1	System Model . . . . .	52
3.3.2	MEC as a Learning System . . . . .	56
3.3.3	Autonomous Data Offloading & Price Setting . . . . .	57
3.3.4	Data Offloading and MEC Server Selection Algorithm . . . . .	62
3.3.5	Empirical Evaluation . . . . .	63
3.3.6	Summary . . . . .	70
3.4	Satisfaction-aware Data Offloading in Surveillance Systems . . . . .	71
3.4.1	System Model . . . . .	73
3.4.2	Towards IP Cameras' Satisfaction . . . . .	75
3.4.3	FAAS Movement based on Reinforcement Learning . . . . .	78
3.4.4	Empirical Evaluation . . . . .	81
3.4.5	Summary . . . . .	86
3.5	Cognitive Data Offloading in Mobile Edge Computing for Internet of Things . . . . .	87
3.5.1	System Model . . . . .	88
3.5.2	The Prospect of Data Offloading . . . . .	92
3.5.3	Optimizing Devices' Overhead . . . . .	97
3.5.4	Empirical Evaluation . . . . .	105
3.5.5	Summary . . . . .	110
3.6	Risk-aware Data Offloading in Multi-access Mobile Edge Computing .	111
3.6.1	System Model . . . . .	112
3.6.2	The Prospect of Data Offloading . . . . .	117

## *Contents*

3.6.3	Prospect-Theoretic Partial Offloading . . . . .	121
3.6.4	Towards Determining the Equilibrium . . . . .	129
3.6.5	Empirical Evaluation . . . . .	132
3.6.6	Summary . . . . .	144
<b>4</b>	<b>Redesigning Resource Management in Wireless Networks</b>	<b>145</b>
4.1	Related Work . . . . .	146
4.2	Contributions . . . . .	148
4.3	Rethinking Uplink Power Control . . . . .	149
4.3.1	Existence of ESE . . . . .	151
4.3.2	Existence of MESE . . . . .	152
4.3.3	Uniqueness and Benefits of MESE . . . . .	152
4.3.4	Algorithm & Convergence . . . . .	157
4.3.5	Empirical Evaluation . . . . .	163
4.3.6	Summary . . . . .	167
<b>5</b>	<b>Smart Technologies</b>	<b>169</b>
5.1	Learning and Game theoretic Demand Response Management . . . . .	169
5.1.1	Related Work . . . . .	170
5.1.2	Contributions . . . . .	176
5.1.3	System Model . . . . .	178
5.1.4	Modeling of the Smart Grid Network as a Distributed Learning System . . . . .	185
5.1.5	Demand Response Management: Problem Formulation & Solution . . . . .	187
5.1.6	PC-DRM Algorithm . . . . .	193
5.1.7	Empirical Evaluation . . . . .	195
5.1.8	Summary . . . . .	204

## Contents

5.2	A Risk-aware Social Cloud Computing Model . . . . .	205
5.2.1	Related Work . . . . .	205
5.2.2	Contributions . . . . .	207
5.2.3	System Model . . . . .	208
5.2.4	The Prospect of Cloud . . . . .	210
5.2.5	Optimizing Resource Allocation: Problem Formulation . . . .	212
5.2.6	Algorithm-Convergence to PNE . . . . .	214
5.2.7	Empirical Evaluation . . . . .	215
5.2.8	Summary . . . . .	221
<b>6</b>	<b>Simulation and Emulation Environment</b>	<b>222</b>
<b>7</b>	<b>Future Work</b>	<b>224</b>
7.1	A Deep Reinforcement Learning Approach for Risk-aware Orchestra- tion in MEC . . . . .	225
7.1.1	Contributions . . . . .	226
7.1.2	System Model . . . . .	229
7.1.3	Mobile Users Computation Offloading . . . . .	234
7.1.4	Mobile Users Association to MEC Servers: a Markov Deci- sion Process . . . . .	239
7.1.5	Full-Fledged Risk-aware Orchestration . . . . .	243
7.1.6	Empirical Evaluation . . . . .	244
7.1.7	Summary . . . . .	256
7.2	Future Directions . . . . .	258
<b>8</b>	<b>Conclusions</b>	<b>260</b>
	<b>References</b>	<b>262</b>

# List of Figures

2.1	Reinforcement Learning . . . . .	22
2.2	Deep Reinforcement Learning . . . . .	27
2.3	Risk-aware Reinforcement Learning . . . . .	37
3.1	SDN-enabled MEC architecture . . . . .	52
3.2	Proposed framework's operation per time slots - Homogeneous Users	64
3.3	Proposed framework's operation per time slots - Heterogeneous Users	66
3.4	Comparative Evaluation per time slots - Different Learning Rates .	68
3.5	Comparative Evaluation per time slots - Different Offloading Poli- cies . . . . .	69
3.6	Pure operation of the proposed framework . . . . .	81
3.7	Percentage of Satisfied Cameras regarding different AoIs' character- istics and Cameras' QoS prerequisites . . . . .	82
3.8	FAAS as an RL-agent: Performance Evaluation . . . . .	83
3.9	Avg. Number of FAAS's visits per AoI and AoI's QoI . . . . .	83
3.10	Scalability Analysis - Number of Cameras/Strategies . . . . .	85
3.11	Percentage of Satisfied Cameras and FAAS's Avg. Reward with respect to different offloading and FAAS's policy approaches . . . . .	86
3.12	Prospect-theoretic Data Offloading in MEC . . . . .	87
3.13	Prospect-theoretic Data Offloading in MEC . . . . .	93
3.14	Pure operation of the proposed framework . . . . .	106



## List of Figures

3.15	Scalability Evaluation - Number of Users . . . . .	107
3.16	Heterogeneous users - loss aversion parameter $k_m$ impact study . . .	108
3.17	Multi-MEC servers environment . . . . .	112
3.18	Pure operation of the proposed framework - Users' Perspective . . .	135
3.19	Pure operation of the proposed framework - MEC Servers' Perspec- tive . . . . .	135
3.20	Computation vs Communication Overhead . . . . .	138
3.21	Time & Offloaded Data - Scalability Evaluation . . . . .	138
3.22	Probability & Overhead - Scalability Evaluation . . . . .	139
3.23	Fragility under Competition . . . . .	141
3.24	Comparative Evaluation . . . . .	142
4.1	Convergence of the MEBRD algorithm in a 2-user uplink power control game . . . . .	163
4.2	Cost of various ESE points vs user ID . . . . .	164
4.3	Execution time of the MEBRD algorithm as a function of the num- ber of each user's transmission power levels . . . . .	166
4.4	Execution time of the MEBRD algorithm as a function of the num- ber of users in the system . . . . .	166
4.5	Comparison of strategy profiles for MEBRD, Energy-Efficiency Maximization and Shannon Maximization . . . . .	168
5.1	Smart Grid Network . . . . .	178
5.2	Customer's Satisfaction Function . . . . .	181
5.3	Smart Grid Network as a Learning System . . . . .	185
5.4	Pure operation of PC-DRM . . . . .	197
5.5	Companies optimal prices and perceived profits . . . . .	199
5.6	Customers' utility and overall consumed energy . . . . .	202

## *List of Figures*

5.7	Power companies' welfare and algorithms' convergence time . . . . .	203
5.8	Pure operation . . . . .	217
5.9	Scalability Evaluation - Increased number of users . . . . .	217
5.10	Loss Aversion impact study . . . . .	218
5.11	Comparative Evaluation - Impact of users' behavior . . . . .	219
7.1	DRL-SDN enabled MEC environment . . . . .	229
7.2	Mobile users' association to MEC servers: DRL-SDN agent's online learning . . . . .	249

# List of Tables

3.1	MEC servers' characteristics . . . . .	63
3.2	Execution Time for different Learning Rates . . . . .	67
3.3	Comparative Evaluation . . . . .	110
7.1	Comparative evaluation of different DRL approaches . . . . .	256

# Chapter 1

## Introduction

### 1.1 Motivation

The recent and rapid advances of Artificial Intelligence (AI) have brought us autonomous intelligent systems that have become more personalized and context-aware, serving people beyond the role of instrumental tools towards intelligent assistants and companions in our daily lives. Generally, people desire intelligent systems to perform actions and act as decision-making agents on behalf of them. In addition, people expect them to learn to perform actions through their interactions with them, and assist them to achieve their subjective goals at work [1, 2], at home [3, 4, 5], or even on the road [6, 7, 8]. The necessity of such autonomous intelligent systems is increasing rapidly alongside people's standards and demands of when an intelligent system that they interact with, it performs actions and decisions in a way they like or prefer based on their personality, behavioral patterns, and reasoning. Moreover, the needs of using and building autonomous intelligent systems that act as decision-making agents are more and more observable by the day in a variety of application domains.

In e-learning, for example, autonomous intelligent tutoring systems are needed to make decisions and track students' skill levels, needs, and weaknesses to provide personalized support and increase learning gains [9, 10, 11]. Self-driving cars interact

## *Chapter 1. Introduction*

with pedestrians and the passengers, while aim on providing transport assistance based on passengers needs and safety protocols [12, 13, 14]. The elderly population has motivated the development of autonomous intelligent systems [15] that are able to provide them company [16], assist with therapy [17], and keep them safe [18]. In addition, the use of wearable mobile devices that need to act as autonomous intelligent decision-making agents has increased dramatically, primarily for fitness, healthcare monitoring, and personalized medicare [19, 20, 21]. Global virus diseases constitute indicative examples, as they have recently motivated the development and incorporation of autonomous intelligent systems into personalized healthcare and monitoring [22]. Moreover, recommendation systems are built upon autonomous intelligent systems that track and learn user's preferences, and habits, to provide personalized content, ads, information, and generally optimize user's engagement on a platform, e.g., social network [23, 24, 25, 26, 27, 28, 29].

Furthermore, alongside the developments in AI, network technologies and their recent advances push forward the proliferation of mobile systems [30, 31, 32, 33], e.g., smartphones, smart home, smart cities, and smart grid technologies [34, 35, 36, 37, 38, 39], or the Internet of Things (IoT) [40, 41, 42] in more generalized terms. In particular, new generation networks [43, 44] have emerged novel applications such as online interactive games, video stream analysis, augmented reality, virtual reality [45, 46, 47, 48, 49], and a huge amount of sensory data that need to be processed and analyzed promptly [50, 51, 52]. In these application domains, the mobile systems aim to provide personalized services and assistance to their owners. The mobile systems interact with their users and act as autonomous intelligent systems that learn to perform actions. However, these mobile systems introduce several key challenges, as most of the time, they are equipped with constrained resources, e.g., energy availability, computing capability, which set practical limitations [53] to their general capabilities as autonomous intelligent decision-making agents. In the majority of the described applications domains, the mobile systems co-exist, while they are compet-

## *Chapter 1. Introduction*

ing for shared resources, e.g., network infrastructure. Therefore, these application domains can be mainly characterized as non-cooperative games [54], where each mobile system seeks to optimize its owner’s objective, while at the same time its actions in the shared environment impact the rest of mobile systems effort to optimize their own users’ objectives. As a result, the non-cooperative nature of these application domains, as long as the volume of data and mobile systems, highlight and motivate the interest of the recent research literature [55, 56, 57, 58] for building autonomous intelligent systems that sense their environment, learn to perform actions, and are mainly orchestrated in a distributed manner by avoiding high complexity centralized approaches with single points of failures. Game Theory [59] constitutes a powerful mathematical tool to study the interactions between the mobile systems under the described non-cooperative nature of the application domains. However, Game Theory is a branch of applied mathematics and mainly studied in economics, which makes its applicability in various application domains challenging.

In the above application domains, despite the aforementioned challenges, the resource-constrained computing systems should address the requirements of flexibility, adaptability, and intelligence. The latter stems from the fact that these environments are characterized by incomplete, dynamic, and uncertain information constraints [60]. Artificial Intelligence (AI), and especially Machine Learning (ML) [61] has established learning as a powerful approach for building personalized autonomous intelligent systems to cope with the above information constraints. In particular, Reinforcement Learning (RL) [62], which is a sub-field of Machine Learning, can extend the capabilities of mobile systems, and make them autonomous intelligent decision-making agents. However, the extension of the regular RL setting into distributed learning algorithms with low complexity and exchange of information to deal with the large volume of data and mobile systems remains a key challenge in the considered application domains [63].

Moreover, human awareness should be incorporated in the decision-making of

autonomous intelligent systems. This is mainly motivated by the fact that people in real life demonstrate risk-aware behavioral attitudes, thus they tend to exhibit risk-seeking or loss-aversion behavior when making decisions in environments with uncertainty. As a result, a real intelligent autonomous system should optimize the user's subjective objective by considering the user's risk-aware characteristics, as they constitute the user's personality and risk capacity. Towards this direction, it has been argued [64], that a simple version of expected utility theory does not properly describe human behavior. Prospect Theory [65] was proposed as an alternative to pure expected utility theory, and has emerged as a realistic model of how people make decisions under uncertainty, by successfully modeling and considering many of their standard biases [66].

To this end, the introduction and development of low complexity distributed decision-making frameworks that cope with the aforementioned challenges is highly required. In particular, these distributed decision-making frameworks should incorporate human awareness and cognition, and effectively address the requirements of intelligence and autonomy, while at the same time the constrained resources of the computing systems are properly considered. Furthermore, environments with shared resources should be further investigated, as the inherited uncertainties, and competition among the computing systems should be studied, and successfully incorporated in the decision-making process.

## **1.2 Dissertation Contributions**

In this dissertation, we cover theoretical foundations, and we propose a novel theory and scalable distributed decision-making frameworks towards transforming computing systems with constrained resources into autonomous intelligent decision-making agents. We focus our study on non-cooperative environments and application domains that are mainly characterized by incomplete, dynamic, and uncertain infor-

## *Chapter 1. Introduction*

mation, and meet the described challenges. In particular, we bridge Game Theory and Reinforcement Learning for introducing distributed low complexity learning algorithms, and we extend the regular RL setting by considering risk-aware characteristics, as these arise in humans' decision-making process in real life. We establish sufficient conditions for convex optimization, existence, and convergence to Equilibrium points in a distributed manner for the considered non-cooperative environments. Our introduced approaches are evaluated through extensive empirical studies, and we demonstrate their state-of-the-art performance in a variety of application domains. The main contributions of this dissertation are summarized below.

1. We propose low complexity distributed decision-making frameworks for computing systems with constrained computing and energy resources. In environments, where the computing systems can retrieve all the required information for supporting their decisions, we focus on establishing sufficient conditions for convex optimization, and convergence to Equilibrium points in a distributed manner. On the other hand, for environments that are characterized by partial or incomplete information, we introduce distributed learning schemes through Reinforcement Learning, and their performance and convergence are empirically evaluated.
2. We introduce additional Equilibrium points for non-cooperative games towards more realistically capturing the actual goals and needs of the computing systems in several real-world application domains. The existence, uniqueness, and convergence to them are theoretically proved and empirically demonstrated. The limitations of the commonly assumed mathematical notion of the Nash Equilibrium point are underlined, and described, while the benefits of the introduced Equilibrium points are studied.
3. Human awareness and risk-aware behavioral patterns are incorporated into the decision-making process to deal with the inherited uncertainties of envi-



ronments with shared resources of limited capabilities. The latter establishes autonomous intelligent decision-making computing systems that can sense and mimic their owner's behavior, towards optimizing the latter's subjective goals. We propose a new Reinforcement Learning setting through reward reshaping such that the human's risk-aware characteristics are properly considered.

4. The introduced decision-making frameworks are extensively analyzed theoretically and empirically through simulations in a variety of real-world applications. Finally, their state-of-the-art performance is demonstrated through comparative and scalability empirical studies.

### 1.3 Publications

All the work presented in this thesis has been published for publication in peer-reviewed journals or conferences. The work in Chapter 3 has been published in the following venues [67, 68, 69, 70]:

- 📄 Mitsis G., P.A. Apostolopoulos, E.E. Tsiropoulou, and S. Papavassiliou. "Intelligent dynamic data offloading in a competitive mobile edge computing market." *Future Internet* 11, no. 5 (2019): 118.
- 📄 P.A. Apostolopoulos, M. Torres, and E.E. Tsiropoulou. 2019. Satisfaction-aware Data Offloading in Surveillance Systems. In *Proceedings of the 14th Workshop on Challenged Networks (CHANTS'19)*. Association for Computing Machinery, New York, NY, USA, 21–26.
- 📄 P.A. Apostolopoulos, E.E. Tsiropoulou and S. Papavassiliou, "Cognitive Data Offloading in Mobile Edge Computing for Internet of Things," in *IEEE Access*, vol. 8, pp. 55736-55749, 2020.

## Chapter 1. Introduction

- 📖 P.A. Apostolopoulos, E.E. Tsiropoulou and S. Papavassiliou, "Risk-Aware Data Offloading in Multi-Server Multi-Access Edge Computing Environment," in IEEE/ACM Transactions on Networking, vol. 28, no. 3, pp. 1405-1418, June 2020.

Furthermore, an extension of the research work that is presented in Chapter 3.6 has been published in the following [52].

- 📖 P.A. Apostolopoulos, G. Fragkos, E. E. Tsiropoulou and S. Papavassiliou, "Data Offloading in UAV-assisted Multi-access Edge Computing Systems under Resource Uncertainty," in IEEE Transactions on Mobile Computing.

The above research work investigates the mobile users' data offloading in multiple ground MEC servers and UAVs, which provide remote computing capabilities, while the UAVs are located in closer proximity to the mobile users due to their flying capabilities. The UAVs' computing uncertainties that mainly stem from their limited energy availability, are properly considered in our introduced risk-aware decision-making framework (Chapter 2.3.2), which incorporates human awareness in the decision-making process.

Our research work in Chapter 4 has been published in [71]:

- 📖 P. Promponas, P.A. Apostolopoulos, E.E. Tsiropoulou and S. Papavassiliou, "Redesigning Resource Management in Wireless Networks based on Games in Satisfaction Form," 2019 12th IFIP Wireless and Mobile Networking Conference (WMNC), 2019, pp. 24-31. (⊗ Best paper award)

, and our introduced and described work in Chapter 5 has been published in the following venues [72, 73]:

- 📖 P.A. Apostolopoulos, E.E. Tsiropoulou, S. Papavassiliou, "Demand response management in smart grid networks: A two-stage game-theoretic learning-based approach". Mobile Networks and Applications. 2018 Oct 4:1-4.

## Chapter 1. Introduction

- ▮ P.A. Apostolopoulos, E.E. Tsiropoulou and S. Papavassiliou, "Risk-Aware Social Cloud Computing Based on Serverless Computing Model," 2019 IEEE Global Communications Conference (GLOBECOM), 2019, pp. 1-6.

Moreover, the following published research work is based on our introduced Equilibrium points (Chapter 2.1.2), and on our described decision-making theory of Reinforcement Learning and Game Theory (Chapters 2.2 & 2.1). The latter ones are utilized towards IoT devices' computation offloading to the MEC servers, where the former ones constitute resource-constrained computing systems that aim to fulfill their minimum Quality-of-Service (QoS) requirements in a distributed manner. The MEC servers and IoT devices act as Reinforcement Learning decision-making agents towards optimizing their cumulative perceived reward. A low complexity two-stage distributed decision-making framework is introduced.

- ▮ P.A. Apostolopoulos, E. E. Tsiropoulou and S. Papavassiliou, "Game-theoretic Learning-based QoS Satisfaction in Autonomous Mobile Edge Computing," 2018 Global Information Infrastructure and Networking Symposium (GIIS), 2018, pp. 1-5.

Furthermore, in the following published research work an evacuation planning mechanism is introduced to support the distributed and autonomous evacuation process within the operation of a Public Safety System. Our introduced decision-making framework *ESCAPE* [74], which is executed by the evacuees' resource-constrained devices, was developed based on the principles of Reinforcement Learning (Chapter 2.2), and Game Theory (Chapter 2.1).

- ▮ G. Fragkos, P.A. Apostolopoulos, E.E. Tsiropoulou, 2019. "ESCAPE: Evacuation Strategy through Clustering and Autonomous Operation in Public Safety Systems" *Future Internet* 11, no. 1: 20.

The following published research work introduces a distributed decision-making orchestration framework in a Smart City environment. Reinforcement Learning is

## Chapter 1. Introduction

utilized to model and study each human's decision-making process that selects in a distributed manner a Point of Interest (PoI), i.e., different places in a Smart City, to visit based on prior experiences [75].

📖 N. Patrizi, P.A. Apostolopoulos, K. Rael and E.E. Tsiropoulou, "Socio-Physical Human Orchestration in Smart Cities," 2019 IEEE International Conference on Smart Computing (SMARTCOMP), 2019, pp. 115-120.

The following research works were published or submitted for publication during my Ph.D. studies, where I had the opportunity and great honor to be a Research Intern for Sandia National Laboratories, and Facebook. In Sandia National Laboratories I performed research towards developing an autonomous airborne imaging system (on a UAV platform) of the National Solar Thermal Test Facility (NSTTF) concentrated solar power (CSP) plant, for identification, assessment, and correction of sun's reflectors through optimization, machine learning, and image processing techniques.

📖 J. Yellowhair, P.A. Apostolopoulos, D.E. Small, D. Novick, and M. Mann, "Development of an Aerial Imaging System for Heliostat Canting Assessments," SolarPACES 2020 (to appear).

📖 J. Yellowhair, P.A. Apostolopoulos, D.E. Small, D. Novick, and M. Mann, "UAS Imaging Path Planner for Heliostat Canting Assessments," SolarPACES 2020 (to appear).

In Facebook, I performed research on personalization for Web-based services with offline Reinforcement Learning [76]. The user's authentication after a failed login attempt was formalized as a Markov Decision Process (MDP) (Chapter 2.2), and the introduced Reinforcement Learning approach was deployed in a production system (Facebook's social network). The developed personalized user authentication led to significantly improved long-term objectives.

▮ Pavlos Athanasios Apostolopoulos, Zehui Wang, Hanson Wang, Chad Zhou, Kittipat Virochsiri, Norm Zhou, and Igor L. Markov. "Personalization for Web-based Services using Offline Reinforcement Learning," submitted to *ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. 2021*.

## 1.4 Dissertation Outline

The remainder of the thesis will be organized as follows:

Chapter 2 presents theoretical concepts of Game Theory, Reinforcement Learning, and Prospect Theory. We focus on non-cooperative environments, thus on a sub-field of Game Theory, and we describe the commonly considered notion of Nash Equilibrium, which is based on the concept of rationality. Motivated by the latter, we introduce novel theory regarding additional Equilibrium points beyond rationality, alongside the required conditions for convergence to them. We detail the key theoretical foundations of Reinforcement Learning and Deep Reinforcement Learning. The latter ones combined with Game Theory introduce several theoretical concepts and decision-making frameworks in Learning Game Theory. Moreover, we present Prospect Theory, which constitutes a behavioral model that captures humans' risk-aware attitudes. Prospect Theory is utilized by our decision-making frameworks for modeling human awareness in the decision-making process. Finally, we detail how the presented and newly introduced theory is leveraged towards developing distributed decision-making frameworks for autonomous intelligent systems. The decision-making frameworks are further theoretically investigated and extended under the considered application domains in the next Chapters, and their performance is empirically evaluated.

Motivated by the benefits and the inherited research challenges of the recently introduced network technologies, Chapter 3 investigates mobile users' computation

## *Chapter 1. Introduction*

offloading to Mobile Edge Computing (MEC) servers. Towards capturing mobile users' devices' limitations, we present low-complexity distributed decision-making frameworks that could serve as the underlying intelligence that enables autonomy for the considered computing systems. The introduced theory in Chapter 2 is further enriched concerning the unique challenges of the studied application domain. Our decision-making frameworks are extensively analyzed theoretically, and empirically evaluated and verified as well.

Chapter 4 introduces a redesigned resource management for wireless networks, based on our presented Equilibrium points and theory in Chapter 2. The latter's challenging theoretical problems regarding existence, convergence, and uniqueness are addressed, proved, and empirically shown. Towards intuitively demonstrating our research work, we leverage a highly motivating application domain and significant problem of efficient resource management for personalized services in wireless networks. This is mainly motivated by the tremendous proliferation of personalized computing devices that aim to autonomously assist their owners daily, and provide personalized services, while at the same time they are characterized by constrained resources and different levels of requirements.

Chapter 5 presents theoretical foundations of how decision-making frameworks could be leveraged as the underlying intelligence for computing systems in smart technologies. Firstly, we introduce a distributed decision-making framework that is based on the principles of Game Theory, and Reinforcement Learning, for Demand Response Management (DRM) and companies' pricing problems in Smart Grid Networks. Secondly, the recently introduced and motivating Cloud concept of Serverless Computing is investigated under uncertainty and risk-aware behavioral patterns. In particular, the task allocation to the Cloud for remote processing either on Virtual Machines (VMs) or Serverless Computing functions is studied, and a risk-aware distributed decision-making framework is presented.

Chapter 7 provides and suggests future research directions based on our insights,

## *Chapter 1. Introduction*

observations, and lessons that we acquired from this thesis. Moreover, we present theoretical formulations and indicative results of an ongoing research work that extends our introduced decision-making frameworks in Chapter 3. Specifically, we introduce a decision-making framework that captures mobile users' computation offloading problems from end-to-end in a stochastic and dynamic environment. Additionally, under the considered realistic environment, we demonstrate the state-of-the-art performance and capabilities of our approach.

Chapter 8 summarizes the contributions, and concludes the thesis.

## Chapter 2

# Preliminaries, and Introduced Theory

This chapter covers theoretical foundations regarding the three main theories that this dissertation lies on, i.e., Game Theory, Reinforcement Learning, and Prospect Theory. We start by reviewing each theory and its well-established theoretical concepts individually. In particular, regarding Game Theory we mainly focus on the concept of non-cooperative games, due to the nature of the application domains that we later study. Furthermore, we introduce key parts of our novel proposed theory, and we present them as mathematical extensions to the above well-defined mathematical concepts. The rest parts of the mathematical theory of our introduced approaches are systematically presented alongside their empirical evaluation in the studied motivating and challenging application domains. The empirical studies best support and illustrate our introduced approaches' applicability and state-of-the-art performance.

### 2.1 Game Theory

This chapter is a critical dissection of the main considered assumptions in deductive and classical Game Theory. Given the breadth and depth of Game Theory, this dissertation does not aim to be considered as an exhaustive analysis of all theoretical



concepts and made assumptions of Game Theory. In contrast, here we focus on non-cooperative games, which constitute a particular sub-concept of Game Theory. The latter choice mainly stems from the fact that in this dissertation we investigate autonomous decision-making in recently introduced research application domains that are mainly characterized by the generated conflicts among the decision-makers. In other words, we are interested in shared environments, where each autonomous decision-making agent is mainly rational, and its actions affect and get affected by the rest of autonomous decision-making agents' actions and rationality.

### 2.1.1 Mathematical Background

Game Theory was given its first mathematical formulation by John von Neuman and Oskar Morgenstern in 1944 [77] aims to model and study situations in which decision-makers interact. The result of the game-theoretic modeling of such interactions is called game, and here is denoted as  $\mathcal{G}$ . The definition of the game  $\mathcal{G}$  is given as follows:

$$\mathcal{G} = (\mathcal{N}, \mathcal{A}_n, u_n) \tag{2.1}$$

where,

- $\mathcal{N}$  is the set of individual decision-makers, who interact.
- $\mathcal{A}_n$  is the set of different actions available to each of the individuals. It is also typically called as the individual's  $n \in \mathcal{N}$  set of strategies.
- and  $u_n$  is the utility function that assigns a value to each individual for each possible combination of choices made by every individual.

Generally, the above abstract game-theoretic model does not make any assumptions about the decision-making agents' behavior, whereas each decision-making agent  $n \in \mathcal{N}$  is by definition assumed as an entity with preferences. We denote as  $a_n \in \mathcal{A}_n$  the chosen action of player  $n$ , and as  $\mathbf{a}_{-\mathbf{n}} = \{a_1, \dots, a_{n-1}, a_{n+1}, \dots, a_{|\mathcal{N}|}\}$  the

## *Chapter 2. Preliminaries, and Introduced Theory*

action profile of all other decision-makers except  $n$ . The utility function  $u_n$  in most branches of Game Theory is meant to represent the individual's  $n$  preference for each possible combination of actions made by every individual, i.e.  $u_n(a_n, \mathbf{a}_{-n})$ . As a result, in almost all game-theoretic models of Game Theory, the utility function is assumed that implicitly drives each decision-making agent's actions, and it takes numerical values on an interval case. Furthermore, the utility functions are commonly interpreted as von Neuman and Oskar Morgenstern utilities, which allows expected utility theory [64] to evaluate probability distributions over possible outcomes of the game  $\mathcal{G}$ .

Moreover, in Game Theory, the decision-making agent's  $n$  utility  $u_n$  denotes a measure of subjective psychological fulfillment, which is essentially justified by reference to some background framework that is typically only known by the individual decision-maker  $n$ . Bob and Alice, who we model as decision-making agents for the example, may have opposite subjective opinions about onions and pickles. For example, Bob may adore more the taste of onions, and with that said he associates a higher utility value to onions rather than to pickles, whereas Alice may assigns hers owns subjective utility values in the exact opposite way. The subjective utility measure of each decision-maker constitutes one of the most important features of Game Theory, as there are relatively few models where utilities that are interpreted as preferences are compared across the decision-making agents. The latter finds its roots in the social philosophy of utilitarianism and is not commonly observed in game theoretical models, whereas it could be certainly found in the research literature [78, 79].

### **Rationality and Non-cooperative Games**

In classical Game Theory and its applications in economics as long as in other non-economic related applications domains [80], the decision-making agents are assumed rational. To this end, each decision-making agent is interested to optimize its utility  $u_n$  by choosing an appropriate action  $a_n$ , and through the anticipation of the actions

## Chapter 2. Preliminaries, and Introduced Theory

of the rest of the decision-making agents, i.e.,  $\mathbf{a}_{-\mathbf{n}}$ . It is worth to be noted that in the case where all decision-makers have optimal actions regardless of what other decision-makers do, then this situation can be modeled without any appeal to Game Theory. Otherwise, in any other case, we need Game Theory!

The decision-maker's  $n$  rationality implies that the agent  $n$  selfishly aims to optimize its utility function  $u_n$ , and under this setting, the game  $\mathcal{G}$  is considered non-cooperative. In contrast, in cooperative games, the decision-making agents are willing to cooperate for the overall "good", e.g., the decision-making agents act such that the system's overall utility  $\sum_{n \in \mathcal{N}} u_n$  is optimized. However, in the investigated application domains, the latter idea contradicts with decision-makers rationality in real life, and as a result, the concept of cooperative games is out of the scope of this dissertation. In the following, we focus our analysis on non-cooperative games, and we start by introducing two of the most commonly used concepts of Equilibrium points in the research literature.

Generally, an action profile  $\mathbf{a}^* = \{a_1^*, \dots, a_n^*, \dots, a_{|\mathcal{N}|}^*\}$  constitutes an Equilibrium point for the game  $\mathcal{G}$ , if and only if, no decision-making agent  $n \in \mathcal{N}$  has the incentive to change its action  $a_n^*$  unilaterally [81]. In non-cooperative games, due to each decision-making agent's rationality, we are mainly interested in two different concepts of Equilibrium points, which constitute stable solutions for the game  $\mathcal{G}$ . The first one is the concept of Nash Equilibrium (NE) [81], which was first introduced by John Nash, who was awarded the 1994 Noble Prize in economics. The NE point is defined as follows, while its assumed key feature is that all decision-making agents make their actions selfishly and independently.

**Definition 1** *An action profile  $\mathbf{a}^* = \{a_1^*, \dots, a_n^*, \dots, a_{|\mathcal{N}|}^*\}$  is a pure Nash Equilibrium (NE) point of the non-cooperative game  $\mathcal{G}$ , if and only if, no decision-making agent  $n \in \mathcal{N}$  can improve its utility function  $u_n$  by deviating unilaterally, i.e.,*

$$u_n(a_n^*, \mathbf{a}_{-\mathbf{n}}^*) \geq u_n(a_n, \mathbf{a}_{-\mathbf{n}}^*), \forall n \in \mathcal{N}, a_n \in \mathcal{A}_n \quad (2.2)$$

Correlated Equilibrium (CE) point is another commonly used solution for a non-cooperative game  $\mathcal{G}$ , and can be regarded as a generalization of NE point's definition, as each NE point is a CE point as well, while the opposite does not always hold. Moreover, the set of CE points of the non-cooperative game  $\mathcal{G}$  is convex, whereas the corresponding NE points are isolated points at the *extrema* of this set [60]. To this end, CE points are structurally simpler than that of NE points, while due to the convexity property of the set of CE points, fairness between the decision-making agents  $\mathcal{N}$  could be well addressed. In particular, the CE point is defined as follows:

**Definition 2** *A probability distribution  $\rho$  over  $\mathcal{A}_1 \times \cdots \times \mathcal{A}_{|\mathcal{N}|}$  is a Correlated Equilibrium (CE) point, if and only if, for all  $n \in \mathcal{N}$ , for all actions  $a_n \in \mathcal{A}_n$ , and all alternative actions  $a'_n \in \mathcal{A}_n$ , it holds true that:*

$$\sum_{\mathbf{a}_{-n} \in \mathcal{A}_{-n}} \rho(a_n, \mathbf{a}_{-n}) (u_n(a_n, \mathbf{a}_{-n}) - u_n(a'_n, \mathbf{a}_{-n})) \geq 0 \quad (2.3)$$

, where  $\mathcal{A}_{-n} = \mathcal{A}_1 \times \cdots \times \mathcal{A}_{n-1} \times \mathcal{A}_{n+1} \cdots \times \mathcal{A}_{|\mathcal{N}|}$ , and  $\rho(a_n, \mathbf{a}_{-n})$  represents the probability that the decision-maker  $n$  takes action  $a_n$ , while all other decision-making agents take action profile  $\mathbf{a}_{-n}$ . In other words, the above definition implies that at the CE point, if the decision-making agent  $n$  takes the action  $a_n$ , then choosing any other action  $a'_n \neq a_n$  can not yield to a higher expected utility for the decision-maker  $n$ .

### 2.1.2 Beyond Rationality: Introduced Equilibrium Points

In this chapter, we introduce additional game-theoretic models and Equilibrium points that are differentiated from the key idea of rationality, which is adopted in the concepts of NE and CE points. The advantages of these Equilibrium points are multi-fold, as the properties of their existence are less restrictive than the commonly adopted notions of NE and CE points, while at the same time they better reflect the actual decision-makers' subjective goals in particular application domains. In particular, the notion of rationality implies that each individual decision-maker is

## Chapter 2. Preliminaries, and Introduced Theory

interested to selfishly optimize its perceived utility function, whereas in several non-cooperative environments each individual's subjective goals could be better expressed by the satisfaction of the individual's utility constraint rather than the latter's optimization [82]. The latter idea was originally introduced as a mathematical theoretical concept in [83], and in [84], and for a particular class of conditions in pure strategies for the case of a finite set of actions, i.e.,  $\mathcal{A}_n, \forall n \in \mathcal{N}$ . Later, the concept was formulated in terms of a fixed point inclusion for the case of pure strategies in [85], in the context of both finite actions, convex, and closed action sets. Here, we introduce a novel game formulation where in contrast with the concept of pure rationality, as this was introduced in Chapter 2.1.1, each decision-maker is characterized by a *minimum feeling* of rationality, thus it aims to satisfy its utility constraint selfishly. In general, a game in satisfaction form is defined as follows:

$$\hat{\mathcal{G}} = (\mathcal{N}, \mathcal{A}_n, u_n, f_n) \quad (2.4)$$

where as before (Equation 2.1),

- $\mathcal{N}$  is the set of individual decision-makers, who interact.
- $\mathcal{A}_n$  is the set of different actions available to each of the individuals.
- $u_n$  is the utility function that assigns a value to each individual for each possible combination of choices made by every individual.
- $f_n$  is the correspondence, i.e.,  $f_n : \mathcal{A}_{-\mathbf{n}} \rightarrow 2^{\mathcal{A}_n}$  that determines the set of actions of decision-maker  $n$  which allows its satisfaction given the actions taken by all other decision-makers. In other words  $f_n(\mathbf{a}_{-\mathbf{n}}) = \{a_n \in \mathcal{A}_n : u_n(a_n, \mathbf{a}_{-\mathbf{n}}) \geq \bar{u}_n\}$ , where  $\bar{u}_n$  is the decision-maker's  $n$  utility constraint, i.e., minimum utility prerequisite.

In a satisfaction non-cooperative game, each decision-maker aims to satisfy its utility constraint careless of whether other decision-makers can satisfy or not their utility

## Chapter 2. Preliminaries, and Introduced Theory

constraints. Under this setting, we are seeking a stable solution, i.e., an Equilibrium point  $\mathbf{a}^+$ , where each decision-maker  $n \in \mathcal{N}$  does not have the incentive to change its action unilaterally, as its utility constraint is satisfied. This Equilibrium point is called *Satisfaction Equilibrium* (SE) point and is defined as follows:

**Definition 3** *An action profile  $\mathbf{a}^+$  is an SE point for the game  $\hat{\mathcal{G}}$ , if and only if:*

$$\forall n \in \mathcal{N}, a_n^+ \in f_n(\mathbf{a}_{-\mathbf{n}}^+) \quad (2.5)$$

, where  $\mathbf{a}_{-\mathbf{n}}^+ = (a_1^+, \dots, a_{n-1}^+, a_{n+1}^+, \dots, a_{|\mathcal{N}|}^+)$ . It is worth to be noted, that since each decision-maker aims to be satisfied, none of them has a particular interest of changing its current action once they are at the SE.

In order to study the existence of an SE in the game  $\hat{\mathcal{G}} = (\mathcal{N}, \mathcal{A}_n, u_n, f_n)$ , let the correspondence  $\mathcal{F} : \mathcal{A} \rightarrow 2^{\mathcal{A}}$ , where  $\mathcal{A} = \mathcal{A}_1 \times \dots \times \mathcal{A}_{|\mathcal{N}|}$ , be defined as follows:

$$\mathcal{F}(\mathbf{a}) = f_1(\mathbf{a}_{-1}) \times \dots \times f_{|\mathcal{N}|}(\mathbf{a}_{-|\mathcal{N}|}) \quad (2.6)$$

then, a SE exists if and only if:

$$\exists \mathbf{a} \in \mathcal{A} : \mathbf{a} \in \mathcal{F}(\mathbf{a}) \quad (2.7)$$

This formulation in combination with the existing fixed point theorems [86] provide sufficient conditions and properties for the existence of the SE [87]. In particular, using the fixed point theorem of Knaster and Tarski [88], the following theorem holds true:

**Theorem 1** *Consider the game  $\hat{\mathcal{G}}$  in satisfaction form  $\hat{\mathcal{G}} = (\mathcal{N}, \mathcal{A}_n, f_n)$ , and let the set  $\mathcal{A}$  have a binary relation denoted by  $\preceq$ . Let also*

- $\mathcal{V} = (\mathcal{A}, \preceq)$  be a complete lattice.
- $\mathcal{F}(\mathbf{a})$  be non-empty  $\forall \mathbf{a} \in \mathcal{A}$ .

## Chapter 2. Preliminaries, and Introduced Theory

- the correspondence  $\mathcal{F}$  in Equation 2.6 satisfies that  $\forall(\mathbf{a}, \mathbf{a}') \in \mathcal{A}^2$ , such that  $\mathbf{a} \preceq \mathbf{a}'$ , it holds that

$$\forall(\mathbf{b}, \mathbf{b}') \in \mathcal{F}(\mathbf{a}) \times \mathcal{F}(\mathbf{a}'), \mathbf{b} \preceq \mathbf{b}' \quad (2.8)$$

To this end, the game  $\mathcal{G}$  has at least one SE in pure strategies. It is worth to be noted, that in order the above theorem to hold true, then it is required that  $\forall \mathbf{a} \in \mathcal{A}$  the set  $\mathcal{F}$  is non-empty, i.e.,

$$\forall n \in \mathcal{N}, \forall \mathbf{a}_{-\mathbf{n}} \in \mathcal{A}_{-\mathbf{n}}, \exists a_n \in \mathcal{A}_n : a_n \in f_n(\mathbf{a}_{-\mathbf{n}}). \quad (2.9)$$

As a result, there could exist multiple action profiles  $\mathbf{a}^+$  satisfying decision-makers utility constraints. For that reason, a representative example among all different SE points is the *Efficient Satisfaction Equilibrium* (ESE) point, at which each decision-maker achieves its minimum utility prerequisites via being simultaneously penalized with the minimum cost. The latter is mainly motivated by the fact that in several application domains, a decision-maker's performed action leads to a perceived utility function value, while it comes with an associated cost at the same time. To capture the notion of decision-maker's  $n$  penalty and effort associated with a performed action  $a_n \in \mathcal{A}_n$ , the concept of the cost function is introduced. For all  $n \in \mathcal{N}$ , the cost function  $c_n : \mathcal{A} \rightarrow [0, 1]$  satisfies the condition:  $c_n(a_n) < c_n(a'_n), \forall (a_n, a'_n \in \mathcal{A}_n^2)$ , if and only if,  $a_n$  introduces a lower cost for decision-maker  $n$  than action  $a'_n$ . The ESE point is defined as follows:

**Definition 4** An action profile  $\mathbf{a}^\diamond$  is an ESE point for the game  $\hat{\mathcal{G}}$ , with cost functions  $c_n, \forall n \in \mathcal{N}$ , if and only if:

$$\forall n \in \mathcal{N}, a_n^\diamond \in f_n(\mathbf{a}_{-\mathbf{n}}^\diamond) \quad (2.10)$$

$$\forall n \in \mathcal{N}, \forall a_n \in f_n(\mathbf{a}_{-\mathbf{n}}^\diamond), c_n(a_n) \geq c_n(a_n^\diamond) \quad (2.11)$$

Similarly, there could exist multiple action profiles  $\mathbf{a}^\diamond$  that constitute ESE points for the game  $\mathcal{G}$  [87, 89]. Here, we introduce another Equilibrium point of special interest

that allows us to rank the different ESE points of the game  $\mathcal{G}$ . In particular, we call that Equilibrium point as *Minimum Efficient Satisfaction Equilibrium* (MESE) point. At the MESE, all decision-makers satisfy their utility constraints, with the minimum cost for themselves, as well as with the minimum total cost from the system's perspective. The MESE point is defined as follows:

**Definition 5** *An action profile  $\mathbf{a}^\dagger$  is an MESE point for the game  $\hat{\mathcal{G}}$ , with cost functions  $c_n, \forall n \in \mathcal{N}$ , and a set of action profiles that are ESE points  $\mathcal{E} = \{\mathbf{a} \in \mathcal{A} : \forall n \in \mathcal{N}, a_n \in f_n(\mathbf{a}_{-\mathbf{n}}), \forall a'_n \in f_n(\mathbf{a}_{-\mathbf{n}}), c_n(a'_n) \geq c_n(a_n)\}$ , if and only if:*

$$\forall n \in \mathcal{N}, a_n^\dagger \in f_n(\mathbf{a}_{-\mathbf{n}}^\dagger) \quad (2.12)$$

$$\forall n \in \mathcal{N}, \forall a_n \in f_n(\mathbf{a}_{-\mathbf{n}}^\dagger), c_n(a_n) \geq c_n(a_n^\dagger) \quad (2.13)$$

$$\forall e \in \mathcal{E}, \sum_{n \in \mathcal{N}} c_n(e_n) \geq \sum_{n \in \mathcal{N}} c_n(a_n^\dagger) \quad (2.14)$$

The uniqueness and existence of MESE are investigated under the setting of a particular application domain, i.e., uplink power control in wireless networks, in Chapter 4.

## 2.2 Reinforcement Learning

Reinforcement Learning [62] dates back to the early days of cybernetics, and work in statistics, psychology, neuroscience, and computer science. In the last five to ten years, Reinforcement Learning (RL) has experienced dramatic growth in attention and interest in the Machine Learning and Artificial Intelligence communities, due to promising and motivating results in a variety of application domains like controlling continuous systems in robotics [90], playing Go [91], Atari [92], and competitive video games [93, 94]. RL is the study of how an agent can interact with its dynamic and unknown environment to learn a policy, which maximizes expected cumulative



reward for a task, or as understood by Sutton and Barto, RL is a fusion of the trial-and-error "law-of-effect" tradition [95]. RL is considered as a third Machine Learning paradigm, alongside Supervised and Unsupervised Learning [62, 61]. In other words, RL is different from Supervised Learning, which describes the learning on a training set of labeled examples provided by a knowledgeable external supervisor, as in RL's interactive problem, it is often impractical to obtain examples of desired behavior that are both correct and representative of all the situations in which the agent has to act. Moreover, RL is different from Unsupervised Learning, as RL is trying to maximize a reward signal instead of trying to find a hidden structure.

### 2.2.1 Mathematical Background

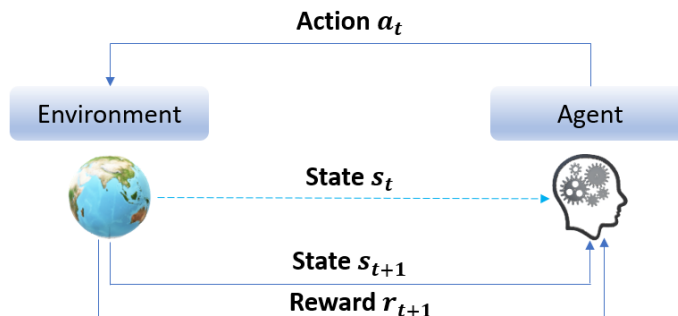


Figure 2.1: Reinforcement Learning

Reinforcement Learning (RL) seeks to control an interactive dynamic environment (Figure 2.1), where at each discrete time slot,  $t = 0, 1, \dots$ , the RL agent observes the environment's state  $s_t$  and responds with action  $a_t$ , while the environment responds with an associated reward  $r_{t+1}$  and transitions into the next state  $s_{t+1}$ . The RL agent aims to determine an optimal policy for performing actions on each given state, such that the perceived cumulative reward over the long-time horizon is maximized. This problem is known as a sequential decision-making problem [96], and the model of Markov Decision Process [97], provides a simple yet powerful method for modeling states, actions, states transitions, and rewards received in each state

## Chapter 2. Preliminaries, and Introduced Theory

transition. For that reason, most of the RL work uses MDPs to model the underlying dynamics of the agent's environment. In particular, the RL agent's environment can be defined by an MDP  $(\mathbb{S}, \mathbb{A}, r, \rho, \gamma)$ , where  $\mathbb{S}$  is the state space with  $s_t \in \mathbb{S}$ ,  $\mathbb{A}$  is the action space with  $a_t \in \mathbb{A}$ ,  $r(s_t, a_t)$  is the reward function,  $\rho(\cdot|s_t, a_t)$  is a conditional probability distribution of the form  $\rho(s_{t+1}|s_t, a_t)$  (environment's dynamics), and  $\gamma \in (0, 1]$  is a scalar discount factor. The probability distribution  $\rho(s_{t+1}|s_t, a_t)$  is also called *transition probability*, and it essentially represents the probability that the state of the environment in the next time slot  $(t + 1)$  is transformed to  $s_{t+1}$  from  $s_t$  given that the RL agent performs the action  $a_t$ .

If complete information of the MDP is known, i.e.,  $\rho(s_{t+1}|s_t, a_t), \forall (s_{t+1}, s_t) \in \mathbb{S}^2, a_t \in \mathbb{A}$ , then the agent's sequential decision-making problem can be effectively solved by using well-defined classic planning techniques, e.g., *Dynamic Programming* [98] or *Teleo-Reactive* [99]. However, in real-world problems the decision-maker's environment's underlying dynamics are unknown. In these problems, classic planning techniques meet significant difficulties in solving them, and RL techniques are thus developed to combat the *incomplete knowledge* challenge. Although RL techniques are based on *Dynamic Programming*, however, they offer two important advantages over *Dynamic Programming*. Firstly, RL methods perform *policy evaluation* and *policy improvement* at the same time simultaneously [62]. In particular, the RL agent can evaluate its current learned policy, i.e., *policy evaluation*, by using it to interact with the environment, while the perceived feedback is used as a correction signal to further improve its learned policy, i.e., *policy improvement*. As a result, RL methods allow the agent to focus only on the policies that are important and spend less time on the less promising ones. Secondly, RL can be empowered with powerful function approximators, e.g., neural networks [100] or till coding [62], to represent the agent's information, so that the experiences learned in one state can be used in other "close" states as well. The latter helps RL to achieve generalization, e.g., over the state-action space, and scale better.

## Chapter 2. Preliminaries, and Introduced Theory

The RL agent's goal in an MDP  $(\mathbb{S}, \mathbb{A}, r, \rho, \gamma)$ , as this was defined above, is to determine autonomously, and by only exploiting its experienced interactions with the environment, an optimal policy  $\pi^*$  for performing actions at any given state. Thus, an optimal policy of the form  $\pi^*(a_t|s_t)$ , where for each state  $s_t$  defines a distribution over possible actions  $a_t \in \mathbb{A}$ . The policy should maximize cumulative reward over time, i.e., an expectation under the environment's dynamics over the long-time horizon:

$$J(\pi) = \mathbb{E} \left[ \sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \right] \quad (2.15)$$

$$s_0 = s, a_0 = a, s_t \sim \rho(\cdot|s_{t-1}, a_{t-1}), a_t \sim \pi(\cdot|s_t)$$

The discount factor essentially indicates how much attention should be paid to the instant reward and the long-term reward respectively. In other words, as  $\gamma \rightarrow 0$ , the importance of the current immediate reward becomes more important, whereas as  $\gamma \rightarrow 1$ , future rewards that contribute to the overall cumulative reward are taken into account more and more. Furthermore, based on Equation 2.15 the Reinforcement Learning objective problem can be viewed as an optimization over the space of policies  $\pi$ :

$$\pi^* = \arg \max_{\pi} \mathbb{E} \left[ \sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \right] \quad (2.16)$$

### Value-based Reinforcement Learning

The *value function*  $V^\pi(s)$  represents the expected discounted cumulative reward that will be received by the RL agent by following policy  $\pi$ , and starting from state  $s$ :

$$V^\pi(s) = \mathbb{E}[r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \cdots | s_t = s, \pi] \quad (2.17)$$

where,  $r_t = r(s_{t+1}|s_t, a_t)$ , or in other words  $r_t = r(s_t, a_t)$ , is the immediate reward received in time step  $t$ ,  $s_t$  is the state visited in  $t$ , and  $a_t \sim \pi(a_t|s_t)$  is the RL agent's performed action. Besides the *value function* defined in Equation 2.17, also known

## Chapter 2. Preliminaries, and Introduced Theory

as *V-values*, another most widely used *value function* in RL, is the *state-action-value function*, referred as *Q-values*, which expresses the expected discounted cumulative reward obtained by the RL agent after performing action  $a_t$  in state  $s_t$ , and thereafter following policy  $\pi$ :

$$Q^\pi(s, a) = \mathbb{E}[r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \cdots | s_t = s, a_t = a, \pi] \quad (2.18)$$

, where  $a_t$  is the RL agent's performed action in state  $s_t$ . A significant result about *value functions*, is that the *V-values* satisfy the Bellman equation for a fixed policy  $\pi$  [62]:

$$V^\pi(s) = \sum_{s'} \rho(s'|s, a) \cdot [r(s'|s, a) + \gamma V^\pi(s')] \quad (2.19)$$

, where  $a \sim \pi(a|s)$ , thus the *value function* of the state  $s$  can be recursively defined by the *value function* of the next state  $s'$ . To this end, the problem of obtaining the current state's *value function* for a fixed policy  $\pi$ , can be solved by obtaining the next state's *value function*, and the immediate reward, which can essentially be received by the RL agent's environment. As a result, the Bellman's equation breaks the RL agent's sequential decision-making problem into sub-problems, and essentially it witnesses that the sequential decision-making problem in the MDP has optimal substructures, i.e., an optimal solution can be constructed efficiently from optimal solutions of its sub-problems [98, 101].

Therefore, Bellman's equation guarantees the convergence property of the *value function*, thus Equation 2.19 can be rewritten as follows:

$$T^\pi V^\pi(s) = V^\pi(s) \quad (2.20)$$

, where  $T^\pi$  is the Bellman operator underlying the fixed policy  $\pi$  such that:

$$(T^\pi V)(s) = \sum_{s'} \rho(s'|s, a) \cdot [r(s'|s, a) + \gamma V(s')] \quad (2.21)$$

, where  $a \sim \pi(a|s)$ , and Equation 2.21 defines a linear system of equations with respect to  $V^\pi$ , and  $T^\pi$  is an *affine linear operator* [102], and if  $0 < \gamma < 1$ , then  $T^\pi$  is

## Chapter 2. Preliminaries, and Introduced Theory

a *maximum-norm contraction* and the fixed-point equation  $T^\pi V^\pi(s) = V^\pi(s)$  has a unique solution  $\forall s \in \mathbb{S}$  [103].

Furthermore, for an optimal policy  $\pi^*$  (Equation 2.16), the *value function* is defined as follows:

$$V^*(s) = \sum_{a'} p^{\pi^*}(a|s) \sum_{s'} \rho(s'|s, a) \cdot [r(s'|s, a) + \gamma V^*(s')] \quad (2.22)$$

, where  $a \sim \pi(a|s)$ , and  $p^{\pi^*}(a|s)$  is the probability of performing action  $a$  in state  $s$  in the space of optimal policies. Moreover, the Bellman equation holds for the *Q-values* as well, thus:

$$Q^\pi(s, a) = \sum_{s'} \rho(s'|s, a) \cdot [r(s'|s, a) + \gamma Q^\pi(s', a')] \quad (2.23)$$

, where  $a \sim \pi(a|s)$ , and similarly for an optimal policy  $\pi^*$  holds true that:

$$Q^*(s, a) = \sum_{s'} \rho(s'|s, a) \cdot \left[ r(s'|s, a) + \gamma \max_{a'} Q(s', a') \right] \quad (2.24)$$

Furthermore, Equation 2.24 can be rewritten with in respect to the expected *Q-value* of the next state as follows:

$$Q^*(s, a) = r(s, a) + \gamma \mathbb{E}_{s' \sim \rho(s'|s, a)} \left[ \max_{a'} Q^*(s', a') \right] \quad (2.25)$$

To this end, the RL problem seeks an optimal policy  $\pi^*$  such that  $Q^*(s, a) \geq Q^\pi(s, a)$ ,  $\forall \pi, s \in \mathbb{S}, a \in \mathbb{A}$ . The latter equation indicates that an optimal policy can be derived by having the optimal *Q-values*, i.e.,  $Q^*$ , as follows:

$$\pi^*(s) = \delta(\arg \max_{a \in \mathbb{A}} Q^*(s, a)), \forall s \in \mathbb{S} \quad (2.26)$$

, where  $\delta$  operator could express either a deterministic policy, or any other stochastic policy, e.g.,  $\epsilon$ -greedy policy [62]. As a result, *Value-based Reinforcement Learning* essentially aims to estimate the optimal *state-action-value* function, i.e.,  $Q^*(s, a) \forall s \in \mathbb{S}, a \in \mathbb{A}$ , as then an optimal policy  $\pi^*$ , e.g.,  $\epsilon$ -greedy strategy, can be defined based on the optimal *Q-values*, and as per Equation 2.26.

**Q-Learning** [62] constitutes the most representative *Value-based* RL approach, and uses a Q-table to store the *Q-values*, i.e.,  $Q(s, a), \forall s \in \mathbb{S}, a \in \mathbb{A}$ . Q-learning aims to estimate the optimal Q-values, i.e.,  $Q^*(s, a), \forall s \in \mathbb{S}, a \in \mathbb{A}$ . For that reason, the RL agent alternates between two phases, i.e., *Q-values* updates, and further interaction with the environment. In particular, the RL agent at each time slot  $t$  observes the environment's state  $s_t$ , and performs an action  $a_t$ , with typically an  $\epsilon$ -greedy policy, i.e., version of  $\pi(a_t|s_t) = \delta(a_t = \arg \max_{a_t \in \mathbb{A}} Q(s_t, a_t))$ . The perceived reward  $r(s_t, a_t)$  and environment's next state  $s_{t+1}$  are used for the update of the Q-values at each training iteration, as follows [62]:

$$Q(s_t, a_t) = Q(s_t, a_t) + \alpha \left[ r(s_t, a_t) + \gamma \max_{a_{t+1}} Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t) \right] \quad (2.27)$$

The described iterative alternation between Q-values update, i.e., *policy improvement*, and RL agent's interaction with the environment, i.e., *policy evaluation*, lead the stored Q-values to approximate the optimal Q-values, i.e.,  $Q^*(s_t, a_t), \forall s_t \in \mathbb{S}, a_t \in \mathbb{A}$ , and as a result the RL agent to establish an optimal policy  $\pi^*$ .

### 2.2.2 Deep Reinforcement Learning



Figure 2.2: Deep Reinforcement Learning

Over the past few years, RL has become even more popular due to its success in addressing sequential decision-making problems through the integration of deep

learning techniques. This combination is typically called Deep Reinforcement Learning (DRL) [104, 105], and it has provided human problem-solving capabilities even in high dimensional spaces for complicated tasks. For instance, DRL has been successfully applied and address complicated problems in real-world applications such as robotics [106], self-driving cars [107], finance [108], and smart grids [109]. A key part that has pushed the success and applicability of DRL in complicated problems, is the recent advances in deep learning [110, 111, 112, 113, 114] that have brought us expressive deep neural networks. The latter are utilized as function approximators and allow DRL to deal with high dimensional state-action spaces, while at the same time perceive enriched and intelligent generalization, e.g., over the state-action space.

The main motivation behind DRL is the fact that in several real-world complex environments that are characterized by high dimensional state and action spaces, it is either impractical or even infeasible to use regular RL techniques. For instance, regular Q-learning uses an Q-table to store all the *Q-values* for every possible state-action pair. The latter would not be possible in a complex environment with high dimensional state-action space. For that reason, DRL uses deep neural networks as function approximators in order to overcome the problem of the high dimensional state and action spaces, and achieve effective generalization. In particular, a deep neural network (Figure 2.2) with parameters  $\theta$  can be used for parameterizing the DRL agent's policy  $\pi_\theta$ , in the case where its policy is directly optimized through Equation 2.16, i.e., *Policy Gradient* methods [115], or a *state-action-value* function  $Q(s_t, a_t; \theta)$ , in the case of *Value-based Reinforcement Learning* (Chapter 2.2.1).

**Deep Q-Network (DQN)** [92] is the most representative framework of DRL, and constitutes the DRL alternate of the regular Q-learning. Specifically, the *state-action-value* function  $Q(s_t, a_t)$  is approximated with a deep neural network with parameters  $\theta$ , i.e.,  $Q(s_t, a_t; \theta)$ . At each time slot  $t$  the DQN receives as input the environment's state  $s_t \in \mathbb{S}$ , and outputs the state-action Q-values  $Q(s_t, a_t; \theta), \forall a_t \in \mathbb{A}$ .

As in the regular Q-learning, the DRL agent alternates between improving the parameters  $\theta$  of the DQN, and data collection via interaction with the environment, and through a stochastic policy, thus  $\pi(a_t|s_t) = \delta(a_t = \arg \max_{a_t \in \mathcal{A}} Q(s_t, a_t; \theta))$ , e.g.,  $\epsilon$ -greedy strategy. In contrast with the regular Q-learning, which takes a single gradient step (Equation 2.27) towards minimizing the difference between the left-hand and right-hand side of Equation 2.25, in DQN a replay buffer of the form  $\mathcal{D} = \{(s_t, a_t, s_{t+1}, r_{t+1})\}$ , where  $r_{t+1} = r(s_t, a_t)$ , is utilized for storing the DRL agent's interactions with the environment. To this end, at each training iteration, the parameters  $\theta$  of the DQN are updated by randomly sampling a batch of interactions of the form  $(s_t, a_t, s_{t+1}, r_{t+1})$ , and through back-propagation and gradient descent [116] with respect to the *Temporal Difference Loss* function  $\mathcal{L}_i(\theta_i)$ , which similarly with Equation 2.27 is defined as follows:

$$\mathcal{L}_i(\theta_i) = \mathbb{E}_{s_t, a_t \sim \mathcal{D}} [(y_i - Q(s_t, a_t; \theta_i))^2] \quad (2.28)$$

Here  $y_i = \mathbb{E}_{s_{t+1} \sim \mathcal{D}} \left[ r(s_t, a_t) + \gamma \max_{a_{t+1}} Q(s_{t+1}, a_{t+1}; \theta_i^-) \right]$ ,  $\theta_i$  are the parameters of the DQN at the iteration  $i$  of the learning procedure, and for better stability in the learning procedure the  $\theta_i^-$  are frozen parameters that are used for estimating the target values, i.e.,  $y_i$ , through another DQN often called the target DQN. The  $\theta^-$  parameters of the target DQN are periodically updated to the latest parameters  $\theta$  of the DQN.

### 2.2.3 Learning Game Theory

The existence of Equilibrium points, as these were introduced in Chapter 2.1, is the first key part of our introduced autonomous decision-making and game theoretic solutions. However, finding these Equilibrium points is the next challenging key part that needs to be tackled, given a shared non-cooperative environment. The latter constitutes a challenging problem, especially in non-cooperative environments that are characterized by incomplete, dynamic, and uncertain information, as the



## Chapter 2. Preliminaries, and Introduced Theory

application domains that we study in this dissertation. For that reason, in this chapter we review Learning Game Theory (LGT) [117, 118].

LGT assumes that the decision-makers can learn over time about the game and the behavior of others. To this end, LGT describes each decision-maker's process of learning as a sequential decision-making problem, where the decision-maker  $n \in \mathcal{N}$  learns to perform actions  $a_n \in \mathcal{A}$  such that its objective goals, e.g., utility maximization, utility prerequisite's satisfaction, are perceived. As a result, Reinforcement Learning, as this was introduced in Chapters 2.2.1 and 2.2.2 could be utilized, such that each decision-maker  $n \in \mathcal{N}$  acts as an RL agent, and learns an optimal policy  $\pi^*$  that maximizes its perceived cumulative long-term reward (Equation 2.15). For example, under the setting of a non-cooperative game  $\mathcal{G}$ , where each decision-maker behaves rational, the decision-maker's  $n$  perceived reward  $r_n$  after performing action  $a_n$ , could be its perceived utility function value, i.e.,  $u_n$ . In the following, we maintain the RL notation, as this was introduced in Chapter 2.2.1, thus we refer to decision-maker's  $n$  perceived reward as  $r_n$ , which generally could be reshaped appropriately concerning the decision-maker's objective goals.

To this end, the purpose of this Chapter is to bridge Game Theory and Reinforcement Learning under the perspective of LGT. For that reason, we introduce distributed RL algorithms that are mainly characterized by low complexity and constitute candidate solutions for the case where the decision-maker is coupled with a resource-constrained computing system. These techniques could be considered as additional RL techniques, like the ones that were introduced in Chapters 2.2.1 and 2.2.2, and could be utilized such that the decision-maker's goals are achieved, and convergence to Equilibrium points is perceived through learning.

The un-neglectable information constraints that are included in real-world application domains can be mainly classified into the following three categories [60]:

- **Incomplete:** a decision-maker has only partial information about the environment, and it knows only the individual information and nothing about the

rest decision-makers.

- **Dynamic:** the environment's states are time-varying and unknown, thus the environment's dynamics (Chapter 2.2.1) are unknown.
- **Uncertain:** the decision-maker's observations are not equal to the actual values, thus the observed values are corrupted by noise.

Here, we introduce a general procedure of distributed learning of Equilibrium points, where each decision-maker behaves as an RL-agent, thus it learns an optimal policy by repeatedly performing actions and by observing the environment's feedback, e.g., state transitions, perceived rewards. For that reason, we assume that the decision epoch is discrete and divided into slots of equal length. Furthermore, by using the notation introduced in Chapters 2.1 and 2.2.1, we denote as  $a = (a_1(t), \dots, a_{|\mathcal{N}|}(t))$  the action profile, and as  $r(t) = (r_1(t), \dots, r_{|\mathcal{N}|}(t))$  the perceived rewards of the decision-makers in slot  $t$ , respectively. Generally, the learning procedure can be expressed as follows:

$$a_n(t+1) = \mathcal{R}[a_n(1), \dots, a_n(t); r_n(1), \dots, r_n(t)] \quad (2.29)$$

, where  $\mathcal{R}$  is the learning rule. In particular, different learning algorithms correspond to different update rules, i.e.,  $\mathcal{R}$ , and the already introduced RL techniques in Chapters 2.2.1, and 2.2.2 can be directly mapped to learning rules.

**Stochastic Learning Automata (SLA)** [119] could be essentially considered as a distributed policy-based RL algorithm [115], where each decision-maker's  $n \in \mathcal{N}$  policy at slot  $t$  is approximated through a probability distribution  $\pi_n(t) \in \Delta(\mathcal{A}_n)$ , where  $\Delta(\mathcal{A}_n)$  is the set of probability distributions over the available action set  $\mathcal{A}_n$ . The update rule  $\mathcal{R}$  of the SLA, is based on the idea that if an action is selected by the decision-maker  $n \in \mathcal{N}$  at slot  $t$ , and a positive reward value  $r_n(t)$  is received, then the probability of choosing this action in the next iteration of the learning procedure increases with regards to the magnitude of the perceived reward. The commonly used

## Chapter 2. Preliminaries, and Introduced Theory

update rule in the research literature is the *linear reward-inaction* (LRI) [62, 120], which is defined as follows:

$$\begin{aligned}\pi_{nm}(t+1) &= \pi_{nm}(t) + \beta \hat{r}_n(t)(1 - \pi_{nm}(t)), m = a_n(t) \\ \pi_{nm}(t+1) &= \pi_{nm}(t) - \beta \hat{r}_n(t)\pi_{nm}(t), m \neq a_n(t)\end{aligned}\tag{2.30}$$

where,  $\hat{r}_n(t)$  is the decision-maker's  $n$  perceived normalized reward, e.g., utility value, in slot  $t$ , and  $\beta$  is the step size. Despite LRI's analytical tractability, several other forms of update rules have been used in research literature, e.g., *linear reward-penalty*, *linear reward- $\epsilon$ -penalty* [121]. Although, the convergence of the SLA algorithm for general non-cooperative games is hard to guarantee, it asymptotically converges to the NE point of potential games, where the definition of a potential game is given as follows:

**Definition 6** A game  $\mathcal{G}$  is an (exact) potential game if there exists a potential function  $\phi : \mathcal{A}_1 \times \cdots \times \mathcal{A}_n \rightarrow R$ , such that  $\forall n \in \mathcal{N}$ ,  $\forall a_n \in \mathcal{A}_n$ , and  $\forall a'_n \in \mathcal{A}_n$ , it holds true that:

$$u_n(a_n, \mathbf{a}_{-n}) - u_n(a'_n, \mathbf{a}_{-n}) = \phi(a_n, \mathbf{a}_{-n}) - \phi(a'_n, \mathbf{a}_{-n})\tag{2.31}$$

In particular, when the step size  $\beta \rightarrow 0$ , the *linear reward-inaction* algorithm asymptotically converges to the NE of potential games with random reward values [122].

**Binary Log-Linear Algorithm (BLLA)** [123] is another low complexity policy-based RL algorithm, where similarly as before each decision-maker  $n \in \mathcal{N}$  approximate its policy  $\pi_n$  through a probability distribution. Assuming that at slot  $t$  the action profile is  $(a_1(t), \dots, a_{|\mathcal{N}|}(t))$ , a decision-maker  $n$  is chosen arbitrarily to perform exploration and learning. The decision-maker  $n$  chooses an action  $\hat{a}_n(t)$  uniformly, e.g., with probability  $\frac{1}{|\mathcal{A}_n|}$ , and observes its perceived new reward  $\hat{r}_n(t)$ , given that the rest decision-makers do not change actions. Therefore, the chosen decision-maker  $n$  updates its policy  $\pi$  as follows:

$$\begin{aligned}\pi_n[a_n(t+1) = \hat{a}_n(t)] &= \frac{e^{\hat{r}_n(t)\beta}}{e^{\hat{r}_n(t)\beta} + e^{r_n(t)\beta}} \\ \pi_n[a_n(t+1) = a_n(t)] &= \frac{e^{r_n(t)\beta}}{e^{\hat{r}_n(t)\beta} + e^{r_n(t)\beta}}\end{aligned}\tag{2.32}$$

where  $\beta$  is the step size. The described iterative procedure is repeated until some stop criterion is met, e.g., probability of every player for choosing a corresponding particular action is sufficiently approaching one. Furthermore, with  $\mathcal{A} = \mathcal{A}_1 \times \cdots \times \mathcal{A}_{|N|}$ , then the following theorem holds true [124]:

**Theorem 2** *If all decision-makers in a potential non-cooperative game  $\mathcal{G}$  adhere to the binary log-linear algorithm, then the unique stationary distribution  $\mu(\mathbf{a}) \in \Delta(\mathcal{A})$  of any action selection profile  $\mathbf{a} \in \mathcal{A}$ , is given as follows:*

$$\mu(a) = \frac{e^{\beta\Phi(\mathbf{a})}}{\sum_{\mathbf{a}' \in \mathcal{A}} e^{\beta\Phi(\mathbf{a}')}} \quad (2.33)$$

, where  $\Phi$  is the potential function of the game  $\mathcal{G}$ .

To this end, the binary log-linear algorithm with a sufficiently large  $\beta$ , asymptotically converges to an action profile that maximizes the potential function, thus to the best NE point of the game  $\mathcal{G}$  [125]. Some recent applications of the presented learning approaches can be found in [126, 127, 128, 129].

## 2.3 Prospect Theory

In this dissertation, we are interested in non-cooperative environments, where the decision-makers compete with each other for shared resources. To this end, the shared resources are considered as Common-pool resources (CPRs), which are characterized by the following properties [130]. In particular, they are non-excludable, thus they are accessible by any decision-maker, while at the same time they are rivalrous or subtractable, thus higher use by one decision-maker leads to less availability for the others. Rational or myopic behavior by decision-makers competing for a CPR often results in sub-optimal outcomes, or even to the potential destruction of the resource. The latter phenomenon is typically referred to in the research literature as the *Tragedy of the Commons* [131].

## Chapter 2. Preliminaries, and Introduced Theory

As a result, a CPR experiences probabilistic failure due to over-utilization, which leads to uncertainty in the outcomes for the decision-makers [132]. Generally, under this setting, where uncertainty is encountered, studies from behavioral economics show that humans are typically neither risk-neutral nor expected utility maximizers, i.e., rational decision-makers - Chapter 2.1.1, when making decisions under uncertainty, and instead exhibit risk-aware attitudes and characteristics [133]. One of the most widely accepted behavioral models of decision-making under probabilistic uncertainty is Prospect Theory, which was firstly introduced by Kahneman and Tversky [65], and describes a decision-making behavioral model under risk and uncertainty.

In particular, Prospect Theory captures humans' risk-aware behavioral characteristics, where a human acting as a decision-maker evaluates her perceived reward in a probabilistic manner, and concerning a reference point, i.e., reference reward, (*reference dependence property*). The latter constitutes a "safe", i.e., guaranteed, outcome that the decision-maker can perceive. The evaluation of decision-maker's experienced reward concerning a reference point formalizes the concepts of gains and losses, and these are expressed through decision-maker's prospect-theoretic utility function. Furthermore, the decision-maker is characterized as risk-seeking or risk-averse based on her perceived prospect-theoretic utility function, i.e., gains or losses. In particular, the decision-maker exhibits a greater dissatisfaction from a potential loss, i.e., less reward than the reference point, compared to its satisfaction from a gain, i.e., higher reward than the reference point, of the same magnitude (*loss aversion property*). In other words, the decision-maker's associated prospect-theoretic utility function is concave on gains and convex for losses, thus the decision-maker exhibits risk-seeking behavior under losses and risk-averse behavior under gains (*diminishing sensitivity property*). The latter gives rise to an *S-shaped* prospect-theoretic utility function. As a result, prospect-theoretic utility functions essentially account for systematic deviations in decision-maker's behavior from the predictions of the classical expected utility theory framework [64].

Prospect Theory has been applied in diverse settings involving decision-making under risk, including finance, insurance, industrial organization, and betting markets [134]. The human-robot interaction [135] could be considered as an indicative example, where humans' risk-awareness should be encountered by the robots for predicting humans' behavior and collaborate with them. Moreover, Prospect Theory has been utilized towards designing advertisement campaigns and products' promotions [136] through the modeling of customers' risk-aware behavioral characteristics and biases. Some research works have focused on examining decision-makers' behavior under uncertainty only for the cases of gains or losses in the examined system, i.e., concave or convex part of decision-maker's prospect-theoretic utility function, respectively, [66, 137]. Furthermore, Prospect Theory has been adopted in various cyber-physical environments and application domains, including dynamic resource management in 5G wireless networks [138, 139, 140, 141], public safety networks [142], anti-jamming communications in cognitive radio networks [143], users' transmission power management and anti-jamming techniques in UAV-assisted networks [144], Quality of Experience [145, 146] in cyber-physical social systems. In this dissertation, we enrich our introduced autonomous decision-making approaches with decision-maker's risk-aware behavior (i.e., concerning both gains and losses) under the principles of Prospect Theory, and by considering the risk of failure of the CPR in the considered application domain, as reflected by the theory of the *Tragedy of the Commons*.

### 2.3.1 Mathematical Background

Following the aforementioned properties of the behavioral model that is introduced by Prospect Theory, each decision-maker's  $n$  prospect-theoretic utility function for an uncertain outcome  $z_n \in \mathbb{R}$  is of the following form:

$$u_n(z_n) = \begin{cases} (z_n - z_n^\circ)^{\alpha_n} & , \text{ when } z_n \geq z_n^\circ \\ -k_n(z_n^\circ - z_n)^{\beta_n} & , \text{ otherwise} \end{cases} \quad (2.34)$$

, where  $z_n^\circ$  is the reference point concerning which losses and gains are defined, and  $\alpha_n, \beta_n$  are real-valued parameters lying in the interval  $(0, 1]$ . In particular, regarding the first branch of Equation 2.34, the maximization of  $u_n$  directly implies the maximization of decision-maker's gains, with respect to its reference point  $z_n^\circ$ , which constitutes the "safe" outcome, i.e., guaranteed perceived reward. On the other hand, based on the second branch of Equation 2.34, the maximization of  $u_n$  implies the minimization of decision-maker's losses, which are evaluated with respect to its reference point  $z_n^\circ$  as well.

The real parameters  $\alpha_n, \beta_n \in (0, 1]$  describe decision-maker's  $n$  sensitivity in gains and losses, respectively, and these are expressed through its perceived outcome  $z_n$ , and its reference point  $z_n^\circ$ . In other words, small values in  $\alpha_n$  describe decision-maker's risk-seeking behavior in losses, and risk-averse behavior in gains, while small values in  $\beta_n$  imply a higher decrease in decision-maker's prospect-theoretic utility function  $u_n$  when its perceived outcome  $z_n$  is close to its reference point  $z_n^\circ$ . Furthermore,  $k_n \in [0, \infty)$  reflects the impact of losses compared to gains in decision-maker's prospect-theoretic utility function  $u_n$ . In particular, for the case,  $k_n > 1$ , the decision-maker  $n$  weighs the losses more than the gains, whereas in the case  $0 < k_n \leq 1$  the decision-maker weighs the gains more or equal than the losses. As a result, different values in  $\alpha_n, \beta_n, k_n$  allows taking into account with high granularity all the different behavioral characteristics of each decision-maker, while in the case where  $\alpha_n = \alpha, \beta_n = \beta, k_n = k, \forall n \in \mathcal{N}$ , a homogeneous decision-makers population is considered with respect to their risk-aware characteristics.

### 2.3.2 Risk-aware Decision-Making Framework

In this chapter, we introduce a risk-aware decision-making framework, by bridging Game Theory, Reinforcement Learning, and Prospect Theory, thus incorporating decision-makers' risk-aware characteristics in their decision-making process. The latter is mainly motivated by the fact that in real-world applications, as the studied

application domains in this dissertation, humans demonstrate risk-aware attitudes during their decision-making process. The autonomous decision-making agents aim to provide personalized experience to their owners, thus the user's risk-aware characteristics should be considered properly. First of all, by adopting the notation that was

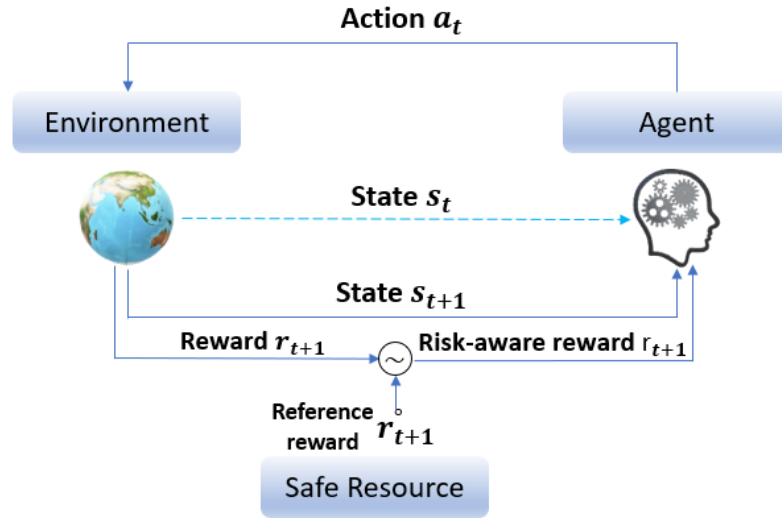


Figure 2.3: Risk-aware Reinforcement Learning

introduced in Chapter 2.2.1, a decision-maker's  $n \in \mathcal{N}$  received reward at slot  $t$ , and after performing action  $a_n \in \mathcal{A}_n$ , is defined as  $r_n(t)$ . As it was described in Chapter 2.2.3, under the concept of LGT, the decision-maker's reward  $r_n(t)$  can essentially be its perceived utility function value  $u_n$ , given the setting of a non-cooperative game  $\mathcal{G}$ . In the introduced RL techniques in Chapters 2.2.1, and 2.2.3, the decision-maker's  $n$  reward  $r_n(t)$  constitutes the environment's feedback that drives its actions, thus its learned policy  $\pi_n$ . As a result, the decision-maker's risk-aware characteristics could directly be incorporated in the introduced RL setting, through reward reshaping (Figure 2.3), thus via the decision-maker's risk-aware reward, which is defined as follows:

$$r_n(t) = \begin{cases} (r_n(t) - r_n^\circ(t))^{\alpha_n} & , \text{ when } r_n(t) \geq r_n^\circ(t) \\ -k_n(r_n^\circ(t) - r_n(t))^{\beta_n} & , \text{ otherwise} \end{cases} \quad (2.35)$$



## Chapter 2. Preliminaries, and Introduced Theory

, where  $r_n(t)$  is the decision-maker's  $n$  perceived reward at slot  $t$ , and is evaluated with respect to its reference reward point  $r_n^\circ(t)$ , which generally it is defined as a function of slot  $t$ , whereas it could hold true that  $r_n^\circ(t) = r_n^\circ, \forall t$ . Furthermore, by denoting as  $\mathbf{p}$  the CPR's probability of failure, and by assuming that in the case of CPR's not failure, the decision-maker's perceived reward  $r_n(t)$  is higher than its corresponding reference reward point  $r_n^\circ(t)$ , and lower in the case of CPR's failure, then Equation 2.35 could be rewritten as follows:

$$r_n(t) = \begin{cases} (r_n(t) - r_n^\circ(t))^{\alpha_n} & , \text{ with probability } (1 - \mathbf{p}) \\ -k_n(r_n^\circ(t) - r_n(t))^{\beta_n} & , \text{ with probability } \mathbf{p} \end{cases} \quad (2.36)$$

Following the properties of rationality (Chapter 2.1), the decision-makers could be modeled as risk-aware expected reward (utility) maximizers with respect to their perceived reward in Equation 2.36 as follows:

$$\mathbb{E}(r_n(t)) = (1 - \mathbf{p}) \cdot (r_n(t) - r_n^\circ(t))^{\alpha_n} + \mathbf{p} \cdot [-k_n(r_n^\circ(t) - r_n(t))^{\beta_n}] \quad (2.37)$$

, where  $\alpha_n, \beta_n, k_n$  are the decision-maker's risk-aware parameters, as these were described in Chapter 2.3.1. It is worth to be mentioned that, as in Chapters 2.2.1 & 2.2.3, the notions of decision-maker's perceived reward  $r_n$  and utility  $u_n$  are used equivalently under the setting of a non-cooperative game  $\mathcal{G}$ . As a result, Equations 2.35 & 2.37 could directly be used in the introduced RL techniques. In particular,  $r_n(t)$  corresponds to the environment's returned reward for the decision-maker  $n$ , after the latter performs its action  $a_n$ , whereas  $r_n^\circ(t)$  is the decision-maker's  $n$  reference reward point. As a result, the new reshaped reward  $r_n(t)$  could express the decision-maker's  $n$  final perceived reward after performing its action  $a_n$ , and in the case of a rational decision-maker  $\mathbb{E}(r_n(t))$  (Equation 2.37) could be used instead. Under this setting, decision-makers' risk-aware characteristics are incorporated in their decision-making process. Furthermore, the existence, uniqueness, and convergence to NE points of the non-cooperative games under the introduced risk-aware decision-making framework, are demonstrated in the studied application domains.

## Chapter 3

# Mobile Edge Computing

In this chapter, we present artificial intelligent and risk-aware autonomous decision-making approaches for the challenging and motivating application of Mobile Edge Computing (MEC). First, we provide an overview of Mobile Edge Computing to demonstrate its essential connection with Mobile Cloud Computing (MCC), and better underline the key benefits and features of the former compared to the latter one. Then, we focus our analysis on MEC, as it constitutes an application domain that has attracted tremendous research attention in the last 5 years, mainly since MEC can offer unique capabilities and features for mobile devices and users. However, MEC brings several key challenges that need to be tackled such that it is utilized effectively and efficiently. Towards this direction, we propose and analyze our decision-making approaches that meet several key and novel features and fulfill significant gaps in the current research literature. Our introduced approaches utilize the introduced theoretical concepts, as these were presented in Chapter 2. In particular, we demonstrate the applicability of our approaches, and we extend the basic theory with respect to the unique properties of the MEC application domain. Furthermore, we evaluate these approaches through extensive empirical work, which confirms our theory and demonstrates our approaches' perceived state-of-the-art performance.

The rise of 5G networks alongside the Internet of Things (IoT) evolution have

skyrocketed the number of connected objects, which was around a few dozen billion in 2015 and is foreseen to reach 500 billion devices that are connected to the Internet by 2030 [147], while the global mobile traffic is expected to increase sevenfold by 2021 [148]. These applications with intensive computation demands mainly ask for low latency and high energy consumption, which is difficult to be met by a mobile device, as it constitutes a resource-constrained computing system with limited computing capability and battery availability [149]. Despite the recent hardware advances in smart devices, several of them are not yet capable of efficiently supporting computationally-intensive applications, as their local computation and energy resources appear still insufficient. For that reason, MCC [150] was initially introduced as a computation offloading paradigm, where mobile devices offload their computation tasks to Cloud servers with rich computing and storage resources. Despite MCC's powerful computing and storage capability, connecting and communicating with remote servers in the cloud includes significant propagation delays in addition to large and unpredictable queuing delays [151], which make it difficult for MCC to fulfill the low latency requirements of the emerged applications of mobile devices.

Therefore, Mobile Edge Computing (MEC) [152, 153] was motivated as an alternative computation offloading paradigm that can promote the computation capability of mobile devices by providing computing services through MEC servers that are located at the edge of the core networks, thus nearby. Specifically, small cell base stations densely deployed and distributed close to mobile devices are empowered with powerful computing capabilities, and turn into MEC servers, thus edge computing nodes that can provide both communication and computation resources to mobile devices. The latter can offload their intensive computation tasks to the MEC servers for remote execution through wireless channels [154]. The data processing by MEC servers near the edge of the network reduces the traffic congestion at the backhaul of the network and makes propagation delays negligible. Furthermore, privacy and security are enhanced through MEC in contradiction with the central cloud that has

a single point of failure.

As a result, MEC provides the greatest help for latency-sensitive and computationally intensive applications, and several application domains such as connected vehicles [155], mobile AR/VR [156], Smart Cities [157, 158], etc, could be benefited and be developed effectively and efficiently by utilizing MEC. However, mobile devices' computation offloading, i.e., data offloading, in a MEC environment is not straightforward. The mobile devices' computation offloading includes computation probabilistic uncertainties due to the dynamic, and shared nature of the MEC environment, which need to be considered in the proposed resource allocation schemes, as they influence the users' experienced completion latency and energy consumption.

For that reason, some of the aforementioned key challenges in mobile devices computation offloading in the MEC environment have attracted great interest in the recent research literature [159]. Towards this direction, Software Defined Networking (SDN) [160] with its network programming capabilities stands out as a natural candidate technology complementing the MEC advancement for designing dynamic, intelligent, and adaptable networks that can address the key challenges in mobile devices computation offloading paradigm. SDN separates the control from the data plane and enables the programmable control mechanism [161]. To this end, MEC can substantially benefit from the SDN technology [162], as a centralized entity, i.e., SDN controller, has a global view of the MEC environment and manages mobile devices computation offloading more efficiently and dynamically.

Furthermore, Artificial Intelligence (AI) technologies have the potential to effectively address unstructured and seemingly intractable problems [163, 164] involving large and high dimensional amounts of data that need to be considered in the design and optimization of mobile devices computation offloading scheme. Reinforcement Learning (RL) [62], and its variant Deep Reinforcement Learning (DRL) [165, 166], as these were introduced in Chapter 2.2, constitute indicative examples of AI-enabled decision-making paradigms [167] that have attracted great interest in wireless net-

works [168, 169]. The latter stems mainly due to RL’s capability to effectively design and optimize decision-making policies by exploiting the interactions with the dynamic and unknown environment, as long as the recent advances in deep neural networks that can empower RL, i.e., DRL, and make it able to deal with high dimensional data, while at the same time enriched and intelligent generalization among the mobile devices is achieved. As a result, the interest of researchers in autonomous decision-making approaches for mobile devices associated with MEC servers, and mobile devices computation offloading is growing [170, 171, 172, 173].

### 3.1 Related Work

In this chapter, we review the current research literature for the main key challenges that arise in a MEC environment. In particular, we focus on the mobile devices computation offloading problem, which includes the required resource allocation, and mobile devices association with MEC servers, in the case where a multiple MEC servers environment is considered. As a result, we present research works that have well-investigated the aforementioned key challenges, by categorizing them into the following categories, thus based on the MEC environment setup, i.e., single/multiple MEC server(s), and key utilized technologies, i.e., RL/SDN-enabled MEC. In the first two categories, with sufficient assumptions and considerations, the research works focus on the data offloading problem through the consideration of well-established mathematical theories and algorithms, e.g., convex optimization, Lyapunov optimization, etc, while in the latter two categories, the authors mainly aim to introduce learning approaches that are able to deal with a more complex mathematical formalization of the aforementioned challenges, thus more realistically, e.g., mobility, environment’s dynamics consideration.

**Single MEC Server Environment:** Regarding the single MEC server setup, Mao

et al [174] have assumed that the computation task requests from the mobile users arrive in a stochastic manner, and they formulated a power consumption minimization problem with task buffer stability constraints to examine the tradeoff between the mobile users' power consumption and the execution delay of the computation tasks. The decision regarding the local execution and computation offloading is based on Lyapunov optimization, while the communication resources, i.e., transmission power and bandwidth, are allocated following the Gauss-Seidel method. A similar problem is considered in [175] under the consideration of a multi-channel wireless interference environment. The authors propose a distributed approach to determine the users' computation offloading decisions based on game theory.

On the other hand, a centralized approach is introduced in [176], [177], targeting the energy-efficient data offloading via jointly optimizing the computation offloading and the radio resource allocation for all the users in the network, to obtain the minimal energy consumption under the latency constraints in a single-MEC server environment. The same problem is studied in [178] under the assumption of mobile users' personalized delay requirements, which introduces additional constraints (as many as the number of users) in the corresponding optimization problem. This problem has been also extended in a MIMO multicell system [179], where multiple users offload their data to a single-MEC server. The formulated optimization problem is non-convex, thus the authors propose an iterative algorithm following the successive convex approximation technique to determine a local optimal data offloading and radio resource allocation. In [180] and [181], the authors assumed an orthogonal frequency division multiple access communication schemes and proposed a computation offloading approach that is minimizing the weighted sum of mobile users' energy consumption under appropriately considered constraints for the completion latencies of the mobile users' tasks.

In [182], the authors study the workload balancing problem in a fog network to minimize the latency of data flows in the communications and processing procedures

by associating mobile devices to suitable base stations. Hierarchical computing infrastructure is proposed in [183] consisting of shallow and deep cloudlets and the authors study the problem of users' data offloading to reduce the latency and improve the quality of service based on a queuing theory analysis. In [184], the authors examine the joint optimization problem of minimizing the system cost in terms of leasing virtual machines for computing purposes, while guaranteeing QoS requirements, and they address it as a mixed-integer nonlinear programming problem. In [185], the author provides a techno-economic analysis via proposing a coalitional game-based pricing scheme to study the users' data offloading problem.

The centralized partial data offloading problem, while the mobile users can harvest energy from the environment, is studied in [186] and [187] based on linear programming and Lyapunov optimization, respectively, towards determining the optimal policies of offloading decision, clock frequency control, power splitting ratio, and transmission power allocation. In [188], the authors focus their study on the communication collisions at the shared network when multiple users offload their data to a single-MEC server. The authors aim to minimize the average application completion time following a mixed integer programming approach. The problem of computation offloading under public safety events [189], where the MEC server can become unreachable for various reasons, e.g., natural disaster, terrorist attacks [190], is studied in [191]. The authors create mobile users' clusters, which can execute the assigned task by communicating in a device-to-device manner until the MEC server becomes reachable again. Furthermore, two single-MEC server architectures have been proposed in [192] and [193], named MVR architecture [192] which enables the use of virtual resources at the MEC server by the mobile users, and HMEC architecture [193], which not only enables the users to exploit the MEC server's computing capabilities and energy availability, but also provides means to protect the users' privacy.

**Multiple MEC Servers Environment:** Limited research work however has been performed so far in the multiple MEC servers environment regarding the full and/or partial offloading and radio resource allocation. In this setting, several additional dimensions arise in the decision-making process, namely: a) determine to which server(s) should a user offload its data, b) determine the total amount of data to be offloaded, and c) optimize the data offloading allocation among multiple MEC servers. All these aspects ideally should be treated jointly, as there is a strong interdependence among them. The latter makes the combined optimization and decision-making problem more complicated with the increasing number of users and MEC servers.

The problem of pure data offloading is studied in [194], where the authors aim to determine the amount of offloaded data to each MEC server (without considering the radio resource allocation) via formulating a multiple knapsack problem. Similarly, two separate problems are formulated in [195] regarding the mobile users' energy consumption minimization and the minimization of the application's execution latency. Both problems are non-convex ones and the authors transform the first problem to a convex one based on the variable substitution technique, while they propose a locally optimal algorithm with the univariate search technique to address the second one. The joint data offloading and radio resource allocation problem has been recently studied in [196] considering that the mobile users can harvest energy from the surrounding environment.

**SDN-enabled MEC:** The capabilities of SDN [197] have been exploited by the MEC environment to intelligently address the mobile users' computation offloading problem. In [198] the authors propose an approach for selecting a computing node, i.e., mobile user's device, MEC server, cloud server, through an SDN controller. The latter acts as a decision-maker that executes the Computing Mode Selection algorithm and its decisions are shared with the mobile users. Moreover, in [199],



SDN is used for designing an SDN enabled MEC framework for vehicular networks, where the authors suggest SDN as a candidate and promising solution for overcoming key challenges such as network scalability, and data traffic overload. The authors deploy an SDN technology with multiple SDN controllers, which can change their status into active or sleep based on the network status. Furthermore, the benefits of exploiting SDN in MEC within the era of Internet of Things (IoT) are discussed in details in the surveys [200, 162].

The authors in [201] focused their investigation on computation-intensive virtual reality and vehicular IoT applications, where an SDN enabled MEC framework was proposed for providing the necessary data plane flexibility, and reduced latency. On the other hand, in [202] SDN's capabilities are exploited for enhancing security in IoT, while in [203] SDN is combined with MEC to overcome the limitations of network densification of IoT within a smart home environment.

**RL-enabled MEC:** Recently, Reinforcement Learning (RL) has been utilized by the MEC environment as a powerful and data-driven decision-making paradigm that deals with the dynamic MEC environment. The majority of the recent research works focus on designing and optimizing mobile users' computation offloading policies in a MEC environment by exploiting RL's capabilities. Specifically, most of these research works introduce RL-based computation offloading policies, where each mobile user acts as decision-making agent and learns an optimal policy by using its observed interactions with the MEC environment. In [204] the authors propose a DRL-based approach to minimize the computation cost of each mobile user concerning energy consumption and buffering delay. The authors introduce a decentralized scheme, where each user learns an optimal continuous power allocation policy for both local execution and computation offloading. For that reason, each user maintains and trains a neural network on its interactions with the MEC environment and through a Deep Deterministic Policy Gradient (DDPG) approach [205] due to the continuous

nature (power allocation) of the problem.

On the other hand, in [206] the authors extend each mobile user's computation offloading policy since, except the power allocation, they also include the computation offloading decision, i.e., selection of a MEC server, and the CPU frequency allocation in the case of local execution. Similarly, the authors assume that each mobile user maintains its own Deep Q-network (DQN) [92], which is trained on mobile user's interactions with the MEC environment. A different RL-based approach is introduced in [105], where a centralized decision-making agent learns an optimal policy for mobile users' computation offloading decisions (offload or not), and the computational resource allocation at the MEC server's side. The policy is learned and optimized through RL and towards minimizing the overall aggregated cost of the system. The authors utilized a DQN network with a pre-classification step before learning to limit the large action space of the agent.

In [207] a semi-distributed computation offloading and scheduling algorithm based on DRL is introduced in an IoT MEC environment. The learned optimal policy minimizes the weighted sum of average delay and power consumption overall IoT devices. The authors consider a stochastic underlying arrival model that describes the dynamics of the MEC environment and utilize DRL for approximating the optimal value functions and determine an optimal policy. In a similar IoT MEC environment in [208], the authors propose a computation offloading problem formalization under the setting of partial offloading, thus each mobile user's application can be arbitrarily divided into two parts, where one part is computed locally, while the remaining part is offloaded to a MEC server. The latter constitutes the offloading strategy of each mobile user, and the authors proposed a DRL-based approach that exploits a DQN to optimize mobiles users' offloading strategies such that the overall system's cost is minimized. Furthermore, the mobile users' association to MEC servers is performed through a separate algorithm, which is based on channel gains, thus through a communication metric related to the wireless channel's conditions and mobile users'

mobility

## 3.2 Contributions

The key contributions of the following presented research work towards addressing the aforementioned challenges, and filling several research gaps in Mobile Edge Computing (MEC), are summarized as follows:

1. The problem of monetary-based pricing of the MEC servers computing services has not been addressed in the existing literature. To this end, an SDN-powered MEC architecture towards dealing with the joint problem of intelligent MEC server selection and mobile users' data offloading in multiple MEC servers and multiple mobile users, is proposed. MEC servers' key characteristics, i.e., pricing, discount, cost, and users' offloaded data are considered for introducing welfare functions for the MEC servers. The mobile users' association to the MEC servers is handled by an RL framework that utilizes each MEC server's reputation score, and past experiences for optimizing mobile users' perceived cumulative reward. A two-layers optimization problem is formulated and solved by the SDN controller aiming at maximizing the MEC servers' profit and also maximizing the perceived satisfaction by the mobile users. The mobile users' maximization problem of their satisfaction is addressed at the first stage as a non-cooperative game among the end-users, who practically aim at maximizing their utility function. A Nash Equilibrium (NE) point is determined, which expresses the optimal amount of offloading data for each mobile user. At the second stage of the joint optimization problem, given the mobile users' offloaded data, an optimization problem of each MEC server's profit (as expressed by its welfare function) is formulated and solved by the MEC servers. A series of detailed simulation experiments and a comparative evaluation are performed to evaluate the performance of the proposed decision-making framework.
2. We introduce a surveillance system consisting of areas of interest (AoI) with IP

### *Chapter 3. Mobile Edge Computing*

cameras, MEC servers, and a Fully Autonomous Aerial System (FAAS). The cameras partially offload the computing tasks related to the videos' processing to the MEC server that is associated with the AoI or to the FAAS, if the latter is flying above the AoI, while the rest are executed locally at the IP cameras. Each IP camera experiences a time and energy overhead to offload its data and process a part of the data locally. A holistic utility function is introduced representing the IP cameras' level of achieved Quality-of-Service (QoS) while accounting for their time and energy constraints that they possess in the video processing procedure. A non-cooperative game among the IP cameras is formulated and the concept of Satisfaction Equilibrium (SE) (Chapter 2.1) is adopted to determine a stable data offloading, where the IP cameras satisfy their minimum QoS prerequisites. A distributed decision-making framework based on RL determines the IP cameras' data offloading at the SE, if the latter exists. If the SE does not exist, the proposed Distributed Learning Satisfaction Equilibrium (DLSE) algorithm converges to the Generalized SE, where only a part of the cameras satisfy their QoS prerequisites. A Reinforcement Learning (RL) algorithm indicates the FAAS's movement by considering the quality of information from the AoIs. Detailed numerical results are presented to evaluate the proposed framework's pure operation, and its scalability performance, while a comparative evaluation is provided to show its drawbacks and benefits.

3. We propose a risk-aware decision-making framework that incorporates human awareness for IoT devices' computation offloading to a MEC server, by leveraging the principles of Game Theory and Prospect Theory. The consideration of the proposed decision-making framework is well motivated and supported by the observations that the MEC IoT environment presents a competitive resource-constrained environment, where the users are making decisions under uncertainty, which stems from the partially available information and the competition to share the limited resources. The individual entities of a system, i.e., cognitive IoT devices, make distributed and autonomous decisions under risk and uncertainty of the associated payoff of their de-

### *Chapter 3. Mobile Edge Computing*

cisions, which is determined in a probabilistic manner, while they may demonstrate systematic deviations from the expected utility theory, where all the individuals are assumed as risk-neutral concerning their choices. To this end, a properly defined prospect-theoretic utility function is utilized, which encapsulates the device's perceived energy and time overhead. The devices' data offloading is formulated as a non-cooperative game, and is solved in a distributed manner by converging to a unique Nash Equilibrium point (Chapter 2.1, whose existence and uniqueness are theoretically proven, and empirically shown. A series of experiments are performed to evaluate the performance and the inherent attributes of the proposed risk-aware decision-making framework.

4. Towards fulfilling the limited research work that has been performed so far in multiple MEC server environments, We extend our risk-aware decision-making framework towards considering and addressing mobile users' computation offloading under energy and time overhead constraints in multiple MEC servers, and multiple mobile users environment. We address the data offloading problem under realistic uncertainties of the multiple MEC servers that have certain and limited capabilities. To this end a heterogeneous environment is introduced, where each user can offload arbitrarily parts of its application to multiple MEC servers for remote execution. The users determine, under risk, the computation load to be offloaded at each MEC server, taking into consideration the computation uncertainty (limited computation capability) at each MEC server, due to its shared nature among the users. The problem of each user determining autonomously the portion of its computation task that will be performed at each MEC server has been formulated as a constrained convex optimization problem of each user's satisfaction utility and is treated as a non-cooperative game among the users. The respective non-cooperative game is solved in a distributed manner, and the existence and uniqueness of a Pure Nash Equilibrium (PNE) are proven. A distributed decision-making framework that converges to the PNE is introduced. Our indicative results demonstrate the performance of the risk-

aware decision-making framework under the considered complex environment, and a comparative evaluation highlights its benefits and state-of-the-art performance.

A detailed description of the introduced decision-making frameworks, alongside their theoretical principles and empirical evaluations, is demonstrated in the following chapters.

### **3.3 Intelligent Data Offloading in Competitive Mobile Edge Computing Market**

Software-Defined Networks (SDN) and Mobile Edge Computing (MEC), capable of dynamically managing and satisfying the end-users computing demands, have emerged as key enabling technologies of 5G networks. In this research work, the joint problem of MEC server selection by the end-users and their optimal data offloading, as well as the optimal price setting by the MEC servers is studied in multiple MEC servers and multiple end-users environments. The flexibility and programmability offered by the SDN technology enable the realistic implementation of the proposed framework. Initially, an SDN controller executes a Reinforcement Learning framework based on the theory of Stochastic Learning Automata (Chapter 2.2.3) towards enabling the end-users to select a MEC server to offload their data.

The discount offered by the MEC server, its congestion, and its penetration in terms of serving end-users computing tasks and its announced pricing for its computing services is considered in the overall MEC selection process. To determine the end-users' data offloading portion to the selected MEC server, a non-cooperative game among the end-users of each server is formulated and the existence and uniqueness of the corresponding Nash Equilibrium are shown. An optimization problem of maximizing the MEC servers' profit is formulated and solved in order to the MEC servers' optimal pricing with respect to their offered computing services and the received offloaded data. To realize the proposed framework, an iterative and low-

complexity algorithm is introduced and designed. The performance of the proposed approach is evaluated through modeling and simulation under several scenarios, with both homogeneous and heterogeneous end-users.

### 3.3.1 System Model

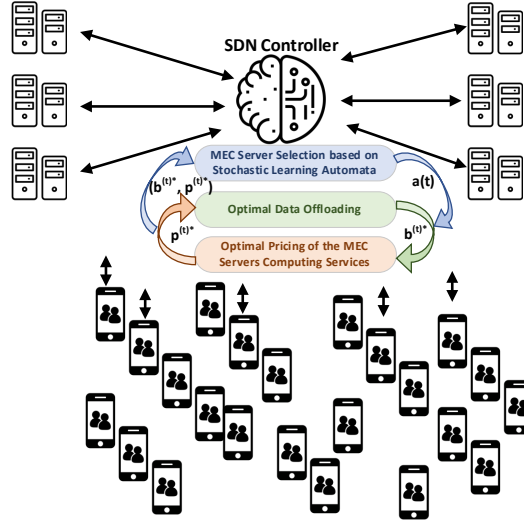


Figure 3.1: SDN-enabled MEC architecture

The SDN-enabled MEC architecture consisting of multiple MEC servers and multiple end-users is presented in Figure 3.1. Each MEC server  $s \in S$ ,  $S = [1, \dots, s, \dots, |S|]$ , communicates with the SDN controller towards setting the price  $p_s^{(t)}$  [\$/bits] of its computing services per time slot  $t$ . The whole operation of the examined system is divided in time slots, where  $T = [1, \dots, t, \dots, |T|]$  denotes their corresponding set. At each time slot, the SDN controller determines the MEC server selection by the end-users (see Section 3.3.2), as well as the optimal price  $p_s^{(t)}$  for each MEC server and the optimal data offloading  $b_{u,s}^{(t)}$  [bits] of each end-user  $u$  to the selected server  $s$  (see Section 3.3.3). Each end-user  $u$ ,  $u \in U$ ,  $U = [1, \dots, u, \dots, |U|]$  receives the required information by the SDN controller to offload its data  $b_{u,s}^{(t)}$  to the selected server  $s$ . Each end-user  $u$  has a maximum amount of data  $I_u^{(t)}$  that should be processed to

perform a computing task, and part of them are offloaded to the MEC server, i.e.,  $b_{u,s}^{(t)} \in A_u^{(t)} = [0, I_u^{(t)}]$ , while the rest of the data are processed locally.

### End-User Utility Function

At the beginning of each time slot, each end-user  $u$  sends to the SDN controller its total computing demands  $I_u^{(t)}$  that is needed to execute a computing task, while the SDN controller determines the optimal amount of offloaded data  $b_{u,s}^{(t)}$  for end-user  $u$  at the MEC server  $s$ , as it will be explained in detail in Section 3.3.3. Given that the MEC servers have bounded and limited computing capabilities, the data offloading strategies of the rest of the end-users, i.e.,  $\mathbf{b}_{-\mathbf{u}}^{(t)}$ , contribute to the configuration of the prices announced by the MEC servers and influence the data offloading  $b_{u,s}^{(t)}$  of end-user  $u$ . Thus, towards formulating the user's  $u$  perceived satisfaction, the end-user's relative data offloading is defined as  $r_u^{(t)} = \frac{b_{u,s}^{(t)}}{B_{-u}^{(t)}}$ , where  $B_{-u}^{(t)} = \sum_{s \in S} \sum_{u' \in U, u' \neq u} b_{u',s}^{(t)}$  expresses the total data offloading of the rest of the end-users  $u', u' \in U - \{u\}$ . The end-user's actual perceived satisfaction  $s_u^{(t)}$  at time slot  $t$  is increasing with respect to its relative data offloading  $b_{u,s}^{(t)}$ , as part of the requested computing task is offloaded to the MEC server and does not consume the end-user's local computing resources. Also, after the end-user offloads its total data  $I_u^{(t)}$  to the MEC server, its perceived satisfaction is saturated as the end-user cannot benefit more from the MEC server's computing services. Thus, without loss of generality and for presentation purposes only, here we adopt a logarithmic function with respect to the end-user's offloaded data  $b_{u,s}^{(t)}$  to capture end-user's actual perceived satisfaction, as follows.

$$s_u^{(t)}(b_{u,s}^{(t)}, \mathbf{b}_{-\mathbf{u}}^{(t)}) = \alpha_u \log(1 + \beta_u r_u^{(t)}) \quad (3.1)$$

where  $\mathbf{b}_{-\mathbf{u}}^{(t)}$  is the vector of all end-users data offloading excluding end-user  $u$ , and the  $\alpha_u, \beta_u \in \mathbb{R}^+$  parameters determine the slope of the logarithmic function in a personalized manner for end-user  $u$ , thus, expressing how easily or not an end-user  $u$  becomes satisfied by offloading its data to the MEC server.



### Chapter 3. Mobile Edge Computing

Additionally, each end-user is charged for using the MEC server's computing services in a fair manner accordingly to its relative data offloading. This policy enables even the low-budget end-users to exploit the MEC servers' capabilities at some degree, by prohibiting the high-budget ones to dominate the system. Thus, the cost function of end-user's  $u$  offloaded data is formulated as follows.

$$c_u^{(t)}(b_{u,s}^{(t)}, \mathbf{b}_{-\mathbf{u}}^{(\mathbf{t})}) = d_u^{(t)} p_s^{(t)} r_u^{(t)} \quad (3.2)$$

where  $d_u^{(t)}, d_u^{(\mathbf{t})} \in \mathbb{R}^+$  expresses end-user's  $u$  spending dynamics in order to use the MEC server's computing services. Specifically, a smaller value of  $d_u^{(t)}$  reflects end-user's  $u$  dynamic behavior to spend more money in order to buy computing support from the MEC servers. The price announced by the MEC server  $s$  is denoted as  $p_s^{(t)} [$/bits]$ .

Following the above analysis, end-user's  $u$  utility function captures both its actual perceived satisfaction  $s_u^{(t)}$  and its corresponding cost  $c_u^{(t)}$  in order to enjoy the MEC server's computing services. The end-user's  $u$  utility function is defined as follows.

$$U_u^{(t)}(b_{u,s}^{(t)}, \mathbf{b}_{-\mathbf{u}}^{(\mathbf{t})}, \mathbf{p}^{(\mathbf{t})}) = s_u^{(t)}(b_{u,s}^{(t)}, \mathbf{b}_{-\mathbf{u}}^{(\mathbf{t})}) - c_u^{(t)}(b_{u,s}^{(t)}, \mathbf{b}_{-\mathbf{u}}^{(\mathbf{t})}) = \alpha_u \log(1 + \beta_u r_u^{(t)}) - d_u^{(t)} p_s^{(t)} r_u^{(t)} \quad (3.3)$$

where  $\mathbf{p}^{(\mathbf{t})} = [p_1^{(t)}, \dots, p_s^{(t)}, \dots, p_{|S|}^{(t)}]$  denotes the vector of the prices announced by all the MEC servers.

#### Mobile Edge Computing Server Characteristics & Profit

Each MEC server  $s, s \in S$  supports a total computing demand of the end-users per time slot equal to  $\sum_{u \in U} b_{u,s}^{(t)}$  from all the end-users that selected the specific MEC server to offload their data. Also, towards providing incentives to the end-users to select a specific MEC server to be served from, the latter provides some discounts  $f_s^{(t)}$  expressed as a percentage of the originally announced price of its computing services. Furthermore, the MEC server has an actual cost  $c_s^{(t)} [$/bits]$  to process the amount

### Chapter 3. Mobile Edge Computing

of data that it receives. Please be also reminded that the MEC server charges the end-users with a price  $p_s^{(t)}$  [\$/bits] for the computing services that it offers.

Additionally, a MEC server increases its positive reputation towards the end-users if it is characterized by a good penetration within the end-users' computing demands. Specifically, the penetration of a MEC server  $s$  is defined as the total amount of data that the server  $s$  processed over the total amount of data that are processed within the SDN-powered MEC system for a total time period  $T$ , i.e.,  $\frac{\sum_{t \in \{1, \dots, T\}} \sum_{u \in U} b_{u,s}^{(t)}}{\sum_{s \in S} \sum_{t \in \{1, \dots, T\}} \sum_{u \in U} b_{u,s}^{(t)}}$ . Also, we assume that each MEC server  $s$  can handle a total amount of data  $B_s^{Max}$ . Thus, an indicative parameter showing the congestion of the MEC server per time slot in terms of processing the end-users' offloaded data is expressed as the ratio of the total amount of data  $\sum_{u \in U} b_{u,s}^{(t)}$  that the MEC server processes in time slot  $t$  over its total computing capability of data  $B_s^{Max}$ , i.e.,  $CONG_s = \frac{\sum_{u \in U} b_{u,s}^{(t)}}{B_s^{Max}}$ . Following the above analysis and combining all the aforementioned factors and parameters that characterize the MEC server  $s$ , its reputation score within the SDN-powered MEC environment is defined as follows.

$$R_s^{(t)} = w_1 \frac{\frac{\sum_{k \neq s} [(1-f_k^{(t)})] p_k^{(t)}}{K}}{(1-f_s^{(t)}) p_s^{(t)}} + w_2 \frac{1}{(1+CONG_s)^3} + w_3 \frac{\sum_{t \in \{1, \dots, T\}} \sum_{u \in U} b_{u,s}^{(t)}}{\sum_{s \in S} \sum_{t \in \{1, \dots, T\}} \sum_{u \in U} b_{u,s}^{(t)}} \quad (3.4)$$

In Equation 3.4, the first term expresses the relative pricing of a MEC server  $s$  in terms of offering its computing services to the end-users, the second term expresses the level of MEC server's congestion towards serving the end-users, while the third term expresses its penetration in serving end-users' computing demands. The weights  $w_1, w_2, w_3$  are configurable parameters that express the relative weight of each term within our study, and it should hold true that  $w_1 + w_2 + w_3 = 1$ . The revenue of each MEC server  $s$  from processing a total amount of end-users' offloaded data  $\sum_{u \in U} b_{u,s}^{(t)}$  depends on the announced price  $p_s^{(t)}$  and the corresponding discount  $f_s^{(t)}$  that the MEC server provides, and is given as follows.

$$REV_s^{(t)}(\mathbf{b}^{(t)}, \mathbf{p}^{(t)}) = (1-f_s^{(t)}) p_s^{(t)} \sum_{u \in U} b_{u,s}^{(t)} \quad (3.5)$$

where  $\mathbf{b}^{(t)}$  is the data offloading vector of all the end-users and  $\mathbf{p}^{(t)}$  denotes the announced prices by all the MEC servers in the system. On the other hand, the MEC server's total monetary cost to perform the processing of the offloaded data, is given as follows.

$$C_s^{(t)}(\mathbf{b}^{(t)}) = c_s^{(t)} \sum_{u \in U} b_{u,s}^{(t)} \quad (3.6)$$

where  $c_s^{(t)}$  is the MEC server's  $s$  computing cost per unit of data. Thus, the MEC server's profit is concluded by subtracting its cost from its revenue and is given as follows.

$$P_s^{(t)}(\mathbf{b}^{(t)}, \mathbf{p}^{(t)}) = REV_s^{(t)}(\mathbf{b}^{(t)}, \mathbf{p}^{(t)}) - C_s^{(t)}(\mathbf{b}^{(t)}) = (1 - f_s^{(t)})p_s^{(t)} \sum_{u \in U} b_{u,s}^{(t)} - c_s^{(t)} \sum_{u \in U} b_{u,s}^{(t)} \quad (3.7)$$

### 3.3.2 MEC as a Learning System

At the SDN controller's side, each end-user is represented and considered as a Stochastic Learning Automaton that senses the environment and it makes future decisions based on its past experience. At each time slot  $t$ , the end-user can select to be served by a MEC server  $s, s \in S$ , thus, the set of end-users' actions at time slot  $t$  is  $a(t) = \{a_1, \dots, a_s, \dots, a_S\}$ . The SDN controller has the information of the end-users' offloaded data  $\mathbf{b}^{(t)}$  and the prices  $\mathbf{p}^{(t)}$  that the MEC servers announce regarding offering their computing services. The SDN controller can determine the reputation score  $R_s^{(t)}$  for each MEC server, which can be normalized towards defining the reward probability as follows.

$$rew_s^{(t)} = \frac{R_s^{(t)}}{\sum_{s \in S} R_s^{(t)}} \quad (3.8)$$

The reward probability  $rew_s^{(t)}, 0 \leq rew_s^{(t)} \leq 1$  represents the potential reward that an end-user may experience by choosing to offload its data to the MEC server  $s$ . Following the theory of the Stochastic Learning Automata (Chapter 2.2.3), the action

### Chapter 3. Mobile Edge Computing

probability vector of an end-user  $u, u \in U$  is  $\mathbf{Pr}_u^{(t)} = [Pr_{u,1}^{(t)}, \dots, Pr_{u,s}^{(t)}, \dots, Pr_{u,S}^{(t)}]$ , where  $Pr_{u,s}^{(t)}$  is defined as the probability of the end-user  $u$  to select the MEC server  $s$  to offload its data. Based on the theory of Stochastic Learning Automata (Chapter 2.2.3), the rule of updating the end-users' action probabilities at the SDN controller is defined as follows.

$$Pr_{u,s}^{(t+1)} = Pr_{u,s}^{(t)} - b \cdot rew_s^{(t)} \cdot Pr_{u,s}^{(t)}, \quad s^{(t+1)} \neq s^{(t)} \quad (3.9a)$$

$$Pr_{u,s}^{(t+1)} = Pr_{u,s}^{(t)} + b \cdot rew_s^{(t)} \cdot (1 - Pr_{u,s}^{(t)}), \quad s^{(t+1)} = s^{(t)} \quad (3.9b)$$

where  $0 < b < 1$  denotes the learning parameter expressing how fast the end-users explore the available options of the MEC servers towards offloading their data. Equation 3.9a represents the probability of end-user  $u$  selecting a different MEC server to offload its data in the next time slot  $t + 1$  compared to end-user's choice in the current time slot  $t$ , while Equation 3.9b expresses the probability of end-user  $u$  to keep being served by the same MEC server. It is noted that initially, the end-users' action probabilities are initialized as  $Pr_{u,s}^{(t=0)} = \frac{1}{S}$ . The MEC server selection learning process executed at the SDN controller is presented in the Data Offloading and MEC Server Selection (DO-MECS) algorithm (see Section 3.3.4).

#### 3.3.3 Autonomous Data Offloading & Price Setting

Following the above described Reinforcement Learning technique of the Stochastic Learning Automata, each end-user has concluded the selection of a MEC server to offload its data. Then, the goal of each MEC server is to maximize its profit by processing the end-users' data, while the goal of each end-user is to maximize its perceived satisfaction, as expressed by its utility function, by offloading the optimal amount of data to the selected MEC server. Thus, a two-layer optimization problem

is formulated, as follows.

$$\mathbf{b}^{(t)*} = \arg \max_{b_{u,s}^{(t)}} U_u^{(t)}(b_{u,s}^{(t)}, \mathbf{b}_{-\mathbf{u}}^{(t)}, \mathbf{p}^{(t)}) \quad (3.10a)$$

$$\mathbf{p}^{(t)*} = \arg \max_{\mathbf{p}^{(t)}} P_s^{(t)}(\mathbf{b}^{(t)}, \mathbf{p}^{(t)}) \quad (3.10b)$$

As it is observed by Equation 3.10a and Equation 3.10b, the MEC servers optimal price  $\mathbf{p}^{(t)*}$  and the end-users optimal data offloading  $\mathbf{b}^{(t)*}$  are interdependent, thus, the joint optimization problem is formulated as a two-layer optimization framework. Initially, the end-users determine their optimal data offloading  $\mathbf{b}^{(t)*}$  via confronting the optimization problem of their utility functions as a non-cooperative game among them. Then, at the second layer, the MEC servers determine their optimal announced prices  $\mathbf{p}^{(t)*}$  given the data offloading of the end-users, via solving an optimization problem. The formulation and solution of the optimization problem are performed at the SDN controller, where its advanced computing capabilities enable fast decision-making. In the following two subsections, we will analyze in detail each layer of the optimization problem.

### Optimal Data Offloading

At first the optimal data offloading  $b_{u,s}^{(t)*}$  of each end-user  $u$  that has selected to offload its data to the MEC server  $s$  at the time slot  $t$  is determined. A non-cooperative game  $G = [U, \{A_u^{(t)}\}, \{U_u^{(t)}\}]$  is formulated among the end-users who compete with each other towards determining their optimal data offloading. The game  $G$  consists of three components: (a) the set of end-users (i.e., players)  $U = [1, \dots, u, \dots, |U|]$ , (b) the strategy space  $A_u^{(t)} = [0, I_u^{(t)}]$ , where  $b_{u,s}^{(t)} \in A_u^{(t)}$ , and (c) the end-user's utility function  $U_u^{(t)}$ . Each end-user wants to maximize its personal utility function, while considering the physical limitations, as follows.

$$\max_{b_{u,s}^{(t)}} U_u^{(t)}(b_{u,s}^{(t)}, \mathbf{b}_{-\mathbf{u}}^{(t)}, \mathbf{p}^{(t)}) \quad (3.11a)$$

$$s.t. \quad 0 \leq b_{u,s}^{(t)} \leq I_u^{(t)} \quad (3.11b)$$

### Chapter 3. Mobile Edge Computing

In the following analysis, our goal is to show the existence and uniqueness of a Nash Equilibrium (Chapter 2.1) for the data offloading game. The necessary and sufficient conditions are: (i) the strategy space  $A_u^{(t)}, \forall u \in U$  should be non-empty, convex and compact subset of an Euclidean space  $\mathbb{R}^U$  and (ii) the utility function  $U_u^{(t)}(b_{u,s}^{(t)}, \mathbf{b}_{-\mathbf{u}}^{(t)}, \mathbf{p}^{(t)})$  is continuous in  $\mathbf{b}_{\mathbf{u}}^{(t)}$  and quasi-concave in  $b_{u,s}^{(t)}$ .

**Theorem 3** *The Nash Equilibrium point of the game  $G = [U, \{A_u^{(t)}\}, \{U_u^{(t)}\}]$  exists and the end-user's best response data offloading strategy is given as follows.*

$$BR_u(\mathbf{b}_{-\mathbf{u}}^{(t)*}) = b_{u,s}^{(t)*} = \frac{B_{-u}^{(t)}}{\beta_u} \left[ \frac{\alpha_u \beta_u}{d_u^{(t)} p_s^{(t)}} - 1 \right] \quad (3.12)$$

where  $0 \leq b_{u,s}^{(t)*} \leq I_u^{(t)}$ .

*Proof:* The strategy space  $A_u^{(t)} = [0, I_u^{(t)}]$  represents the amount of data that the end-user  $u$  can offload to a MEC server  $s$ , thus by definition it is non-empty, convex, and compact subset of the Euclidean space  $\mathbb{R}^U$ . Also, based on Equation 3.3, the utility function  $U_u^{(t)}(b_{u,s}^{(t)}, \mathbf{b}_{-\mathbf{u}}^{(t)}, \mathbf{p}^{(t)})$  is continuous in  $\mathbf{b}_{\mathbf{u}}^{(t)}$ . Furthermore, we determine the second order derivative of the utility function  $U_u^{(t)}(b_{u,s}^{(t)}, \mathbf{b}_{-\mathbf{u}}^{(t)})$  with respect to  $b_{u,s}^{(t)}$ , as follows:

$$\frac{\partial^2 U_u^{(t)}(b_{u,s}^{(t)})}{\partial b_{u,s}^{(t)2}} = -\frac{\alpha_u \beta_u^2}{B_{-u}^{(t)2}} \cdot \frac{1}{[\beta_u + \frac{\beta_u b_{u,s}^{(t)}}{B_{-u}^{(t)}}]^2} < 0 \quad (3.13)$$

Given that  $\frac{\partial^2 U_u^{(t)}(b_{u,s}^{(t)})}{\partial b_{u,s}^{(t)2}} < 0$ , the  $U_u^{(t)}(b_{u,s}^{(t)}, \mathbf{b}_{-\mathbf{u}}^{(t)}, \mathbf{p}^{(t)})$  is concave in  $b_{u,s}^{(t)}$ , thus, it is also quasi-concave in  $b_{u,s}^{(t)}$ . Therefore, the Nash Equilibrium point of the game  $G = [U, \{A_u^{(t)}\}, \{U_u^{(t)}\}]$  exists. Towards determining the best response strategy of each end-user, we calculate the critical points of the  $U_u^{(t)}(b_{u,s}^{(t)}, \mathbf{b}_{-\mathbf{u}}^{(t)}, \mathbf{p}^{(t)})$ , as follows.

$$\frac{\partial U_u^{(t)}}{\partial b_{u,s}^{(t)}} = 0 \Leftrightarrow b_{u,s}^{(t)} = \frac{B_{-u}^{(t)}}{\beta_u} \left( \frac{\alpha_u \beta_u}{d_u^{(t)} p_s^{(t)}} - 1 \right) \quad (3.14)$$

The data offloading of each end-user  $u$  should satisfy the physical limitations, i.e.,  $0 \leq b_{u,s}^{(t)} \leq I_u^{(t)}$ , thus we have the following cases.

### Chapter 3. Mobile Edge Computing

CASE A: If  $d_u^{(t)} p_s^{(t)} > \alpha_u \beta_u$  then the best response strategy is  $b_{u,s}^{(t)*} < 0$ . But since the physical limitation imposed states that  $0 \leq b_{u,s}^{(t)}$  and our function is concave, then the best response should be  $b_{u,s}^{(t)*} = 0$ .

CASE B: If  $d_u^{(t)} p_s^{(t)} < \frac{\alpha_u \beta_u}{I_u^{(t)} \frac{\beta_u}{B_{-u}^{(t)}} + 1}$  then the best response strategy is  $b_{u,s}^{(t)*} > I_u^{(t)}$ . But since the physical limitation imposed states that  $b_{u,s}^{(t)} \leq I_u^{(t)}$  and our function is concave, then the best response should be  $b_{u,s}^{(t)*} = I_u^{(t)}$ .

CASE C: If  $\frac{\alpha_u \beta_u}{I_u^{(t)} \frac{\beta_u}{B_{-u}^{(t)}} + 1} \leq d_u^{(t)} p_s^{(t)} \leq \alpha_u \beta_u$  then the best response strategy is  $0 \leq b_{u,s}^{(t)*} \leq I_u^{(t)}$  which satisfies the physical limitation. In this case, the best response is given by the equation  $b_{u,s}^{(t)*} = \frac{B_{-u}^{(t)}}{\beta_u} \left( \frac{\alpha_u \beta_u}{d_u^{(t)} p_s^{(t)}} - 1 \right)$ .

Theorem 3 proves the existence of the Nash Equilibrium point of the game  $G$  and determines the best response strategy for each end-user  $u, u \in U$ . In the following theorem, the uniqueness of the Nash Equilibrium point of the game  $G$  is examined.

**Theorem 4** *The Nash Equilibrium point  $b_{u,s}^{(t)*}, \forall u \in U, s \in S$  of the game  $G$  is unique.*

*Proof:* The uniqueness of the Nash Equilibrium point  $b_{u,s}^{(t)*} = BR_u(\mathbf{b}_{-u}^{(t)*})$ , for cases A and B, the Nash Equilibrium point is trivially unique, while for case C we should show that the best response strategy  $BR_u(\mathbf{b}_{-u}^{(t)*})$  is a standard function [209]. The properties of a standard function are the following:

- Positivity  $\mathbf{f}(\mathbf{x}) > \mathbf{0}$ ;
- Monotonicity: if  $\mathbf{x} \geq \mathbf{x}'$ , then  $\mathbf{f}(\mathbf{x}) \geq \mathbf{f}(\mathbf{x}')$ ;
- Scalability: for all  $a > 1$ ,  $a \cdot \mathbf{f}(\mathbf{x}) \geq \mathbf{f}(a \cdot \mathbf{x})$ .

If a fixed point exists in a standard function, then it is unique [209]. Using Equation 3.12, the above properties of the standard function can be easily shown for the end-user's best response function  $BR_u(\mathbf{b}_{-u}^{(t)*})$ . Thus, the Nash Equilibrium point

of the game  $G$  is unique. In conclusion, it is noted that the optimal data offloading of each end-user is given by Equation 3.12.

### Optimal Pricing of the MEC Servers Computing Services

Here, the optimal pricing of the MEC server's computing services is determined towards maximizing the MEC servers' profit given the offloaded data of the end-users. Combining Equation 3.7, Equation 3.10b and Equation 3.12, the corresponding optimal pricing problem of the MEC servers can be written as follows.

$$\begin{aligned} \mathbf{p}^{(t)*} &= \arg \max_{\mathbf{p}^{(t)}} P_s^{(t)}(\mathbf{b}^{(t)}, \mathbf{p}^{(t)}) \\ &= (1 - f_s^{(t)}) p_s^{(t)} \sum_{u \in U} \left[ \frac{B_{-u}^{(t)}}{\beta_u} \left[ \frac{\alpha_u \beta_u}{d_u^{(t)} p_s^{(t)}} - 1 \right] \right] - c_s^{(t)} \sum_{u \in U} \left[ \frac{B_{-u}^{(t)}}{\beta_u} \left[ \frac{\alpha_u \beta_u}{d_u^{(t)} p_s^{(t)}} - 1 \right] \right] \end{aligned} \quad (3.15)$$

Based on Equation 3.15, it is observed that the optimal pricing problem of the MEC servers' computing services is a function only of their prices  $p_s^{(t)}$ ,  $s \in S$ .

**Theorem 5** *The optimal pricing announced by each MEC server for its computing services given the end-users offloaded data and towards maximizing its own profit is given as follows.*

$$p_s^{(t)*} = \left[ \frac{\alpha_u \beta_u c_s^{(t)} \sum_{u \in U} \frac{B_{-u}^{(t)}}{d_u^{(t)}}}{(1 - f_s^{(t)}) \sum_{u \in U} B_{-u}^{(t)}} \right]^{1/2} \quad (3.16)$$

*Proof:* Towards determining the optimal pricing announced by each MEC server, we take the first order derivative with respect to  $p_s^{(t)}$  and determine the critical points.

$$\frac{\partial P_s^{(t)}(\mathbf{b}^{(t)}, \mathbf{p}^{(t)})}{\partial p_s^{(t)}} = -\frac{1}{\beta_u} (1 - f_s^{(t)}) \sum_{u \in U} B_{-u}^{(t)} + \frac{c_s^{(t)} \alpha_u}{p_s^{(t)2}} \sum_{u \in U} \frac{B_{-u}^{(t)}}{d_u^{(t)}} = 0 \quad (3.17)$$

Thus, the critical points are given as per Equation 3.16. Furthermore, by checking the second order derivative of  $P_s^{(t)}(\mathbf{b}^{(t)}, \mathbf{p}^{(t)})$  with respect to  $p_s^{(t)}$ , we have:

$$\frac{\partial^2 P_s^{(t)}(\mathbf{b}^{(t)}, \mathbf{p}^{(t)})}{\partial p_s^{(t)2}} = -2c_s^{(t)} \frac{\alpha_u}{p_s^{(t)3}} \sum_{u \in U} \frac{B_{-u}^{(t)}}{d_u^{(t)}} < 0 \quad (3.18)$$

Thus,  $p_s^{(t)*}$  maximizes the MEC server's profit  $P_s^{(t)}(\mathbf{b}^{(t)}, \mathbf{p}^{(t)})$ .



### 3.3.4 Data Offloading and MEC Server Selection Algorithm

We introduce an iterative and low-complexity decision-making algorithm towards realizing the Data Offloading and MEC Server Selection (DO-MECS algorithm). The DO-MECS algorithm consists of two main components. At the first component, the MEC server selection by the end-users is executed following the theory of the Stochastic Learning Automata, as presented in Section 3.3.2. Then, at the second component of the DO-MECS algorithm, the end-users' optimal data offloading and the MEC servers' optimal pricing are determined, as presented in Section 3.3.3. It is noted that the first part of the DO-MECS algorithm runs at the beginning of each time slot, while the second part of the algorithm runs for multiple iterations within each time slot.

DO-MECS Algorithm:

*Step 1 (Initialization):* At the first time slot  $t = 0$ , set the initial MEC server selection probability vector as  $\mathbf{Pr}_u(\mathbf{t} = \mathbf{0})$ , where  $Pr_{u,s}(t = 0) = \frac{1}{S}, \forall u \in U, s \in S$ .

*Step 2 (MEC Server Selection):* At the beginning of each time slot ( $t > 0$ ), each end-user chooses a MEC server to offload its data based on its action probability vector  $\mathbf{Pr}_u(\mathbf{t})$ . If  $Pr_{u,s}(t) \geq 0.999$  for all the MEC servers  $s, s \in S$ , then stop. Otherwise, set  $i = 0$ , where  $i$  denotes the iteration of the second part of the algorithm.

*Step 3 (Optimal Data Offloading):* Each end-user has been associated with a MEC server and all the MEC servers announce their prices. Each end-user determines its optimal data offloading based on Equation 3.12.

*Step 4 (Optimal Pricing):* Given the end-users' offloading data, each MEC server determines the optimal pricing of its computing services based on Equation 3.16.

*Step 5 (Convergence):* If  $|b_{u,s}^{(t)*}|_{i+1} - b_{u,s}^{(t)*}|_i| \leq \epsilon_1$  and  $|p_s^{(t)*}|_{i+1} - p_s^{(t)*}|_i| \leq \epsilon_2, \forall s \in S, u \in U$ , where  $\epsilon_1, \epsilon_2$  (small positive constants) are the convergence control parameters, then stop. Otherwise, go to Step 3.

*Step 6 (Update):* Update the end-users' action probabilities based on Equation 3.9a

and Equation 3.9b and return to Step 2.

### 3.3.5 Empirical Evaluation

We provide a detailed empirical evaluation that illustrates the operation, features and benefits of the proposed DO-MECS framework. We focus on the pure operational characteristics of our framework, while a comparative evaluation of our approach against alternative methodologies is provided. Unless otherwise explicitly indicated, a detailed Monte Carlo analysis [210] has been executed for all the presented numerical results considering averages over 1000 executions.

#### Operation of the DO-MECS Framework

Table 3.1: MEC servers' characteristics

	Cost $c$	Discount $f_s$
Server 1	0.12	0.05
Server 2	0.14	0.04
Server 3	0.20	0.02
Server 4	0.17	0.03
Server 5	0.13	0.05

Towards illustrating the successful operation of the DO-MECS framework, we performed detailed simulations considering two main cases regarding the end-users that reside within the MEC environment: a) homogeneous end-users, and b) heterogeneous end-users, regarding their sensitivity on the pricing imposed by the MEC servers (i.e., end-user dynamics  $d_u^{(t)}$  in Equation 3.2). In our simulations, we consider  $S = 5$  MEC servers and  $U = 100$  end-users, while for demonstration purposes the weights  $w_1, w_2, w_3$  in Equation 3.4 have been considered of same importance, and each one equal to  $1/3$ . The parameters that characterize the different MEC servers are presented in Table 3.1.

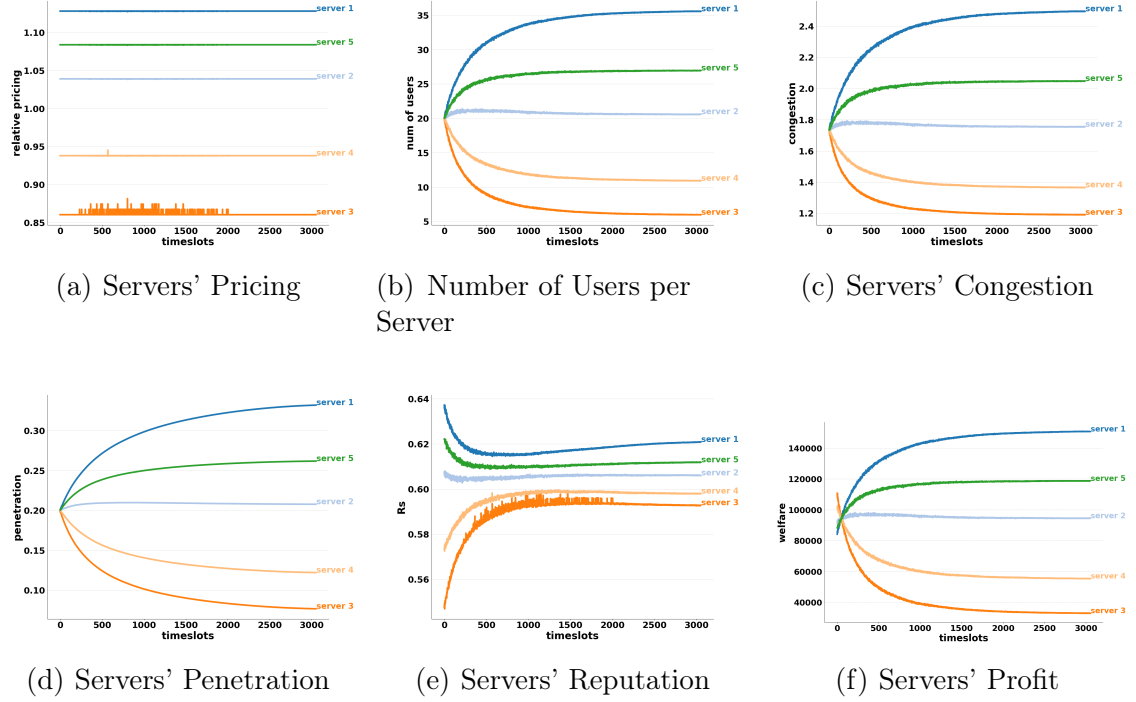


Figure 3.2: Proposed framework's operation per time slots - Homogeneous Users

**Homogeneous End-Users:** Initially, concerning the scenario of homogeneous end-users, Figure 3.2 presents, in a comprehensive manner, indicative numerical results regarding the pure operation of the DO-MECS algorithm, to gain some insight into the key operational characteristics and contributions of the various components of our framework. Specifically, Figure 3.2a presents the relative pricing of each MEC server, i.e.,  $\frac{\sum_{k \neq s} [(1-f_k^{(t)})] p_k^{(t)}}{(1-f_s^{(t)}) p_s^{(t)}}$ , as it is determined at the end of each time slot with respect to the time slots that DO-MECS algorithm needs to converge. It is observed that in all cases convergence is obtained in less than 3000 time slots, corresponding to the actual running time of less than 14 seconds for learning rate  $b = 0.2$ . Note that significantly lower convergence times can be achieved if higher learning rates are considered, as demonstrated later in Section Comparative Evaluation. As it is presented in Figure 3.2a and Figure 3.2b, the greater is the relative pricing for each

### Chapter 3. Mobile Edge Computing

MEC server, the more attractive it becomes for the end-users. Server 1 accumulates the majority of the end-users since from Table 3.1 we notice that Server 1 has both the smallest cost and offers the highest discount compared to the other MEC servers. The same trend and reasoning follow for the rest of the servers. Please note here that due to the homogeneity of the considered population each end-user offloads the same amount of data (in this experiment offloads its total data, i.e.,  $I_u^{(t)} = 1000\text{Bytes}$ ), to the corresponding selected MEC server, as determined by the MEC Server Selection process (Step 2 of DO-MECS Algorithm) based on the theory of the Stochastic Learning Automata (Section 3.3.2). In the following, a different scenario with heterogeneous end-users is considered and demonstrated, where the end-users decide to offload different amounts of data, based on the overall system dynamics.

As expected, the congestion on each MEC server, i.e.,  $CONG_s = \frac{\sum_{u \in U} b_{u,s}^{(t)}}{B_s^{Max}}$ , follows the same trend as the number of end-users selecting each MEC server (Figure 3.2b). The latter observation is expected, as the more end-users select to offload their data to a MEC server, the more congested that MEC server becomes (Figure 3.2c) and a greater penetration, i.e.,  $\frac{\sum_{t \in \{1, \dots, T\}} \sum_{u \in U} b_{u,s}^{(t)}}{\sum_{s \in S} \sum_{t \in \{1, \dots, T\}} \sum_{u \in U} b_{u,s}^{(t)}}$ , is achieved by that server. In particular, the MEC servers' penetration in serving the end-users computing demands is presented in Figure 3.2d. Furthermore, from Equation 3.4, we observe that the reputation score  $R_s$  depends on the relative pricing, the congestion, and the penetration of the MEC servers. The  $R_s$  essentially controls the probability based on which each end-user will select a server to offload its data. In Figure 3.2e, the results illustrate that the proposed DO-MECS framework tries to boost "weaker" servers in order to allow them to gain some traction on the market. Additionally, Figure 3.2f presents the profit  $P_s^{(t)}(\mathbf{b}^{(t)}, \mathbf{p}^{(t)})$  that each MEC server receives based on the its price announcement and the end-users' data offloading. The results reveal that Server 1 achieves the highest profit due to the combined effect of having the lowest cost (Table 3.1) and attracting a large number of end-users, even though it presents the lowest price as shown in Figure 3.2a. The same trend is followed from the rest of

the servers, which indicates that the announced price by the MEC server is not the only dominant factor in shaping the server's profit, but also the number of end-users that select to be served by a server is a key parameter in determining the server's overall profit.

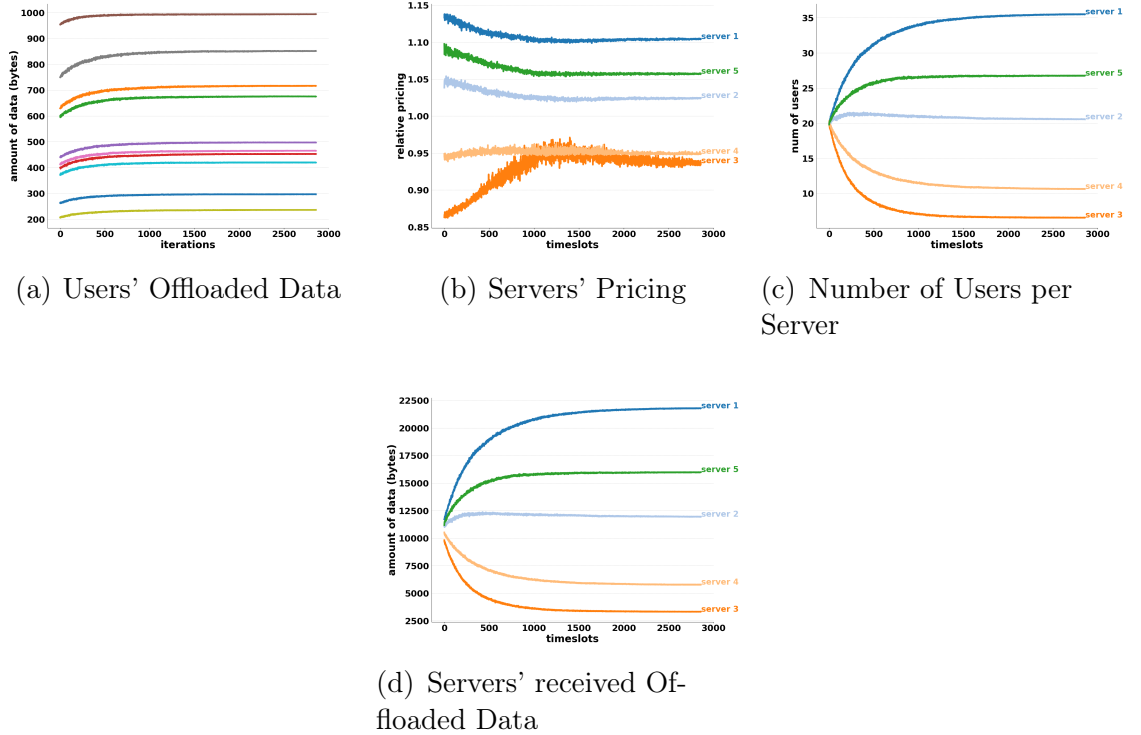


Figure 3.3: Proposed framework's operation per time slots - Heterogeneous Users

**Heterogeneous End-Users:** Here, we consider the scenario of heterogeneous end-users, i.e., different spending dynamics (i.e.,  $d_u^{(t)}$ ) and therefore potentially may offload different parts of their total data  $I_u^{(t)}$  to the selected MEC server. Specifically, in Figure 3.3a, we present the convergence of the amount of offloaded data for 10 indicative end-users from the overall available set in the simulated scenario. The results indicate that as the end-users have different spending dynamics, the announced price by each MEC server has a different impact on each end-user in terms of de-

termining its amount of offloaded data. Due to the differentiation of the end-users' spending dynamics, the MEC servers are motivated to adjust their announced prices to better adapt to the volume of the end-users' offloaded data. The aforementioned behavior is captured in Figure 3.3b, where it is observed that the "weaker" servers are willing to drop their price to increase their stability and penetration on the market, while the stronger ones increase their price to avoid congestion. Moreover, in Figure 3.3c and Figure 3.3d, the total number of end-users per MEC server and the corresponding amount of offloaded data per MEC server are presented, respectively.

### Comparative Evaluation

We present a comparative analysis of the performance of our proposed framework against some alternative strategies, to reveal its benefits and advantages. Initially, we present the impact of the learning rate parameter of the Stochastic Learning Automata (see Chapters 2.2.3, 3.3.2) in the operation of the DO-MECS framework, while we evaluate the benefits and drawbacks of different data offloading mechanisms as well. The learning rate parameter  $b$  of the Stochastic Learning Automata

Table 3.2: Execution Time for different Learning Rates

Learning Rate	Execution Time (sec)
$b = 0.1$	147.2
$b = 0.2$	27.5
$b = 0.3$	11.6
$b = 0.4$	6.4
$b = 0.5$	4.2

algorithm (*policy-based* RL technique - see Chapter 2.2.3) is an important factor regarding the convergence of the DO-MECS framework to the optimal stable state, as it expresses a trade-off between exploration and exploitation [62]. Greater values of the learning rate would lead to faster convergence, however, smaller ones allow the end-users to better exploit the available options and ultimately conclude to better

states. To demonstrate the above tradeoff, a comparative evaluation between different values of the learning rate is performed. Table 3.2 shows the average execution time of our DO-MECS framework until convergence is achieved, while Figure 3.4a and Figure 3.4b present the average MEC server's profit and the average end-user's utility for different learning rates, respectively. Indeed, it is observed that small values of the learning rate parameter  $b$  conclude to slow convergence of the DO-MECS algorithm, however, they allow the MEC servers and the end-users to achieve higher average profit and higher average utility, respectively. Based on Figure 3.4a and Figure 3.4b, we can see that the difference in the convergence state (i.e., average MEC servers' profit and average end-users' utility) between learning rates  $b = 0.1$  and  $b = 0.2$  is negligible, while the difference in the convergence time is significant. This is also evident from the execution times presented in Table 3.2, where for  $b = 0.2$  the DO-MECS algorithm converges five times faster than in the case where  $b = 0.1$ . Thus, a learning rate of  $b = 0.2$  presents a good balance between optimality and efficiency.

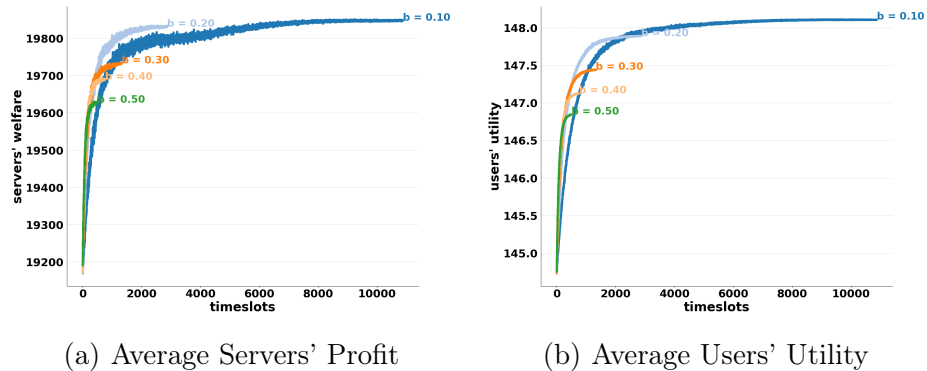


Figure 3.4: Comparative Evaluation per time slots - Different Learning Rates

Towards evaluating the significance of our proposed autonomous data offloading mechanism, i.e., DO-MECS framework, a comparison between our mechanism and a computationally simplistic offloading policy where each end-user offloads a fixed por-

tion (i.e. percentage) of its data, was performed, while for fairness purposes the rest of our proposed framework (i.e. server selection and optimal pricing mechanisms) was kept intact in all strategies. Specifically, concerning the alternative data offloading mechanism three different variations were examined, where the end-users send 25%, 58.6% and 100% of their total data  $I_u^{(t)}$ , respectively, to the selected MEC servers. It should be noted here that the alternative with a fixed percentage of 58.6% was selected because it corresponds to the same exactly average end-user data offloading, as the one produced by our proposed framework in the considered experiment.

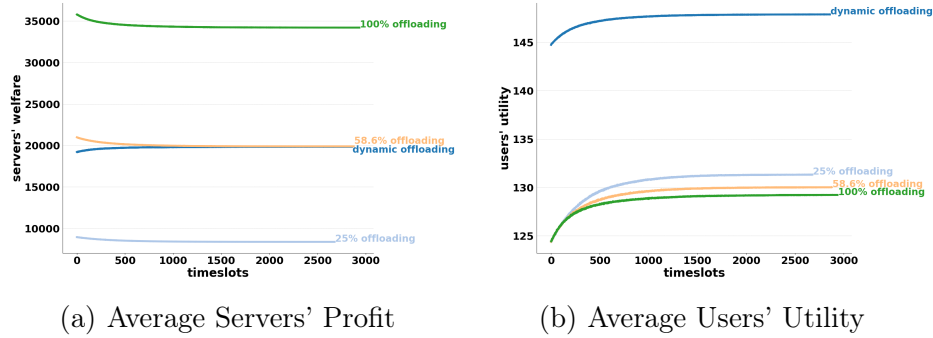


Figure 3.5: Comparative Evaluation per time slots - Different Offloading Policies

The corresponding comparative results are depicted in Figure 3.5a and Figure 3.5b, where the average MEC servers' profit and the average end-users' utility, respectively, as a function of the time for the different offloading mechanisms are obtained. In particular, it is evident that as expected the more data the end-users offload to the MEC servers, the higher profit the MEC servers experience. However, this happens at the cost of a very low average utility experienced by the end-users, as clearly demonstrated from the curves corresponding to the 100% offloading alternative. Moreover, it is observed that allowing the end-users to send a constant amount of data without enabling them to dynamically adapt their offloading amount of data based on the system's conditions (as our framework evangelizes), always results in significantly lower average end-users' utility. As a result, the proposed DO-MECS framework of-



fers incentives to the end-users to participate in the non-cooperative data offloading game to dynamically and autonomously determine the optimal amount of data, while the MEC servers experience the best levels of profit that they can achieve based on the decisions of their customers, i.e., end-users.

### **3.3.6 Summary**

In the proposed autonomous decision-making framework, the joint problem of MEC server selection by the end-users, along with their optimal data offloading and the optimal price setting by the MEC servers is studied in multiple MEC servers and multiple end-users environments. The flexibility and programmability offered by the SDN technology enable the realistic implementation of the proposed framework. In particular, the MEC server selection part of the framework is based on a Reinforcement Learning technique adopting the theory of the Stochastic Learning Automata. The end-users optimal data offloading and the MEC servers' optimal pricing of their computing services is formulated as a two-layer optimization problem.

At the first layer, a non-cooperative game among the end-users of each server is formulated towards maximizing the perceived satisfaction of each end-user, as expressed by an appropriately formulated utility function. The existence and uniqueness of the game's NE point are shown, thus concluding to the end-users' optimal data offloading strategy. At the second layer of the proposed framework, an optimization problem of each MEC server's profit is formulated and the corresponding optimal price of its computing services is determined. A low-complexity Data Offloading and MEC Server Selection (DO-MECS) algorithm are introduced to realize the overall framework. The operation and performance of the proposed framework were extensively evaluated through modeling and simulation, while the presented detailed numerical results demonstrate its performance and benefits in the examined setting.

### 3.4 Satisfaction-aware Data Offloading in Surveillance Systems

We exploit the Fully Autonomous Aerial Systems' (FAAS) and the Mobile Edge Computing (MEC) servers' computing capabilities to introduce a novel data offloading framework and support the energy and time-efficient video processing in surveillance systems based on the satisfaction games. A surveillance system is introduced consisting of Areas of Interest (AoIs), where a MEC server is associated with each AoI and a FAAS is flying above the AoIs to support the IP cameras' computing demands. Each IP camera adopts a utility function capturing its Quality of Service (QoS) considering the experienced time and energy overhead to offload and process remotely or locally the data. A non-cooperative game among the cameras is formulated to determine the amount of offloading data to the MEC server and/or the FAAS, and the novel concept of Satisfaction Equilibrium (SE) is introduced where the IP cameras satisfy their minimum QoS prerequisites instead of maximizing their performance by consuming additional system resources. A distributed learning algorithm determines the IP cameras' stable data offloading. Also, a reinforcement learning algorithm indicates the FAAS's movement among the AoIs exploiting the accuracy, timeliness, and certainty of the collected data by the IP cameras per AoI. Detailed numerical and comparative results are presented to show the operation and efficiency of the proposed framework.

Generally, surveillance systems have recently gained great attention due to the increased number of terrorist attacks, which challenge public safety and homeland security [211]. With the advent of the Internet of Things (IoT), the smart Internet Protocol (IP) cameras have enabled the surveillance systems to capture real-time video and process it locally [212] or remotely in the cloud computing environment. However, the surveillance systems confront the challenges of increased computing demand to process the recorded information.

### *Chapter 3. Mobile Edge Computing*

The authors in [213] have introduced an image uploading process from the IP cameras to the cloud, where the images captured by the IP cameras are stored and processed at the cloud to decrease the cost of storing and processing the information locally. In [214], a drone-assisted surveillance system is studied, where the videos captured by the drone are forwarded to Fog Computing nodes through the drones' ground controller, to be processed and track vehicles' movement. Following a similar philosophy, in [215] an Unmanned Aerial Vehicle (UAV)-based crowd surveillance system is introduced where the UAVs capture videos that either offload to MEC servers for further processing or process them on board. The authors discuss the drawbacks and benefits of the two choices in terms of energy consumption and processing time. In [216], the drones' video capturing capability is exploited to track the moving target object and the drones offload part of the computing tasks to a control center, while the rest are executed locally at the drone.

The UAVs, the drones, and the remote computing capabilities (i.e., Cloud, MEC, and Fog Computing) have improved the performance of the surveillance systems. However, the UAVs and the drones still require human control from the ground to indicate the path that they follow during their flight. To address this issue, the FAAS has been recently introduced in the robotics and automation research field [217]. The FAAS is a flying robotic system equipped with sensors, surveillance systems, computing resources, wireless communication interfaces, or any combination of them and can operate fully autonomously with no human intervention.

In all the aforementioned approaches, each entity involved in the surveillance system (i.e., IP cameras) aims at maximizing its QoS. The improvement of those entities' QoS is proportional to the consumption of communication and computing resources. However, a surveillance system is a resource-constrained setting, thus the maximization of each involved entity's QoS is a sub-optimal solution. Towards this direction, the games in satisfaction form have been introduced in the field of Game Theory [218], where the autonomous entities aim to "satisfy" their minimum

QoS prerequisites in a distributed manner instead of targeting at maximizing their QoS, and have been applied in the uplink power control problem in wireless cellular networks [85]. Here, the FAAS's and the MEC servers' computing capabilities are exploited to introduce a novel data offloading framework based on the satisfaction games in order ultimately to support the energy and time-efficient video processing in surveillance systems consisting of IP cameras.

### 3.4.1 System Model

A surveillance area of size  $L \times L$  consisting of AoIs (e.g., banks, airports) is considered, where the set of AoIs is  $\mathbb{A} = \{1, \dots, i, \dots, A\}$  and they are randomly placed with coordinates  $Z_i = (X_i, Y_i)$ ,  $X_i, Y_i \leq L$ . Each AoI has a set of IP cameras  $\mathbb{C}_i = \{1, \dots, j, \dots, C_i\}$  that collect and process data [212] and a MEC server  $M_i$ , which supports their computing demands. A FAAS flies between the AoIs with a velocity  $v$  and altitude  $d$ . At each timeslot  $t$ , the FAAS receives and processes data from the AoI's cameras of which the FAAS is located above. The set of timeslots is  $\mathbb{T} = \{1, \dots, t, \dots, T\}$  and at each timeslot, the FAAS is located above only one AoI. The set of collected data by each IP camera  $j \in \mathbb{C}_i$  belonging to the AoI  $i$  per timeslot  $t$  is denoted as  $D_{ij}^{(t)} = (B_{ij}^{(t)}, CP_{ij}^{(t)}, \phi_{ij}^{(t)}, dt_{ij}^{(t)}, de_{ij}^{(t)})$ , where  $B_{ij}^{(t)}$  [bits] is the total collected information,  $CP_{ij}^{(t)} = \phi_{ij}^{(t)} B_{ij}^{(t)}$  is the number of required CPU cycles to process the data, where  $\phi_{ij}^{(t)} > 0$  is the level of the video processing task's intensity,  $dt_{ij}^{(t)}$  denotes the time constraint during which the data should be processed, and  $de_{ij}^{(t)}$  is the IP camera's energy availability for the timeslot  $t$ . The amount of collected data  $B_{ij}^{(t)}$  can be partitioned into subsets of specific size, which can be offloaded for remote processing to the MEC server  $M_i$  or the FAAS, assuming that the last one is located at the AoI  $i$  for the timeslot  $t$ . For the rest of the analysis, we drop the  $(t)$  for notational convenience, since the same hold true  $\forall t \in \mathbb{T}$ .

### Communication and Computation Model

We denote  $\mathbf{s}_i = (\mathbf{s}_{i1}, \dots, \mathbf{s}_{ij}, \dots, \mathbf{s}_{iC_i})$  the vector of strategies for the IP cameras residing in the AoI  $i$ , where  $\mathbf{s}_{ij} = (ch_{ij}, a_{ij})$  and  $a_{ij} \in [0, 1]$  is the IP camera's data offloading percentage, and  $ch_{ij} = 0$  if the IP camera offloads its  $a_{ij} \cdot B_{ij}$  amount of data to the MEC server  $M_i$ , while  $ch_{ij} = 1$  if it offloads to the FAAS. Therefore, considering that the FAAS is located at the AoI  $i$ , then for each other AoI  $i' \in \mathbb{A}, i' \neq i$  it holds true that  $ch_{i'j} = 0, \forall j \in \mathbb{C}_{i'}$ . Thus, since the AoIs do not interfere with each other due to their distant locations, the IP camera's  $j$  in AoI  $i$  uplink data rate is:

$$R_{ij} = W_i \cdot \log\left(1 + \frac{p_{ij}g_{ij}}{\sigma_0^2 + \sum_{k \in \mathbb{C}_i \setminus \{j\}, ch_{ik}=ch_{ij}, a_{ik} \neq 0} p_{ik}g_{ik}}\right) \quad (3.19)$$

where  $W_i$  is the AoI's  $i$  bandwidth,  $p_{ij}$  is the IP camera's  $j$  transmission power to offload part of its data,  $g_{ij}$  is the channel gain between the IP camera  $j$  and the MEC server  $M_i$  (if  $ch_{ij} = 0$ ) or the FAAS (if  $ch_{ij} = 1$ ), and  $\sigma_0^2$  indicates the background noise power. The IP camera  $j$  in the AoI  $i$  experiences the data transmission time overhead  $O_{ij}^{tr,t} = \frac{a_{ij} \cdot B_{ij}}{R_{ij}} [sec]$  by offloading  $a_{ij}B_{ij}$  amount of data and the data transmission energy consumption  $O_{ij}^{tr,e} = p_{ij} \frac{a_{ij} \cdot B_{ij}}{R_{ij}} [Joules]$ . Each MEC server  $M_i$  and the FAAS have the computing capability  $f_{M_i}$  and  $f_F [Cycles/sec]$  respectively, which is shared among the IP cameras that are being served by them. The allocated computing capability to each IP camera  $j$  in order to remotely process its offloaded data is given as:

$$f_{ij} = \frac{a_{ij}B_{ij}\phi_{ij}}{\sum_{k \in \mathbb{C}_i \setminus \{j\}, ch_{ik}=ch_{ij}} a_{ik}B_{ik}\phi_{ik}} \cdot ((1 - ch_{ij})f_{M_i} + ch_{ij}f_F) \quad (3.20)$$

where the first factor of Equation 3.20 reveals that an IP camera with a higher processing intensity (i.e.,  $\phi_{ij}$ ) and greater amount of offloaded data acquires a higher computing capability, while the second one reveals that each IP camera  $j$  can offload a part of its data to only one computing resource (i.e., either the MEC server  $M_i$  or the FAAS). Based on the IP camera's  $j$  remote computing capability (Equation

3.20), its offloaded data processing time overhead is  $O_{ij}^{p,t} = \frac{a_{ij}B_{ij}\phi_{ij}}{f_{ij}}$ . Moreover, the IP camera  $j$  has a local computing capability  $f_{ij}^l[Cycles/sec]$  and processes the rest  $(1 - a_{ij})B_{ij}$  data locally. Thus, its local processing time overhead is  $\frac{(1-a_{ij})B_{ij}\phi_{ij}}{f_{ij}^l}$  and its local processing energy overhead is  $(1 - a_{ij})B_{ij}\phi_{ij}e_{ij}$ , where  $e_{ij}[J/Cycle]$  is its local energy consumption to process the data. The IP camera's  $j$  overall time overhead is given as follows.

$$O_{ij}^t = \max \left\{ \frac{a_{ij} \cdot B_{ij}}{R_{ij}} + \frac{a_{ij}B_{ij}\phi_{ij}}{f_{ij}}, \frac{(1 - a_{ij})B_{ij}\phi_{ij}}{f_{ij}^l} \right\} \quad (3.21)$$

while its overall energy consumption is formulated as:

$$O_{ij}^e = p_{ij} \frac{a_{ij} \cdot B_{ij}}{R_{ij}} + (1 - a_{ij})B_{ij}\phi_{ij}e_{ij} \quad (3.22)$$

### 3.4.2 Towards IP Cameras' Satisfaction

Each IP camera aims to satisfy its QoS prerequisites expressed in terms of time  $dt_{ij}$  and energy  $de_{ij}$  demands by offloading an amount of data and processing the rest locally. Thus, we formulate a generic utility function that represents each IP camera's QoS as follows.

$$u_{ij}(\mathbf{s}_{ij}, \mathbf{s}_{-ij}) = \begin{cases} -\left(\frac{dt_{ij} - O_{ij}^t}{dt_{ij}}\right) \cdot \left(\frac{de_{ij} - O_{ij}^e}{de_{ij}}\right) & \text{if } O_{ij}^t \geq dt_{ij}, O_{ij}^e \geq de_{ij} \\ \left(\frac{dt_{ij} - O_{ij}^t}{dt_{ij}}\right) \cdot \left(\frac{de_{ij} - O_{ij}^e}{de_{ij}}\right) & \text{otherwise} \end{cases} \quad (3.23)$$

where  $\mathbf{s}_{-ij}$  is the strategy vector of all the IP cameras of the AoI  $i$  except the IP camera  $j$ . Assuming that the FAAS is located at the AoI  $i$ , it is evident by Equation 3.21, that when the IP camera's  $j$  chosen computing resource (i.e., the MEC server or the FAAS) is overloaded, then its perceived time and energy overhead increase, and its utility value  $u_{ij}$  is negative if the IP camera does not satisfy at least one of its QoS prerequisites (i.e.,  $dt_{ij}$ ,  $de_{ij}$ ). Thus, each IP camera  $j$  aims to fulfill its time and energy demands, i.e.,  $u_{ij} \geq 0$ , via autonomously determining its offloading strategy  $\mathbf{s}_{ij} = (ch_{ij}, a_{ij})$ . A non-cooperative game is played among the IP cameras per AoI to determine a stable data offloading vector that fulfills the IP cameras' QoS prerequisites. The game is written in the satisfaction form

### Chapter 3. Mobile Edge Computing

(Chapter 2.1)  $G_i = [\mathbb{C}_i, \{S_{ij}\}_{j \in \mathbb{C}_i}, \{u_{ij}\}_{j \in \mathbb{C}_i}, \{h_{ij}\}_{j \in \mathbb{C}_i}]$ , where  $\mathbb{C}_i$  is the set of the IP cameras in the AoI  $i$ , and considering that the FAAS is located in the AoI  $i$ , then  $S_{ij} = \{(a_{ij}^n, 0), \dots, (a_{ij}^N, 0), \dots, (a_{ij}^n, 1), \dots, (a_{ij}^N, 1)\}$ , while if the FAAS is not located in the AoI  $i$ , then  $S_{ij} = \{(a_{ij}^n, 0), \dots, (a_{ij}^N, 0)\}$ , and  $a_{ij}^n$  is the  $n^{th}$  available offloading percentage, thus  $a_{ij}^n \in [0, 1]$ ,  $\forall n \leq N, N \in \mathbb{N}$ . Moreover,  $u_{ij}$  is the AoI's  $i$  IP camera's  $j$  utility as expressed in Equation 3.23, and  $h_{ij}$  is the satisfaction correspondence defined as follows [218].

$$h_{ij}(\mathbf{s}_{-ij}) = \{\mathbf{s}_{ij} \in S_{ij} | u_{ij}(\mathbf{s}_{ij}, \mathbf{s}_{-ij}) \geq 0\} \quad (3.24)$$

At the Satisfaction Equilibrium (SE) point (see definition in Chapter 2.1) of each non-cooperative game  $G_i$ , the IP cameras satisfy their minimum QoS prerequisites without overspending the system's resources, where the latter would occur if they were targeting at their QoS maximization. We consider that the computing resources per AoI can support the IP cameras' minimum QoS prerequisites, i.e., an SE exists, and our goal is to achieve the better exploitation of the system's resources by allocating the cameras' computing tasks.

Towards determining the SE for each non-cooperative game  $G_i$ , we propose the Distributed Learning Satisfaction Equilibrium Algorithm (DLSE) that allows the IP cameras to autonomously learn and converge to it. Each camera evaluates its utility (Equation 3.23) by receiving its allocated remote computing capability (Equation 3.20) from the MEC server or FAAS and the interference factor (i.e.,  $\sum_{k \in \mathbb{C}_i, ch_{ik}=ch_{ij}, a_{ik} \neq 0} p_{ik}g_{ik}$  in Equation 3.19) and converges to the strategy  $\mathbf{s}_{ij}^+$ . Assuming that the elements of the offloading strategy set  $S_{ij}$  are indexed with  $l_{ij}$ , thus  $\mathbf{s}_{ij}^{(l_{ij})}$  is the  $l_{ij}th$  offloading strategy, then  $l_{ij} \leq L_{ij}$ , and  $L_{ij} = 2N$  if the FAAS is located in the AoI  $i$ , otherwise  $L_{ij} = N$ . Let us denote the IP camera's  $j$  offloading strategy at instant  $r > 0$  as  $\mathbf{s}_{ij}(r) \in S_{ij}$ , where it is chosen following a discrete probability distribution  $\boldsymbol{\pi}_{ij}(r) = (\pi_{ij}^{(1)}(r), \dots, \pi_{ij}^{(l_{ij})}(r), \dots, \pi_{ij}^{(L_{ij})}(r))$ , where  $\pi_{ij}^{(l_{ij})}(r)$  is the probability with which the AoI's  $i$  IP camera  $j$  chooses its action  $\mathbf{s}_{ij}^{(l_{ij})}$  at instant  $r > 0$ .

---

**Algorithm 1** DLSE Algorithm

---

```

1: Input/Initialization: AoI  $i$ ,  $\mathbb{C}_i, D_{ij}, \forall j \in \mathbb{C}_i, conv = 0, r = 1$ 
2: Output:  $\mathbf{s}_i^+ = (\mathbf{s}_{i1}^+, \dots, \mathbf{s}_{ij}^+, \dots, \mathbf{s}_{iC_i}^+)$ 
3: Each  $j \in \mathbb{C}_i$  sets  $S_{ij}, L_{ij}$  based on the FASS's existence
4:  $\pi_{ij}^{(l_{ij})}(0) = \frac{1}{L_{ij}}, \forall j \in \mathbb{C}_i, l_{ij} \leq L_{ij}$ 
5: Each  $j \in \mathbb{C}_i$  picks  $\mathbf{s}_{ij}(0)$  based on  $\boldsymbol{\pi}_{ij}(0)$ , and evaluates  $U_{ij}$ 
6: while conv == 0 do
7:   for  $j = 1$  to  $C_i$  do
8:      $u_{ij} = u_{ij}(\mathbf{s}_{ij}(\mathbf{r} - 1), \mathbf{s}_{-ij}(\mathbf{r} - 1)), b_{ij}(r) = \frac{U_{ij} + u_{ij}}{2U_{ij}}$ 
9:     if  $u_{ij} \geq 0$ , then  $\mathbf{s}_{ij}(r) = \mathbf{s}_{ij}(r - 1), \boldsymbol{\pi}_{ij}(r) = \boldsymbol{\pi}_{ij}(r - 1)$ 
10:    else  $\forall l_{ij} \leq L_{ij}$ 
11:       $\pi_{ij}^{(l_{ij})}(r) = \pi_{ij}^{(l_{ij})}(r - 1) + \lambda_{ij} b_{ij}(r) (1_{\{\mathbf{s}_{ij}^{(l_{ij})} = \mathbf{s}_{ij}(r-1)\}} - \pi_{ij}^{(l_{ij})}(r - 1))$ 
12:    end for
13:    if SE or GSE point reached then conv = 1
14:    else  $r = r + 1$ 
15: end while

```

---

Each IP camera's initial policy (Chapters 2.2, 2.2.3) is approximated through a uniform probability distribution, thus  $\pi_{ij}^{(l_{ij})}(r = 0) = 1/L_{ij}, \forall l_{ij} \leq L_{ij}$ , where  $L_{ij}$  is the number of the IP camera's  $j$  offloading strategies. Let  $U_{ij}$  denote the maximum utility that each IP camera  $j$  perceives if it was the only one inside the AoI  $i$ . Each IP camera updates its probability distribution  $\boldsymbol{\pi}_{ij}$  based on a learning parameter  $\lambda_{ij}$ , so that higher probabilities are allocated to offloading actions which lead the IP camera  $j$  to perceive a higher utility  $u_{ij}$ . Let us introduce the definition of a clipping action, which is considered for the study of the DLSE Algorithm's convergence to an SE point.

**Definition 7 (Clipping Action)** *At each non-cooperative game  $G_i$ , an IP camera  $j$  has a clipping action  $\mathbf{s}_{ij}^c \in S_{ij}$  iff  $\forall \mathbf{s}_{-ij} \in S_{-ij}, \mathbf{s}_{ij}^c \in h_{ij}(\mathbf{s}_{-ij})$ , where  $S_{-ij} = S_{i1} \times$*



$$\cdots \times S_{i(j-1)} \times S_{i(j+1)} \times \cdots \times S_{iC_i} \text{ [85].}$$

Therefore, Definition 7 reveals that once an IP camera concludes to a clipping action  $\mathbf{s}_{ij}^c$  at an instance  $r'$  of the DLSE Algorithm, then  $\forall r \geq r'$  the IP camera keeps the same offloading strategy, i.e.,  $\mathbf{s}_{ij}(r) = \mathbf{s}_{ij}^c$ . Thus, assuming that there exists an IP camera  $j' \neq j$ , such that its satisfaction correspondence  $h_{ij}(\mathbf{s}_{-ij'j}) = \emptyset$ ,  $\forall \mathbf{s}_{-ij'j} \in S_{-ij'j}$ , where  $\mathbf{s}_{-ij'j}$  is the offloading strategy vector of all the IP cameras except the camera  $j'$  and the IP camera  $j$  which plays its clipping action  $\mathbf{s}_{ij}^c$ , and  $S_{-ij'j}$  is the corresponding set of vectors, then the DLSE Algorithm converges to a Generalized SE (GSE) point, which is considered as generalization of the SE point (see Chapter 2.1), and whose definition is given as:

**Definition 8 (Generalized SE)** *A strategy profile is a GSE  $\mathbf{s}_i^- = (\mathbf{s}_{i1}^-, \dots, \mathbf{s}_{iC_i}^-)$  of the non-cooperative game  $G_i$ , if there exists a partition of the  $\mathbb{C}_i$  given by  $\mathbb{C}_i^s$  and  $\mathbb{C}_i^u$ , such that  $\forall j \in \mathbb{C}_i^s$ ,  $\mathbf{s}_{ij} \in h_{ij}(\mathbf{s}_{-ij})$  and  $\forall j' \in \mathbb{C}_i^u$ ,  $h_{ij'}(\mathbf{s}_{-ij'}) = \emptyset$ .*

Given the DLSE algorithm and the existence of at least one SE point for each game  $G_i$ , and that there is no clipping action, then the DLSE Algorithm converges to the SE point for each game  $G_i$ . Otherwise, in the existence of a clipping action  $\mathbf{s}_{ij}^c$  for at least one IP camera, the DLSE Algorithm converges to a GSE point.

### 3.4.3 FAAS Movement based on Reinforcement Learning

we deploy three different quality factors that capture the QoI that each AoI's surveillance system provides. We consider that several algorithms (e.g., object detection, move detection) can be executed locally at the IP cameras' and remotely at the MEC servers' and FAAS's computing resources to assess each AoI's QoI. These algorithms assign values to the following quality factors at the end of each timeslot based on each IP camera's captured data.

(a) **Accuracy** refers of how the observed information inside each AoI conforms to the reality. After the processing of the IP cameras' collected data, the number of

### Chapter 3. Mobile Edge Computing

the correctly detected events  $AE_{ij}$  is evaluated and the IP camera's accuracy is  $q_{ij}^{acc} = \frac{AE_{ij}}{TE_{ij}}$ , where  $TE_{ij}$  is the total number of events that were captured. The overall accuracy of the AoI  $i$  is defined as  $Q_i^{acc} \triangleq \frac{1}{C_i} \sum_{j \in \mathbb{C}_i} q_{ij}$ .

(b) **Timeliness** refers to the availability of the information at the desired time. The timeliness factor is defined as  $q_{ij}^{tls} \triangleq \frac{D_t}{D_t + O_{ij}^t}$ , where  $D_t$  is the duration of each timeslot  $t$  and  $O_{ij}^t$  is the IP camera's overall time overhead to offload and process the data. The AoI's overall timeliness factor is  $Q_i^{tls} \triangleq \frac{1}{C_i} \sum_{j \in \mathbb{C}_i} q_{ij}^{tls}$ .

(c) **Certainty**: refers to the measurement of confirmation of the information and is strictly related to each IP camera's hardware characteristics (e.g., recording rate, sensor's pixels). In particular, this quality factor depicts the probability of error regarding the captured data of each IP camera and it is denoted as  $q_{ij}^{crt}$ . The overall certainty of the AoI  $i$  is evaluated as  $Q_i^{crt} \triangleq \frac{1}{C_i} \sum_{j \in \mathbb{C}_i} q_{ij}^{crt}$ .

Finally, each AoI's  $i, i \in \mathbb{A}$  overall QoI for a specific timeslot is based on the past QoI values and is given as follows:

$$QoI_i = w_i^{acc} \frac{\sum_{t' \leq t} Q_i'^{acc}}{t} + w_i^{tls} \frac{\sum_{t' \leq t} Q_i'^{tls}}{t} + w_i^{crt} \frac{\sum_{t' \leq t} Q_i'^{crt}}{t} \quad (3.25)$$

where  $w_i^{acc}, w_i^{tls}, w_i^{crt} \in [0, 1]$  are the corresponding weights of each quality factor. Here, the FAAS movement is formalized as a sequential decision-making problem, where the FAAS acting as an RL-agent (Chapter 2.2) aims to maximize a long-term objective. In particular, at each timeslot, the FAAS is located at an AoI  $i$ , and acts as a computing resource, it provides a higher QoS to the corresponding IP cameras since the MEC server  $M_i$  is less overloaded. Therefore, the IP cameras' QoS prerequisites could be more easily met with the existence of the FAAS at the AoI  $i$ , and the overall AoI's timeliness quality factor to be increased. The existence of the FAAS at the AoIs with high QoI is important since the overall surveillance system's performance and effectiveness could be increased by decreasing the delay between an event's detection and further actions (e.g., policy, ambulance). Also, the FAAS's limited energy availability should be considered for both FAAS's fly-

### Chapter 3. Mobile Edge Computing

ing movement and its role as a computing resource. Considering that the FAAS is located at the AoI  $i$ , and by denoting as  $E_P[\text{Joules/Cycle}]$  the FAAS's energy consumption to process the received data, then its processing energy consumption is  $E^p = E_P \cdot \sum_{j \in \mathbb{C}_i, ch_{ij}=1} a_{ij} B_{ij} \phi_{ij}$ . Furthermore, considering that the FAAS was located at the AoI  $i', i' \neq i$  at the previous timeslot and the FAAS's velocity is  $v$ , then its movement energy consumption is  $E^m = E_M \cdot \frac{\sqrt{(X_i - X_{i'})^2 + (Y_i - Y_{i'})^2}}{v}$ , where  $E_M[\text{Watts}]$  is the FAAS's constant consumed energy while moving with velocity  $v$ . Based on the above discussion, we formulate the reward (Chapter 2.2) that the FAAS experiences while visiting an AoI  $i$  as follows.

$$rw_i = -\frac{\epsilon_3 \cdot \frac{E^p + E^m}{E}}{\epsilon_1 \cdot QoI_i + \epsilon_2 \cdot P_i} \quad (3.26)$$

where  $P_i = \frac{|\mathbb{C}_i^s|}{C_i}$ ,  $\mathbb{C}_i^s = \{j \in \mathbb{C}_i | u_{ij} \geq 0\}$  denotes the ratio of the IP cameras that meet their QoS prerequisites (i.e.,  $dt_{ij}, de_{ij}$ ),  $E$  is the FAAS's energy availability, and  $\epsilon_1, \epsilon_2, \epsilon_3 \in [0, 1]$  denote the weights of the AoI's QoI, performance (i.e.,  $P_i$ ) and the FAAS's consumed normalized energy, respectively. The physical meaning of the negative reward value is that reward values closer to zero benefit the FAAS.

In the following, we adopt a Reinforcement Learning (RL) approach that enables the FAAS to autonomously learn its dynamic environment and decide which AoI to visit per timeslot towards maximizing its long-term objective (Equation 3.26) [62]. Two of the most widely used RL algorithms are the Q-learning [219] and SARSA [62] algorithms, as these were introduced in Chapter 2.2, which via stochastic approximation conditions lead the decision-maker to converge to its optimal decision policy with high probability [220],[221]. In our case, for the FAAS's sequential decision making problem (i.e., the AoI  $i$  that selects to be located at each timeslot  $t$ ) we deploy the SARSA algorithm, which first examines the uncertain environment (i.e., the set of AoIs  $\mathbb{A}$ ), and then derives the optimal strategy based on the model knowledge that has already been constructed. The FAAS interacts with the environment (i.e., surveillance system), receives the corresponding feedback, i.e., reward (Equation 3.26), and updates its policy.

### 3.4.4 Empirical Evaluation

A detailed numerical performance evaluation and comparative study of the proposed architecture is conducted through modeling and simulations. We consider a surveillance system consisting of  $A = 7$  AoIs with  $C_i = 30, \forall i \in \mathbb{A}$  cameras. The cameras are randomly distributed in an area with a radius less than  $L = 500m$  from each MEC server  $M_i$ . The considered application characteristics are  $B_{ij} \in [1000, 5000]KB$  and  $CP_{ij} \in [1000, 5000]MCycles$ . The IP cameras' strategy space consists of 11 data offloading strategies, where  $\mathbf{a}_{i,j} \in [0, 1]$  with step 0.1. Also, we have  $e_{ij} = 10^{-9}J/Cycle$ ,  $f_{ij}^l \in [10^{-2}, 10^{-1}]$ ,  $W_i = 5MHz$ ,  $\sigma_0^2 = 10^{-13}$ ,  $p_{i,j} \in [0, 1]W$ ,  $dt_{i,j} \in [0, \frac{CP_{ij}}{f_j}]sec$ ,  $de_{i,j} \in [0, CP_{ij}e_{ij}]J$ ,  $g_{i,j} = \frac{1}{d_{ij}^2}$ , where  $d_{ij}$  is the IP camera's  $j$  distance from the MEC server  $M_i$  or FAAS,  $w_i^{acc} = 0.333$ ,  $w_i^{tls} = 0.333$ ,  $w_i^{crt} = 0.333$ ,  $Q_i^{acc}, Q_i^{tls}, Q_i^{crt} \in [0, 1]$ ,  $E_P = 10^{-9}J$ ,  $E_M = 0.0013W$ ,  $E = 17.28 \cdot 10^6J$ ,  $\epsilon_1 = 0.35$ ,  $\epsilon_2 = 0.55$ ,  $\epsilon_3 = 0.10$ , the duration of a timeslot is  $1h$ , and  $v = 6.25m/s$  [222]. The following analysis demonstrates: (i) the pure operation and characteristics of the proposed framework, (ii) its scalability performance, and (iii) a comparative evaluation.

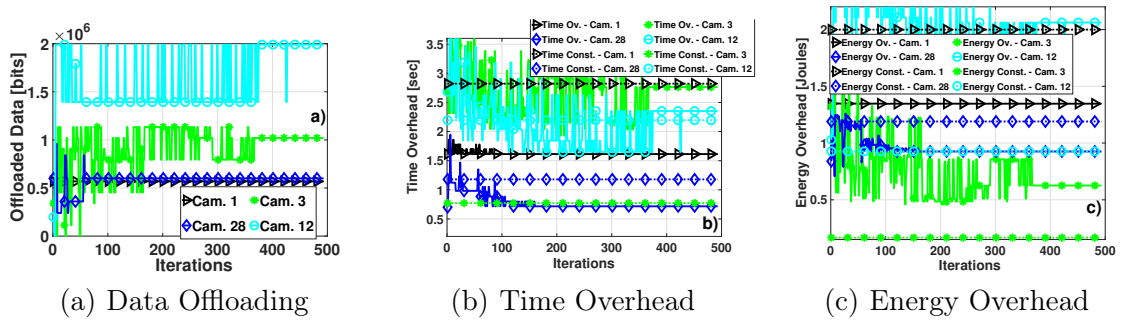


Figure 3.6: Pure operation of the proposed framework

#### Pure Operation of the Algorithm

In Figure 3.6.a-c, the amount of offloaded data, the time and the energy overhead, and constraints are presented for four IP cameras. The IP cameras with ID 12 and 3

### Chapter 3. Mobile Edge Computing

	Comp. Capab.		Distance		Time Const.		Energy Const.		Cameras		Percentages of Satisfied IP Cameras						
	Same	Order	Same	Order	Same	Order	Same	Order	Same	Order	AoI1	AoI2	AoI3	AoI4	AoI5	AoI6	AoI7
1		X	X		X		X		X		0.22	0.31	0.33	0.94	0.366	0.39	0.43
2	X			X	X		X		X		0.82	0.82	0.81	1	0.75	0.49	0.14
3	X		X			X			X		0.04	0.30	0.59	1	0.76	0.88	0.95
4	X		X		X			X	X		0.04	0.12	0.34	1	0.72	0.81	0.92
5	X		X		X		X			X	1	0.91	0.71	0.92	0.29	0.02	0

Figure 3.7: Percentage of Satisfied Cameras regarding different AoIs' characteristics and Cameras' QoS prerequisites

have strict time and energy constraints (Figure 3.6.b,c), thus they choose to offload a large amount of data to the MEC server to satisfy their QoS prerequisites. However, it is observed that even if they choose such a strategy, they cannot meet their QoS demands and the DLSE algorithm converges to a GSE point. On the other hand, the IP cameras with ID 1 and 28 have a relaxed time and energy constraints and they achieve to satisfy them, while the stricter the constraints are, the less time and energy overhead they experience, and the more data they offload.

Several factors influence whether or not an IP camera meets its QoS prerequisites, such as the MEC server's computing capability, IP cameras' average distance from the MEC server, time and energy constraints, and the number of cameras per AoI. In Table 3.7, the percentages of satisfied IP cameras per AoI are presented for different scenarios. Only one influential factor was changed per each scenario varying from  $1x < 2x < \dots < 7x$ , where  $x$  is any of the aforementioned influential factors and the order follows the AoI's ID, while the values of the rest factors remain the same. The results reveal that as the IP cameras' average distance from the MEC server increases, their communication channel conditions deteriorate, thus a smaller number of IP cameras meets its QoS prerequisites. Also, as the computing capability of the MEC server per AoI becomes stronger, the MEC server can more efficiently serve the cameras' computing demands, thus a greater number of them fulfills its QoS prerequisites. As the cameras' time and energy constraints become stricter, a smaller number of IP cameras gets satisfied, and as the number of cameras per AoI increases, the latter becomes congested in terms of the communication and computing aspects,

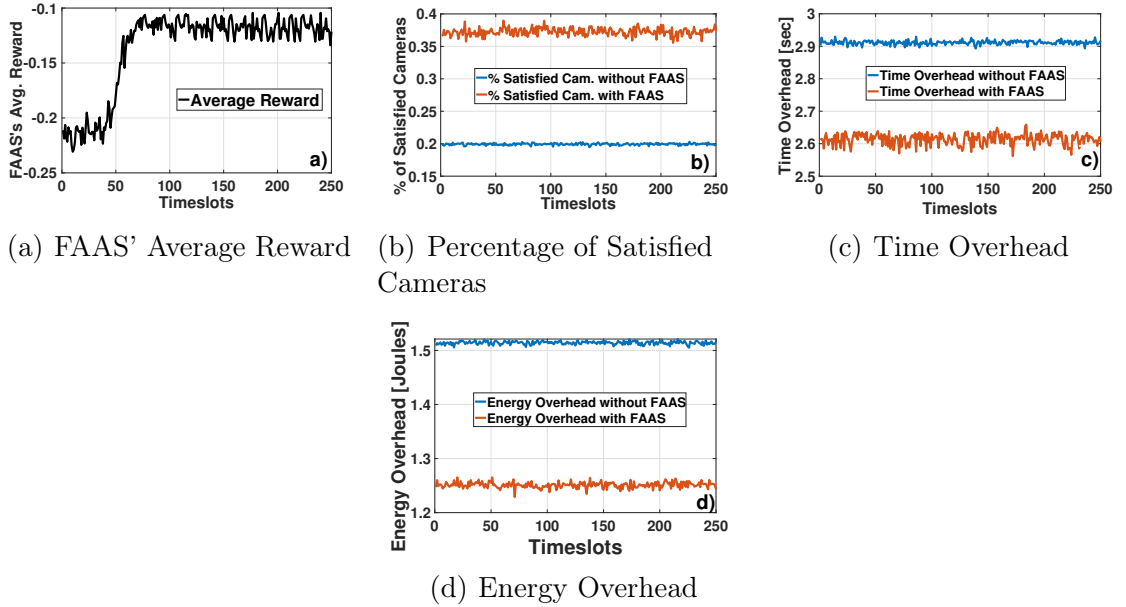


Figure 3.8: FAAS as an RL-agent: Performance Evaluation

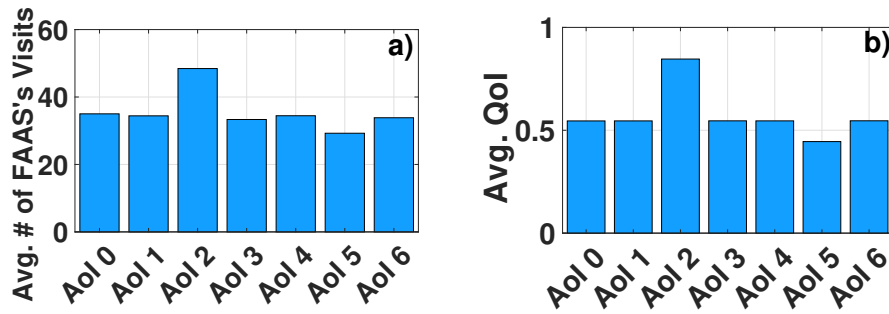


Figure 3.9: Avg. Number of FAAS's visits per AoI and AoI's QoI

thus a smaller number of cameras meets its QoS demands. Finally, it is observed that AoI 4 has superior performance in all scenarios as the FAAS resides at that AoI.

Figure 3.8.a presents the FAAS's average reward versus the timeslots. After almost 50 timeslots, the FAAS learns its environment and then it can choose the path that provides the maximum reward (Equation 3.26). In Figure 3.8.b-c, it is shown

that the percentage of satisfied cameras is significantly higher and the corresponding IP cameras' time and energy overheads are lower as the time evolves, when the FAAS visits those areas, thus showing the great benefits of adopting the FAAS in the overall considered architecture.

Figure 3.9.a, b depict the average number of FAAS's visits and the average Quality of Information per AoI in a time frame of 250 timeslots. It is observed that if an AoI has high QoI, the SARSA algorithm will efficiently consider the FAAS's perceived reward and enable the FAAS to visit more often the critical AoIs, i.e., the ones having high value of QoI.

### Scalability Evaluation

Here, we provide a scalability analysis for increasing number of cameras and available strategies to show the performance of the proposed framework. Figures 3.10a & 3.10b show the time and energy overhead and the percentage of satisfied IP cameras for increasing number of cameras per AoI. The results reveal that as the number of cameras increases, the AoIs become more congested in terms of their communication and computing environment, thus the IP cameras' time and energy overhead increases, while the percentage of the cameras that meet their QoS prerequisites decreases. Additionally, as the number of data offloading strategies increases (Figure 3.10c & 3.10d), the IP cameras have greater flexibility of choices, thus the IP cameras' time and energy overhead decreases and the corresponding percentage of satisfied cameras increases. The latter phenomenon is observed as the cameras can more accurately select the amount of data that they offload, thus better exploit the system's resources.

### Comparative Evaluation

Finally, comparative scenarios are presented to confirm the benefits of our proposed approach. The comparative scenarios are broken into two sets, examining: (i) the

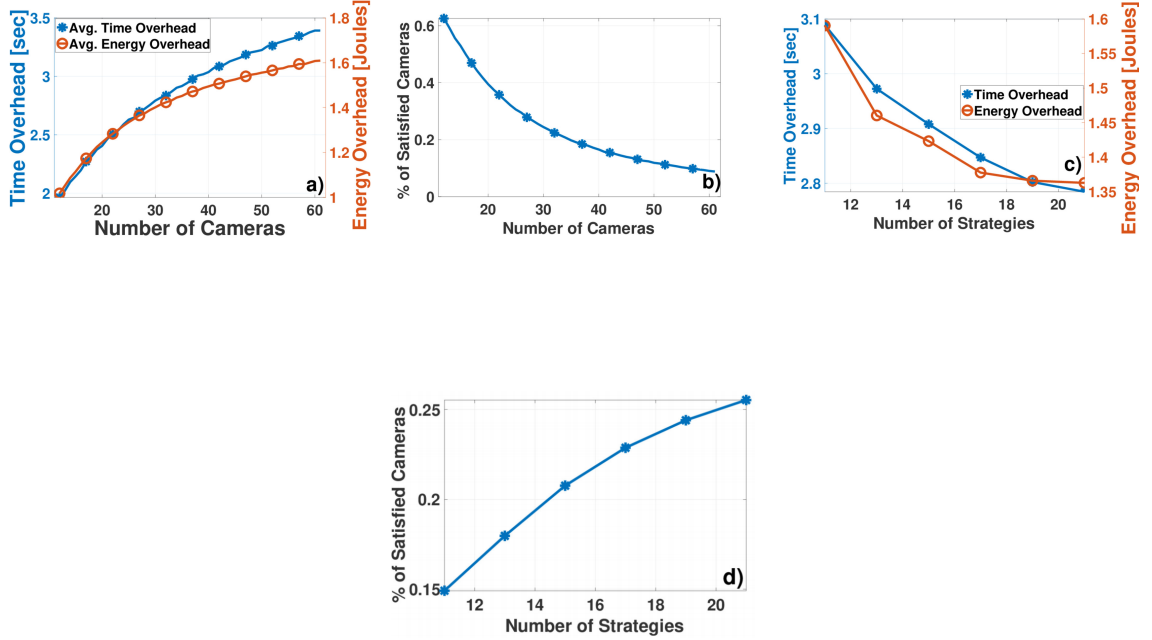


Figure 3.10: Scalability Analysis - Number of Cameras/Strategies

Satisfaction Equilibrium’s benefits, and (ii) the benefits of the adoption of Reinforcement Learning. Regarding the first set of comparative scenarios, five different approaches are presented: i) minimizing the energy (MEO) and ii) the time overhead (MTO), iii) determining the Nash Equilibrium (NE), vi) offloading the entirety (OE), and v) random amount of the data to the MEC server. As shown, the novel concept of SE resulted in the highest percent of satisfied cameras (Figure 3.11.a). In Figure 3.11.b, different scenarios of FAAS’s navigation among the AoI are presented. In the examined scenarios, the FAAS visits the area: i) closest to the current area, ii) with the largest average energy constraint, iii) sequentially, vi) maximizing its reward, v) randomly, and vi) with the largest average time constraint. The results reveal that the SARSA algorithm produced an average FAAS’s reward closer to zero



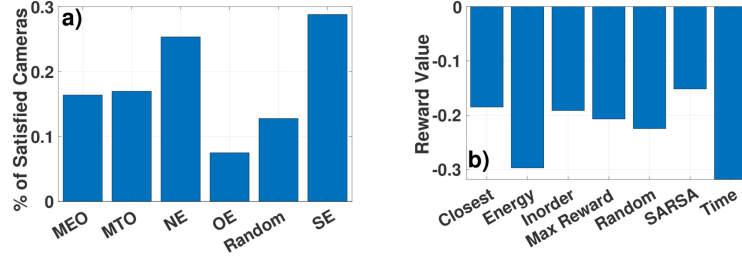


Figure 3.11: Percentage of Satisfied Cameras and FAAS's Avg. Reward with respect to different offloading and FAAS's policy approaches

compared to the other scenarios, thus indicating a better FAAS's path in terms of collecting valuable information from the surveillance system. Therefore, we conclude that the proposed approach demonstrates superior performance among all the scenarios, achieving both the highest percentage of satisfied cameras and the largest FAAS's reward.

### 3.4.5 Summary

The problem of the IP cameras' data offloading in a surveillance system consisting of AoIs and assisted by MEC servers and a FAAS, in order the IP cameras to fulfill their energy and time QoS prerequisites, is studied. Our introduced decision-making framework lead the IP cameras to converge to an SE or a GSE point based only on local information. Furthermore, the FAAS acts as an additional computing resource, and its sequential decision-making problem (i.e., movement among the AoIs) is addressed via the SARSA algorithm, an RL technique that demonstrates good results for real-world problems. The SARSA algorithm determines an optimal policy that optimizes the FAAS's long-term objective, which is constructed by several factors (i.e., QoI, AoI's performance, energy consumption).

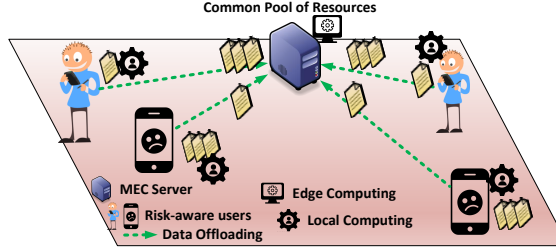


Figure 3.12: Prospect-theoretic Data Offloading in MEC

### 3.5 Cognitive Data Offloading in Mobile Edge Computing for Internet of Things

Data offloading to Mobile Edge Computing (MEC) servers is an attractive choice for resource-constrained computing systems such as Internet of Things (IoT) devices, towards reducing their computational effort. In this research work, we investigate the potential of partial data offloading to MEC servers, under the perspective of users' cognitive IoT devices presenting loss averse and gain seeking behavior (Chapter 3.5.2). Due to the sharing nature of the access environment and the MEC server's computational characteristics, we treat the MEC server option as a CPR (Chapter 3.5.2) with uncertain payoff returned to the users, while the local computation capability is treated as a safe option for each user. Following the properties of Prospect Theory, users' prospect-theoretic utilities are formulated exploiting the local computing and offloading overhead options under probabilistic uncertainty. Such modeling allows for the infusion of human awareness, inherent cognitive biases, and behavioral characteristics into the devices' operation, their data offloading decisions, and the edge computing environment that the devices are interacting with. Accordingly, each user's optimal offloaded data to the MEC server is obtained as the outcome of a non-cooperative game, with users attempting to maximize their own utilities. The existence and uniqueness of a Pure Nash Equilibrium (PNE) are proven under the probabilistic nature of the respective payoff functions, while a distributed algorithm

that convergences to the PNE are designed. Numerical results are provided that demonstrate the operation and superiority of the proposed framework under different IoT scenarios and behaviors, considering both homogeneous and heterogeneous users.

### 3.5.1 System Model

#### Computation and Communication Model

: A set of  $\aleph = \{1, \dots, i, \dots, N\}$  collocated devices is considered, where each device  $i \in \aleph$  has a computational intensive task  $T_i$  to be completed. Furthermore, we consider the uplink of a wireless network, consisting of a base station (BS) acting as a MEC server, with an upper-bounded computation capability for task execution. We consider a quasi-static scenario, where the set of devices remains unchanged during a computation offloading period. Data partitioned oriented applications are considered where each device  $i \in \aleph$  has a computation task  $T_i = (I_i, C_i)$ , where  $I_i$  and  $C_i$  denote the computation input bits and the total number of CPU cycles required to accomplish the computation task  $T_i$ , respectively. We consider  $C_i = \lambda_i * I_i$ , where the parameter  $\lambda_i$  ( $\lambda_i > 0$ ) expresses the computational complexity of the task requested by the device  $i, i \in \aleph$  and its value depends on the nature of the application, e.g., a higher  $\lambda_i$  expresses a more computation-intensive task. We assume that each computation task  $T_i$ , can be arbitrarily partitioned into subsets of any size, so each device can offload an amount of data  $b_i \in [0, I_i]$  to the MEC server and keep the rest for local computing.

We have  $b_i = 0$ , if user  $i \in \aleph$  decides to compute its whole task locally. We consider a typical interference limited communication environment, where the MEC server is the receiver of the users' transmitted data and each user experiences the interference imposed by the transmissions of the rest of the users in the examined MEC IoT environment. Given the decision profile  $\mathbf{b} = [b_1, b_2, \dots, b_N]$  of all users,

### Chapter 3. Mobile Edge Computing

the uplink data rate for the computation offloading of device  $i$  is [223]:

$$R_i = W * \log_2(1 + \frac{P_i * G_i}{\sigma^2 + \sum_{j=1, b_j \neq 0, j \neq i}^N P_j * G_j}) \quad (3.27)$$

$P_i$  the user's  $i$  transmission power,  $G_i$  the channel gain between the device  $i$  and the BS, and  $\sigma^2$  is the background noise. An overview of the overall prospect-theoretic data offloading in a mobile edge computing cognitive-enabled IoT environment is depicted in Figure 3.12.

1) *Offloading Overhead*: A user  $i$  offloads  $b_i \in [0, I_i]$  amount of data to the MEC server, where the latter executes this part of the task  $T_i$  on behalf of the user. The user  $i$  has a total offloading overhead consisting of the following terms: a) the energy consumption to transmit the data  $b_i$ , b) the transmission time and c) the execution time of the computation task at the MEC server. The energy consumption overhead  $O_i^{f,e}$  is determined by the consumed energy during the transmission of the data  $b_i$  to the MEC server as follows:  $O_i^{f,e} = \frac{b_i * P_i}{R_i}$ . The transmission time overhead  $O_i^{f,tr}$  is given as:  $O_i^{f,tr} = \frac{b_i}{R_i}$ . Similarly, the execution time of the offloaded data  $b_i$  depends on the computing resources (rate of return)  $F_i^f$  that the MEC server devotes to the computation task of user  $i$ , as follows:  $O_i^{f,t} = \frac{\lambda_i * b_i}{F_i^f}$ . More details about  $F_i^f$  are provided in Section 3.5.2.

Therefore, the total offloading overhead for user  $i, i \in \aleph$  to offload  $b_i$  data can be obtained as follows:

$$O_i^f(\mathbf{b}) = w_i^e * O_i^{f,e} + w_i^t * (O_i^{f,tr} + O_i^{f,t}) \quad (3.28)$$

where  $w_i^t, w_i^e \in [0, 1], w_i^t + w_i^e = 1$ , denote the weights of the time delay and energy consumption overheads, respectively, that can be tuned by each user according to different priorities and considerations, e.g., low battery consideration ( $w_i^e > w_i^t$ ) or delay sensitive application ( $w_i^t > w_i^e$ ). It is noted that the normalization of the energy consumption and the time delay overhead is appropriately taken into account in the weights  $w_i^e$  and  $w_i^t$ , so as both contributions to be treated fairly in terms of

### Chapter 3. Mobile Edge Computing

their order of magnitude and impact.

2) *Local Computing Overhead*: A user  $i$  executes  $(I_i - b_i)$  amount of data of its computation task  $T_i$  locally on its device. In this case, the user  $i$  has a total local computing overhead consisting of the following terms: a) the local execution time overhead and b) the local energy consumption overhead. The local execution time overhead is given as:  $O_i^{l,t} = \frac{\lambda_i * (I_i - b_i)}{F_i^l}$ , where  $\lambda_i * (I_i - b_i)$  is the number of cycles required for the local computation, and  $F_i^l$  denotes the local computation capability (CPU cycles per second) of user  $i$ . Similarly, the local energy consumption overhead of the user  $i$ , is given as:  $O_i^{l,e} = f_i * \lambda_i * (I_i - b_i)$ , where  $f_i \in \mathbb{R}^+$  denotes the consumed energy per CPU cycle. Therefore, the total local computing overhead of the user  $i$  is given as follows:

$$O_i^l(b_i) = w_i^t * O_i^{l,t} + w_i^e * O_i^{l,e} \quad (3.29)$$

Taking into account that a user may offload part of its computation task to the MEC server, based on Equation 3.28 and Equation 3.29 its total experienced overhead is:

$$O_i(\mathbf{b}) = O_i^l(b_i) + O_i^f(\mathbf{b}) \quad (3.30)$$

It is highlighted that in Equation 3.30 we consider the total overhead that a device experiences by executing part of its computation tasks locally and at the MEC server, as if the two parts are not executed in parallel. If the two parts of the computation task were executed in parallel, we could consider the largest term instead, i.e.,  $\max(O_i^{f,tr} + O_i^{f,t}, O_i^{l,t})$ , as the time delay overhead. Thus, Equation 3.30 could be written as  $O_i(\mathbf{b}) = w_i^t * \max(O_i^{f,tr} + O_i^{f,t}, O_i^{l,t}) + w_i^e * (O_i^{f,e} + O_i^{l,e})$  and would not affect the structure of the rest analysis, which would remain valid.

### Devices' Actual Utility

In this section, the users' actual utilities expressing their satisfaction from executing part of their computation task at the MEC server (CPR) and the rest locally at

### Chapter 3. Mobile Edge Computing

the device, are formulated. The exploitation of the MEC server's computation capabilities via offloading part of the user's computation task to the server provides a corresponding satisfaction to the user, which depends on the server's workload. This satisfaction is captured by the rate of return function  $F_i^f$ , which is personalized based on each device's task's computational complexity  $\lambda_i$ , and decreases as the total computation offloading by all devices increases due to the upper-bounded computation capacity of the MEC server. Specifically, the MEC server provides its computation capabilities to the users in a fair and proportional manner. Thus, the devices whose computation tasks are characterized by higher computational complexity, i.e.,  $\lambda_i$ , experience an improved rate of return as the MEC server managed to fulfill their demanding computation tasks. The rate of return function for each device  $i, i \in \mathbb{N}$  is formulated as:

$$F_i^f(b_T) = \frac{\lambda_i}{\sum_{j=1, b_j \neq 0}^N \lambda_j} * d(b_T) \quad (3.31)$$

where  $b_T = \sum_{j=1}^N b_j$  is the total amount of offloaded data by all the devices to the MEC server and  $d(b_T)$  is the production function of the MEC server expressing its computing performance with respect to the total data offloading. The production function is formulated as follows:

$$d(b_T) = \begin{cases} (1 - \frac{b_T}{b_{th}}) * F_{MEC} & \text{if } b_T \leq b_{th} \\ 0 & \text{otherwise} \end{cases} \quad (3.32)$$

where  $F_{MEC}$  [CPU Cycles/sec] denotes the MEC server's upper bound computation capability, which is shared among the different offloaded tasks. The parameter  $b_{th}$  denotes the MEC server's received bytes threshold value, where if  $b_T \geq b_{th}$  the MEC server is considered unable to execute the offloaded tasks into a specific duration of time, thus it "fails". This concept is well-known in the literature as the "Tragedy of the Commons" [131]. As a result, in this case, it is more beneficial for the device  $i \in \mathbb{N}$  to execute its whole task  $T_i$  locally. The consideration of including the MEC server's received bytes threshold value in our analysis captures the operation of a realistic

MEC system, where if the MEC server was overwhelmed with data to process, then it would become over-exploited concluding in increased delays. In that case, the computing services offered by the MEC server to the devices become unsatisfactory and of no value to them.

**Proposition 1** *The production function  $d(b_T)$ , and each device's  $i \in \mathbb{N}$  rate of return function  $F_i^f(b_T)$ , are strictly decreasing with respect to the total offloaded data  $b_T$ .*

The above proposition holds true in our environment, given that if the MEC server becomes overloaded, the device's choice of offloading part of its task to the MEC server becomes less beneficial as the user suffers the burden of long computation time delays stemming from the over-exploited MEC server. In the following analysis, without loss of generality and for simplicity in the presentation, we consider  $w_i^t = w_i^e = 1/2, \forall i \in \mathbb{N}$ , thus, each device has equal sensitivity to the time delay and the energy consumption overhead. Each device is associated with an actual utility function formulated as a linear combination of the overhead experienced by executing a part of its computation task to the MEC server and the rest part locally. Thus, the device's actual utility can be formulated via combining Equation 3.28 and Equation 3.29 as follows:

$$z_i(\mathbf{b}) = b_i * \left( \frac{\lambda_i}{F_i^f} + \frac{P_i + 1}{R_i} \right) + \lambda_i * (I_i - b_i) * \left( \frac{1}{F_i^l} + f_i \right) \quad (3.33)$$

### 3.5.2 The Prospect of Data Offloading

#### Partial Offloading under Prospect Theory

In real mobile applications, users do not always adopt risk-neutral behavior, instead, they tend to demonstrate different actions under losses or gains with respect to their actual utility. Towards capturing the device-centric risk-based decision-making in the MEC environment, the Prospect Theory is adopted (Chapter 3.5.2). Following

this behavioral model, individuals make decisions under risk and uncertainty of the associated payoff of their choices, which is estimated with some probability.

Therefore, the users' actual utility as expressed in Equation 3.33, is evaluated with respect to a reference point (reference dependence property) [130]. This reference point is considered as the zero point (i.e., ground truth) of the users' actual utility. Given the reference point and users' offloaded data, they determine their prospect-theoretic probabilistic payoff. In our work, we consider as the reference point of each user the corresponding experienced overhead, if the whole task was locally executed.

Users' prospect-theoretic utility function is a concave function with respect to the user's actual perceived utility above the reference point, a convex function below it, and has a greater slope in losses compared to the gains (loss aversion property), as presented in Figure 3.13. This formulation is well-aligned with the observation that the users weigh more the losses compared to the gains of the same amount of overhead in terms of dissatisfaction and satisfaction, respectively (diminishing sensitivity property).

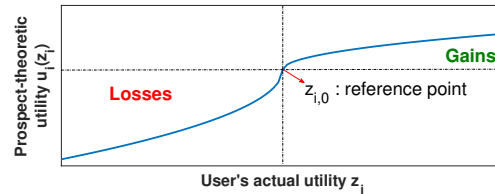


Figure 3.13: Prospect-theoretic Data Offloading in MEC

### MEC: A Common Pool of Resources

In the MEC environment under consideration, the MEC server is considered as a Common Pool of Resources (CPR), since it is: a) non-excludable, in the sense that all the users have access to arbitrarily offload their computation tasks to the server, and b) rivalrous and subtractable for the users, as the reservation of computation capabilities by one user from the MEC server, reduces the ability to reserve compu-



tation cycles by another [131] (Chapter 3.5.2). Each user's goal is to determine in an autonomous manner the offloaded data  $b_i$  to the CPR with some uncertainty in the expected obtained outcome while maintaining its remaining amount of data  $(I_i - b_i)$  locally to be executed at the device, which is a "safe" computation resource in terms of a priori knowing the total local overhead, so as to minimize its overall perceived overhead. The probability of failure of the MEC server (CPR) is denoted by  $p(b_T)$ .

**Proposition 2** *The MEC server (CPR) is characterized by the following properties.*

1. *The probability of failure  $p(b_T)$  is strictly increasing, convex and twice continuously differentiable with respect to  $b_T \in [0, b_{th})$ , with  $p(b_T) = 1, \forall b_T \geq b_{th}$ .*
2. *User's  $i$  strategy set of offloading an amount of data to the MEC server is  $S_i = [0, \min(I_i, b_{th})]$ ,  $\forall i \in \mathbb{N}$ .*

It is noted that the corresponding probability of failure  $p(b_T)$  being strictly increasing with respect to  $b_T$ , allows to properly capture the reliability characteristics of the MEC server. Some examples of the MEC server's probability of failure are the logarithmic, the linear, or the exponential function with respect to  $b_T$ . The choice of an appropriate functional form could be based on various operational factors and characteristics (e.g., the MEC server's robustness to failure, non-linear server's behavior to traffic loads and computing utilization, etc.), and assuming that satisfies the properties in Proposition 2, it does not harm the validity of our analysis.

Here, without loss of generality and for demonstration purposes only, we consider a linear probability of failure function given as follows:  $p(b_T) = \frac{b_T}{b_{th}}$ . It is highlighted that in the case of an underloaded MEC server, the probability of failure function will return small values, thus, concluding to better experience and satisfaction for the user that offloads its computation tasks to the MEC server.

### Prospect-Theoretic Utility

Based on the Prospect Theory, the prospect-theoretic utility of a user is defined as follows [224]:

$$u_i(z_i) = \begin{cases} (z_{i,0} - z_i)^{\alpha_i} & \text{if } z_i \leq z_{i,0} \\ -k_i * (z_i - z_{i,0})^{\gamma_i} & \text{if } z_i > z_{i,0} \end{cases} \quad (3.34)$$

where  $z_i$  is the user's  $i, i \in \mathbb{N}$  actual utility as defined in Equation 3.33 and  $z_{i,0}$  denotes the reference point of user's prospect-theoretic utility. Each user's reference point  $z_{i,0}$  is defined as the actual utility that it experiences by executing its whole task  $T_i$  locally at the device.

$$z_{i,0} = z_i|_{b_i=0} = \lambda_i * I_i * \left( \frac{1}{F_i^l} + f_i \right) \quad (3.35)$$

As stated earlier, we have omitted the weight  $w_i^t = w_i^e = 1/2$  for simplicity in the presentation. Each device's  $i, i \in \mathbb{N}$  parameters  $\alpha_i, \gamma_i \in (0, 1]$  express the user's sensitivity to the gains and losses of its actual utility  $z_i$ , respectively, as these were described in Chapter 3.5.2. If the MEC server (CPR) does not fail due to the over-offloading of users' data, then each user perceives an actual utility given by the Equation 3.33. In this case, the actual perceived utility (overhead) is lower than the reference point  $z_{i,0}$ , i.e.,  $z_i \leq z_{i,0}$ , as at the reference point the user  $i$  executes its whole task  $T_i$  locally. Therefore, via subtracting the actual utility  $z_i$  (Equation 3.33) from the reference point  $z_{i,0}$  (Equation 3.35) and shaping the result according to the first branch of Equation 3.34, we have  $u_i(z_i) = [b_i(\frac{\lambda_i}{F_i^l} + \lambda_i f_i - \frac{P_i+1}{R_i} - \frac{\lambda_i}{F_i^f})]^{\alpha_i}$ . On the other hand, if the MEC server becomes overloaded and fails to serve the users' offloaded computation tasks, the users' overhead is given by Equation 3.33 with  $b_i = 0$ , as the users have to execute their whole tasks locally, though they experienced the energy consumption and transmission time overhead from offloading  $b_i$  data to the MEC server, before the stage of failure is reached. Thus, user's actual utility is  $z_i = z_{i,0} + \frac{b_i}{R_i} + \frac{b_i}{R_i} P_i$  and is greater than the reference point  $z_{i,0}$ . Therefore,

### Chapter 3. Mobile Edge Computing

by subtracting the reference point from user's actual utility, the second branch of Equation 3.34 can be written as  $u_i(z_i) = -k_i(b_i \frac{P_i+1}{R_i})^{\alpha_i}$ .

Following the aforementioned argumentation, and as per Equation 2.35, we can readily rewrite the user's prospect-theoretic utility (Equation 3.34) as follows:

$$u_i(z_i) = \begin{cases} [b_i(\frac{\lambda_i}{F_i^l} + \lambda_i f_i - \frac{P_i+1}{R_i} - \frac{\lambda_i}{F_i^f(b_T)})]^{\alpha_i} & \text{if } z_i \leq z_{i,0} \\ -k_i(b_i \frac{P_i+1}{R_i})^{\alpha_i} & \text{if } z_i > z_{i,0} \end{cases} \quad (3.36)$$

Moreover, for notational convenience we define  $\bar{d}_i(b_T) \triangleq (\frac{\lambda_i}{F_i^l} + \lambda_i f_i - \frac{P_i+1}{R_i} - \frac{\lambda_i}{F_i^f(b_T)})^{\alpha_i} > 0$  assuming that the server has not failed and  $\epsilon_i \triangleq (\frac{P_i+1}{R_i})^{\alpha_i}$ , and therefore Equation 3.36 can be written as:

$$u_i(z_i) = \begin{cases} b_i^{\alpha_i} \bar{d}_i(b_T) & \text{if } z_i \leq z_{i,0} \\ -k_i \epsilon_i b_i^{\alpha_i} & \text{if } z_i > z_{i,0} \end{cases} \quad (3.37)$$

The MEC server's failure to serve the users depends on the total offloaded data by all of them. Given that the probability of the MEC server's failure is  $p(b_T)$ , the probability that the server survives and executes the offloaded computation tasks is accordingly  $(1 - p(b_T))$ . As a result, considering the probability of MEC server's failure, Equation 3.37 can be written equivalently as follows:

$$u_i(z_i) = \begin{cases} b_i^{\alpha_i} \bar{d}_i(b_T), & \text{with prob. } 1 - p(b_T) \\ -k_i \epsilon_i b_i^{\alpha_i}, & \text{with prob. } p(b_T) \end{cases} \quad (3.38)$$

Each user's expected prospect-theoretic utility based on all users' offloaded data  $\mathbf{b} = [b_1, b_2, \dots, b_N]$ , and as per Equation 2.37 is given as follows.

$$\begin{aligned} \mathbb{E}(u_i) &= b_i^{\alpha_i} \bar{d}_i(b_T)(1 - p(b_T)) - (k_i \epsilon_i b_i^{\alpha_i})p(b_T) \\ &= b_i^{\alpha_i} [\bar{d}_i(b_T)(1 - p(b_T)) - k_i \epsilon_i p(b_T)] \\ &\triangleq b_i^{\alpha_i} g_i(b_T) \end{aligned} \quad (3.39)$$

where  $g_i(b_T) = \bar{d}_i(b_T)(1 - p(b_T)) - k_i \epsilon_i p(b_T)$  is the effective rate of return of the MEC server for the user  $i, i \in \mathbb{N}$ .

### 3.5.3 Optimizing Devices' Overhead

#### Problem Formulation

The goal of each device is to minimize its perceived overhead from its computation task's execution via sophisticatedly and selfishly offloading part of the task to the MEC server. This problem can be formulated as a maximization problem of each user's prospect-theoretic utility function, as follows:

$$\max_{b_i \in S_i} \mathbb{E}(u_i) = b_i^{\alpha_i} g_i(b_T) \quad (3.40)$$

In the modeling considered in this work, the latency and energy factors have been directly considered as part of the corresponding overheads, computed for both offloading and local computing cases (i.e., Equation 2 and Equation 4), while their combined optimization is treated and achieved through the solution of the optimization problem in Equation 3.40. Following the vision of ultra-reliable low latency communications and respecting the energy limitations of the IoT devices, as part of the emerging Tactile Internet [225], this problem could be further extended by considering hard constraints on the required latency of the computation task and the energy availability of the devices. Accordingly, these constraints are expected to reduce the users' strategy space and the corresponding feasible solution space.

The above maximization problem can be confronted as a non-cooperative game (Chapter 2.1) among the users who act as players making the optimal decisions about themselves in a selfish and distributed manner. Let  $G = [\mathbb{N}, \{S_i\}, \{\mathbb{E}(u_i)\}]$  denote the non-cooperative game among the  $N$  users, where each user's strategy space is  $S_i = [0, \min(I_i, b_{th})]$  and its payoff is its expected prospect-theoretic utility function  $\mathbb{E}(u_i)$ . Towards solving the non-cooperative game, the concept of Nash equilibrium is adopted. The Nash equilibrium (NE) of the non-cooperative game  $G$  is the vector of users' amount of offloaded data  $\mathbf{b}^* = [b_1^*, \dots, b_i^*, \dots, b_N^*]$ , where no user has the incentive to change its own strategy (i.e., amount of offloaded data) given the strategies of the rest of the users (Chapter 2.1 - Definition 1). Let  $\mathbf{b}_{-i}^* =$

$[b_1^*, \dots, b_{i-1}^*, b_{i+1}^*, \dots, b_N^*]$  denote the vector of offloaded data of all users except user  $i$  at the NE point (Chapter 2.1).

It is noted that a Prospect Theory-based game-theoretic approach is adopted to treat the aforementioned problem, instead of other approaches, due to its distributed and computationally efficient nature, while properly capturing the users' behavioral characteristics. The sequential best response dynamics mechanism is adopted to determine the game's PNE, which as also confirmed later in the empirical evaluation, converges fast to it and in a scalable manner, due to its best response nature, in contrast for example to other learning-based techniques, which need large exploration and exploitation time to determine a stable solution. In addition, in several cases, a large amount of reliable data and extensive time would be required for the proper training of a supervised learning-based approach, for instance. However, a machine learning (ML) based approach (and in particular reinforcement learning) could further complement the proposed framework and support the applicability of the best response determining process, in terms of treating potential incompleteness of the available information under uncertain environments, such as the communication and computing environment.

### Existence and Uniqueness of PNE

Let us denote the best response strategy  $B_i(\mathbf{b}_{-i}) : S_{-i} \rightrightarrows S_i$  of user  $i$ , as follows:

$$B_i(\mathbf{b}_{-i}) = \arg \max_{b_i \in S_i} \mathbb{E}(u_i(b_i, \mathbf{b}_{-i})), \mathbf{b}_{-i} \in S_{-i} \quad (3.41)$$

where  $\mathbf{b}_{-i} = [b_1, \dots, b_{i-1}, b_{i+1}, \dots, b_N]$  is the data vector of all users excluding user  $i$ , and  $S_{-i} = S_1 \times \dots \times S_{i-1} \times S_{i+1} \times \dots \times S_N$  the corresponding mixed strategy.

**Theorem 6** *For each user  $i, i \in \mathbb{N}$ , its best response strategy exists and it is single-valued, such that  $b_i^* = B_i(\mathbf{b}_{-i}) = \arg \max_{b_i \in S_i} \mathbb{E}(u_i(b_i, \mathbf{b}_{-i}))$ .*

### Chapter 3. Mobile Edge Computing

We adopt the notation  $b_{-i} = \sum_{j=1, j \neq i}^N b_j$  to depict the total offloaded data of all users except user  $i, i \in \aleph$ . In order to prove the above theorem, we first present Berge's Theorem [226] and then we prove the following Lemmas 1 - 3.

**Theorem 7** *Let  $\Theta$  and  $X$  be two metric spaces, and  $\Gamma : \Theta \rightrightarrows X$  a compact valued correspondence. Let the function  $\Phi : X \times \Theta \rightarrow \mathbb{R}$  be jointly continuous in  $X$  and  $\Theta$ . We define:*

1.  $\sigma(\theta) := \arg \max_{x \in \Gamma(\theta)} \Phi(x, \theta)$
2.  $\Phi^*(\theta) := \max_{x \in \Gamma(\theta)} \Phi(x, \theta), \forall \theta \in \Theta$

*If  $\Gamma$  is continuous at  $\theta \in \Theta$ , then*

1.  $\sigma : \Theta \rightrightarrows X$  is compact-valued, upper hemicontinuous and closed at  $\theta$
2.  $\Phi^* : \Theta \rightarrow \mathbb{R}$  is continuous at  $\theta$

**Lemma 1** *For each user  $i, i \in \aleph$  the following hold true:*

1.  $b_i^* = 0$  if and only if  $b_{-i} \geq \bar{b}_i$ , where a value  $\bar{b}_i \in [0, b_{th}]$  exists.
2.  $b_i^* > 0$  and  $b_i^* + b_{-i} < \bar{b}_i$ , if  $b_{-i} < \bar{b}_i$  and there exists an interval  $A_i \subset [0, \bar{b}_i)$  such that  $g_i(\bar{b}_i) = 0$ .

*Proof:* Initially, we clarify that the user's  $i, i \in \aleph$  best response strategy  $b_i^*$  can either be zero, i.e.,  $B_i(\mathbf{b}_{-i}) = b_i^* = 0$  or a positive value, i.e.,  $B_i(\mathbf{b}_{-i}) = b_i^* \in S_i$ , and the best response value can never be equal to  $b_{th}$ , i.e.,  $b_i^* = b_{th}$  as in this case  $p(b_{th}) = 1$ , thus the MEC server (CPR) fails and the user's expected prospect-theoretic utility is negative. The first order derivative of the effective rate of return of the MEC server for the user  $i, i \in \aleph$  is given as follows:

$$\frac{\partial g_i(b_T)}{\partial b_T} = \frac{\partial \bar{d}_i(b_T)}{\partial b_T} (1 - p(b_T)) - \frac{\partial p(b_T)}{\partial b_T} (\bar{d}_i(b_T) + k_i \epsilon_i) \quad (3.42)$$

### Chapter 3. Mobile Edge Computing

It is obvious that  $\frac{\partial p(b_T)}{\partial b_T} > 0$ , as the probability of failure, i.e.,  $p(b_T)$  is strictly increasing with respect to  $b_T$ . Also  $(1 - p(b_T)) > 0$ . Moreover,  $\frac{\partial \bar{d}_i(b_T)}{\partial b_T} < 0$ , since the  $\bar{d}_i(b_T)$  is strictly decreasing with respect to  $b_T$  based on Proposition 1. Thus,  $g_i(b_T)$  is strictly decreasing with respect to  $b_T$ .

CASE A: If  $g_i(0) \leq 0$ , then  $g_i(b_i) \leq 0, \forall b_i \in S_i$  and  $\mathbb{E}(u_i) \leq 0$ . So, in this case the only best response for the user  $i$  is the zero value, i.e.,  $B_i(\mathbf{b}_{-i}) = b_i^* = 0$ . As a result,  $\bar{b}_i = 0$  and the interval  $A_i$  is not defined.

CASE B: If  $g_i(0) > 0$ , then since  $g_i(b_{th}) = -k_i \epsilon_i < 0$ , we know that  $\exists \bar{b}_i \in [0, b_{th}]$  such that  $g_i(\bar{b}_i) = 0$  based on the Intermediate Value Theorem [227]. As a result, if  $b_{-i} \geq \bar{b}_i$  then  $B_i(\mathbf{b}_{-i}) = 0$ , as  $\forall b_i \neq 0$ , it holds true that  $g_i(b_i + \bar{b}_i) < 0$  due to the fact that  $g_i(\cdot)$  is strictly decreasing, thus  $\mathbb{E}(u_i(b_i, \mathbf{b}_{-i})) < 0$ . On the other hand, if  $b_{-i} < \bar{b}_i$  then  $g_i(b_i + b_{-i}) > 0, \forall b_i \in (0, \bar{b}_i - b_{-i})$ . So, in this case  $\exists b_i : g_i(b_i + b_{-i}) > 0$ , thus, the zero value cannot be the best response for the user  $i, i \in \aleph$ , if and only if  $b_{-i} \in [0, \bar{b}_i)$ . Also, because of the positive value of the expected prospect-theoretic utility at the best response, i.e.,  $\mathbb{E}(u_i(b_i^*, \mathbf{b}_{-i})) > 0$  it is true that  $b_i^* + b_{-i} \in (0, \bar{b}_i)$ , and as a result the interval  $A_i$  exists and is defined as  $A_i = (0, \bar{b}_i)$

**Lemma 2** For each mobile user  $i, i \in \aleph$ , its best response  $b_i^*$  is single-valued  $\forall b_{-i} \in S_{-i}$

*Proof:* Based on Lemma 1, case A, we have shown that the best response strategy is single-valued, i.e.,  $B_i(\mathbf{b}_{-i}) = 0$  if and only if there exists a value  $\bar{b}_i \in [0, b_{th}]$  such that  $b_{-i} \geq \bar{b}_i$ . Thus, in the following we examine the case B as presented in Lemma 1, where we have already shown that there exists at least one best response strategy, i.e.,  $B_i(\mathbf{b}_{-i}) > 0$ . Given that there exists at least one best response strategy  $B_i(\mathbf{b}_{-i})$ , it should be one of the solutions of the expected prospect-theoretic utility's first order

### Chapter 3. Mobile Edge Computing

derivative, as follows:

$$\begin{aligned} \frac{\partial \mathbb{E}(u_i)}{\partial b_i} &= [b_i^{a_i} \frac{\partial \bar{d}_i(b_T)}{\partial b_T} + a_i b_i^{a_i-1} \bar{d}_i(b_T)](1 - p(b_T)) \\ &\quad - b_i^{a_i} \bar{d}_i(b_T) \frac{\partial p(b_T)}{\partial b_T} \\ &\quad - k_i \epsilon_i [a_i b_i^{a_i-1} p(b_T) + b_i^{a_i} \frac{\partial p(b_T)}{\partial b_T}] \end{aligned} \quad (3.43)$$

It is noted that  $\frac{\partial \mathbb{E}(u_i)}{\partial b_i} = \frac{\partial \mathbb{E}(u_i)}{\partial b_T}$ , since  $b_T = b_i + b_{-i}$ . Also,  $-b_i^{a_i} \bar{d}_i(b_T) \frac{\partial p(b_T)}{\partial b_T} < 0$  and  $-k_i [a_i b_i^{a_i-1} p(b_T) + b_i^{a_i} \frac{\partial p(b_T)}{\partial b_T}] < 0$ . Thus, to determine the root of Equation 3.43, it should hold true:

$$[b_i^{a_i} \frac{\partial \bar{d}_i(b_T)}{\partial b_T} + a_i b_i^{a_i-1} \bar{d}_i(b_T)] > 0 \quad (3.44)$$

Calculating the second derivative of  $\mathbb{E}(u_i)$  we have:

$$\begin{aligned} \frac{\partial^2 \mathbb{E}(u_i)}{\partial b_i^2} &= [b_i^{a_i} \frac{\partial^2 \bar{d}_i(b_T)}{\partial b_T^2} + 2a_i b_i^{a_i-1} \frac{\partial \bar{d}_i(b_T)}{\partial b_T}](1 - p(b_T)) \\ &\quad - 2b_i^{a_i-1} [b_i \frac{\partial \bar{d}_i(b_T)}{\partial b_T} + a_i \bar{d}_i(b_T)] \frac{\partial p(b_T)}{\partial b_T} \\ &\quad - b_i^{a_i} \bar{d}_i(b_T) \frac{\partial^2 p(b_T)}{\partial b_T^2} \\ &\quad - k_i \epsilon_i [2a_i b_i^{a_i-1} \frac{\partial p(b_T)}{\partial b_T} + b_i^{a_i} \frac{\partial^2 p(b_T)}{\partial b_T^2}] \\ &\quad + a_i(a_i - 1)b_i^{a_i-2} [\bar{d}_i(b_T)(1 - p(b_T)) - k_i \epsilon_i p(b_T)] \end{aligned} \quad (3.45)$$

Specifically, due to the fact that  $b_i$  satisfies (3.44),  $\frac{\partial^2 \bar{d}_i(b_T)}{\partial b_T^2} < 0$ ,  $\frac{\partial \bar{d}_i(b_T)}{\partial b_T} < 0$ ,  $\frac{\partial p(b_T)}{\partial b_T} > 0$  and  $\frac{\partial^2 p(b_T)}{\partial b_T^2} = 0$ , it is true that  $\frac{\partial^2 \mathbb{E}(u_i)}{\partial b_i^2} < 0, \forall b_i \in (0, \bar{b}_i)$ , thus  $\mathbb{E}(u_i)$  is strictly concave. Moreover, given that  $\bar{d}_i(b_T)$  is concave decreasing, the function from the inequality (3.44), i.e.,  $b_i^{a_i} \frac{\partial \bar{d}_i(b_T)}{\partial b_T} + a_i b_i^{a_i-1} \bar{d}_i(b_T)$ , is decreasing with respect to  $b_i$ . For small values of  $b_i$ , i.e.,  $b_i \rightarrow 0$  and  $b_{-i} < \bar{b}_i$  it holds true that  $b_i^{a_i} \frac{\partial \bar{d}_i(b_T)}{\partial b_T} + a_i b_i^{a_i-1} \bar{d}_i(b_T) > 0$ . Defining  $C := \sup\{b_i \in S_i : b_i^{a_i} \frac{\partial \bar{d}_i(b_T)}{\partial b_T} + a_i b_i^{a_i-1} \bar{d}_i(b_T) > 0\}$ , inequality 3.44 holds true only in the interval  $[0, C]$ . Thus, the expected prospect-theoretic utility function has a unique maximum in  $[0, C]$ .

**Lemma 3** *The best response strategy of the user  $i, i \in \mathbb{N}$ ,  $b_i^* : S_{-i} \rightrightarrows S_i$  is continuous for  $b_{-i} \in S_{-i}$ .*



### Chapter 3. Mobile Edge Computing

*Proof:* The  $b_i^* : S_{-i} \rightrightarrows S_i$  is mapped to  $\sigma$  and the expected prospect-theoretic utility is mapped to the function  $\Phi$  (see the notation in Theorem 7). We compute  $b_i^* \in S_i$  and define the correspondence  $\Gamma : S_{-i} \rightrightarrows [0, 1]$  for any joint strategies of users other than  $i$ . Therefore,  $\Gamma$  is compact valued, and both upper and lower hemicontinuous. Hence,  $b_i^*$  is upper hemicontinuous from Theorem 7 and as it is single-valued (Lemma 2), is continuous.

Based on Theorem 7 and Lemmas 1 - 3, we proved that for each user  $i$ , its best-response strategy  $B_i(\mathbf{b}_{-i})$  exists and is single-valued and continuous. Thus, we proved Theorem 1.

**Theorem 8** *A pure Nash equilibrium  $\mathbf{b}^* = [b_1^*, \dots, b_N^*]$  of the non-cooperative game  $G = [\aleph, \{S_i\}, \{\mathbb{E}(u_i)\}]$  exists.*

*Proof:* The strategy set  $S_i, \forall i \in \aleph$  is a convex compact subset of the Euclidean space and so is the joint strategy space,  $S = S_1 \times \dots \times S_N \subset \mathbb{R}^{|\aleph|}$ . By defining a mapping  $T : S \rightarrow S$  such that  $T(b_1, \dots, b_N) = (b_1^*, \dots, b_N^*)$ , from Lemma 2,  $T$  is single-valued and from Lemma 3 is continuous. Brouwer's fixed point theorem guarantees the existence of a strategy profile  $s = \{b_i^*\}_{i \in \aleph} \in S$  that is invariant under the best response mapping and therefore is a PNE of  $G$  [226].

The best response  $b_i^*$  of user  $i, i \in \aleph$  satisfies the equation  $\frac{\partial \mathbb{E}(u_i)}{\partial b_i} |_{b_i^*} = 0$ . Based on the latter condition and Equation 3.39, we define the function  $h_i(b_T) = \frac{-a_i g_i(b_T)}{\frac{\partial g_i(b_T)}{\partial b_T}} = b_i$  which satisfies  $h_i(b_i^* + b_{-i}) = b_i^*$ , when  $b_i^* > 0$ .

**Lemma 4** *The function  $h_i(b_T)$  is strictly decreasing with respect to  $b_T$ ,  $b_T \in A_i$ , where  $A_i$  is as defined in Lemma 1.*

*Proof:* We have that  $\bar{d}_i(b_T)$  is decreasing, and that:

$$\frac{1}{a_i} \frac{\partial h_i(b_T)}{\partial b_T} = - \frac{(\frac{\partial g_i(b_T)}{\partial b_T})^2 - g_i(b_T) \frac{\partial^2 g_i(b_T)}{\partial b_T^2}}{(\frac{\partial g_i(b_T)}{\partial b_T})^2} \quad (3.46)$$

### Chapter 3. Mobile Edge Computing

The numerator is equal to  $[\frac{\partial \bar{d}_i(b_T)}{\partial b_T}(1 - p(b_T))]^2 + [\frac{\partial p(b_T)}{\partial b_T}(\bar{d}_i(b_T) + k_i \epsilon_i)]^2 - g_i(b_T)\rho - 2\bar{d}_i(b_T)\frac{\partial p(b_T)}{\partial b_T}k_i \epsilon_i$ , which is positive, and  $\rho = \frac{\partial^2 \bar{d}_i(b_T)}{\partial b_T^2}(1 - p(b_T)) - \bar{d}_i(b_T)\frac{\partial^2 p(b_T)}{\partial b_T^2} - k_i \epsilon_i \frac{\partial^2 p(b_T)}{\partial b_T^2} \leq 0$ . Thus,  $h_i(b_T)$  is strictly decreasing.

**Theorem 9** *The pure Nash Equilibrium of  $G = [\aleph, \{S_i\}, \{\mathbb{E}(u_i)\}]$  is unique.*

*Proof:* We use the notation  $b_T^*$  to denote the total amount of offloaded data at the PNE point of the game  $G$ . The proof of Theorem 9 is based on the reduction to absurdity. Let us suppose that we have two distinct PNE points,  $b_{T(1)}^*, b_{T(2)}^*$ . Without loss of generality we assume that  $b_{T(2)}^* > b_{T(1)}^*$ . We define the set  $Sup \triangleq \{i \in \aleph : b_T^* < \bar{b}_i\}$ , thus it includes every user with non zero amount of offloading data to the MEC server. Thus, we have  $Sup_2 \subseteq Sup_1$ . We have that  $\sum_{j \in Sup_1} h_j(b_{T(1)}^*) = b_{T(1)}^*$ ,  $\sum_{j \in Sup_2} h_j(b_{T(2)}^*) = b_{T(2)}^*$ . So,  $\sum_{j \in Sup_2} h_j(b_{T(1)}^*) + \sum_{j \in Sup_1 \setminus Sup_2} h_j(b_{T(1)}^*) = b_{T(1)}^* \Rightarrow \sum_{j \in Sup_2} h_j(b_{T(1)}^*) \leq b_{T(1)}^* < b_{T(2)}^* = \sum_{j \in Sup_2} h_j(b_{T(2)}^*)$ . However,  $h_i(b_T)$  is decreasing, so  $h_i(b_{T(1)}^*) > h_i(b_{T(2)}^*), \forall j \in Sup_2$ , which is contradiction. So,  $b_{T(1)}^* = b_{T(2)}^*$ .

#### Algorithm - Convergence to PNE

A direct consequence of Lemma 4, is that the best response strategy of a user is decreasing in the total amount of offloading data. As a result,  $G$  belongs to the class of *best-response potential games*, thus, the sequential best response dynamics converge to the PNE of the game  $G$  [228]. From Theorem 9 and Lemma 2, we conclude that each user's  $i$  best response is unique. Specifically, its best response is zero if and only if the total offloaded data of the rest users is greater than its threshold value, i.e.,  $b_{-i} \geq \bar{b}_i$ . Otherwise, its best response must be the root of the first order derivative of the expected prospect-theoretic utility, thus  $\frac{\partial \mathbb{E}(u_i)}{\partial b_i} = 0$ .

Each user  $i$  in order to compute its best response, first receives the total amount of offloaded data of the rest users, i.e.,  $b_{-i}$  and then determines if it is zero, i.e., whether  $b_{-i} \geq \bar{b}_i$  holds true. The later is satisfied if and only if  $g_i(b_{-i}) \leq 0$ . If the user  $i$  finds that  $b_{-i} < \bar{b}_i$ , then the best response  $b_i^*$  exists and is single-valued

### Chapter 3. Mobile Edge Computing

(Theorem 1). Specifically, due to the existence of the unique root of  $\frac{\partial \mathbb{E}(u_i)}{\partial b_i} = 0$ , and regarding that  $\frac{\partial^2 \mathbb{E}(u_i)}{\partial b_i^2} < 0$ , thus  $\frac{\partial \mathbb{E}(u_i)}{\partial b_i}$  is strictly decreasing with respect to  $b_i$ , then the unique root  $r_i^*$  can be found via binary search into  $[0, b_{th}]$  with an approximation  $\epsilon$ , such that  $\epsilon \rightarrow 0$ , and finally the best response to be  $b_i^* = \min(I_i, r_i^*)$ .

The complexity of the binary search is  $\mathcal{O}(\log_2 b_{th})$  [229]. In each iteration,  $N$  users execute Algorithm 1 and given that the rest operations involve arithmetical calculations and  $Ite$  iterations are needed for convergence to the PNE, the complexity of the distributed algorithm is  $\mathcal{O}(N * Ite * \log_2 b_{th})$ . It is noted that the iterations scale very well with respect to the increasing number of users.

---

#### **Algorithm 2 :** Distributed Algorithm for Convergence to PNE

---

**Input:** Set of Users  $\aleph = \{1, 2, \dots, i, \dots, N\}$

**Output:** Vector at PNE  $\mathbf{b}^* = [b_1^*, \dots, b_i^*, \dots, b_N^*]$

$Ite = 0, b_i = randi(0, \min(b_{th}, I_i)), \forall i \in \aleph$

**while** PNE not reached **do**

$Ite = Ite + 1$

**for**  $i = 1$  to  $N$  **do**

User  $i$  receives the vector  $\mathbf{b}_{-i}$

**if**  $(g_i(b_{-i}) \leq 0)$  **then**

$b_i^* = 0$

**else**

$r_i^* = BinarySearch([0, b_{th}], \epsilon)$

$b_i^* = \min(I_i, r_i^*)$

**end if**

**end for**

Check convergence to PNE

**end while**

---

### 3.5.4 Empirical Evaluation

In this section, we provide some numerical results illustrating the operation, features and benefits of the proposed prospect-theoretic decision-making framework. We focus on the pure operational characteristics of our framework, in terms of efficiently controlling the devices' offloaded data. A scalability analysis is provided, and we study our framework's operation under heterogeneous devices. Finally, a comparative empirical evaluation of our approach against alternative offloading strategies is provided. In particular, we consider a base station acting as a MEC server with a coverage area of radius  $R_0 = 50\text{m}$  and  $N = 50$  devices. Device's channel gain is modeled as  $g_i = \frac{1}{d_i^\theta}$ , where  $d_i$  is the distance of device  $i, i \in \aleph$  from the MEC server, and  $\theta$  is the distance loss exponent (e.g.,  $\theta = 2$ ). The transmission's bandwidth is considered  $W = 5\text{MHz}$ . Each device transmits with power  $P_i = \frac{d_i^2}{R_0^2} \text{Watt}$ , which is proportional to its distance from the MEC server. For each device we consider  $F_i^l \in [0.1, 1]$  GHz and  $f_i = 10^{-9} \frac{\text{Joules}}{\text{Cpu-Cycle}}, \forall i \in \aleph$  [230]. A face recognition application is considered with  $I_i \in [1000, 2000]$  KB and  $C_i \in [1000, 2000]$  Mega-Cycles [231, 232]. In the following, unless otherwise explicitly stated, we assume homogeneous users with parameters  $a_i = 0.2$  and  $k_i = 5, \forall i \in \aleph$ . For the MEC server we consider that  $F_{MEC} = 10^3$  GHz and  $b_{th} = 10\% \times \sum_{i=1}^{50} I_i = 6.92 \times 10^6 \text{Bytes}$ .

#### Pure Operation of the Algorithm

Figure 3.14a presents for three indicative users the optimal amount of offloaded data to the MEC server, as a function of the iterations required to converge to the PNE point. We observe that for practical purposes less than twenty iterations are required to reach the PNE, starting from randomly selected initial values of offloaded data. Moreover, each device converges to a different amount of offloaded data, as its decision-making is based both on the MEC server's congestion, and its characteristics. Devices' characteristics are better captured by the factor  $\frac{\lambda_i * I_i}{d_i * F_i^l}$ , presented in Figure

3.14b, which indicates that if a device has either high computation capability ( $F_i^l$ ) or long distance ( $d_i$ ) from the server, then it desires to offload a lower amount of data to avoid the transmission's overhead. However, the more demanding is the device's application (i.e., increased value of  $\lambda_i * I_i$ ) the more the offloading action is desired by the device, so as to reduce the local overhead by executing part of its application at the server. Thus, the higher is the factor  $\frac{\lambda_i * I_i}{d_i * F_i^l}$ , the more beneficial is for the device to offload a larger amount of data to the MEC server.

Figure 3.14c illustrates the average expected overhead and prospect-theoretic (PT) utility as a function of the iterations, while the MEC server's probability of failure is shown in the contained sub-figure. The results reveal that initially the expected average overhead and prospect-theoretic utility, are decreasing and increasing respectively, while the probability of failure also increases, as initially, the MEC server is not congested (low probability of failure) due to the initial random feasible values of users' offloaded data. Thus, the users have a high incentive to increase their offloaded data to reduce their overhead and increase their expected prospect-theoretic utility. As time evolves, this trend leads to the MEC server's overloading (increased probability of failure) and the offloading action becomes less beneficial. Therefore, after a certain point the total offloading at the MEC server reduces and its corresponding probability of failure also reduces, while the users' offloaded data converge to the PNE point, i.e.,  $b_i^*, \forall i \in \mathbb{N}$  (Figure 3.14a).

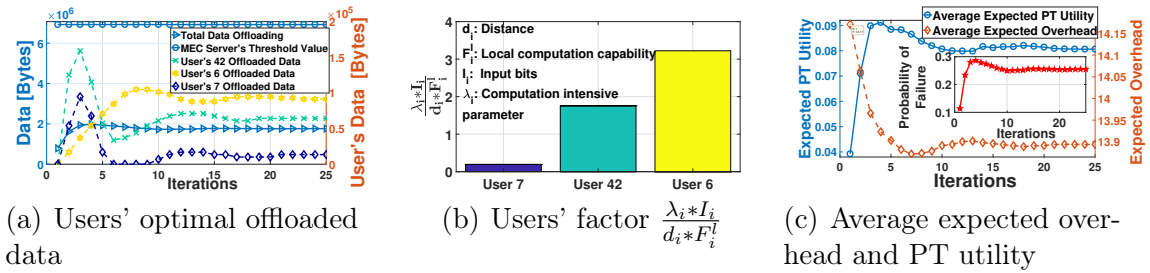
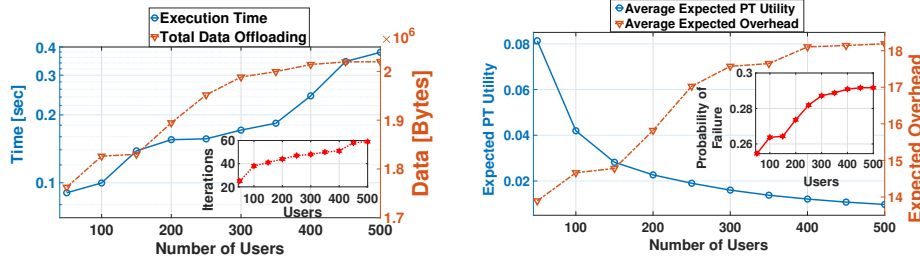


Figure 3.14: Pure operation of the proposed framework

### Scalability Evaluation

Figure 3.15a illustrates the necessary time for convergence to PNE (and as contained sub-figure the corresponding iterations), as well as the total amount of offloaded data to the MEC server, as the number of users increases. It is observed that our prospect-theoretic framework scales very well concerning the increasing number of users, as for an almost ten-fold increase in the number of users, the execution time increases at a significantly lower rate, i.e., a four-fold increase. Similar observations follow concerning the actual number of iterations as well. Furthermore, Figure 3.15b presents the average expected overhead and prospect-theoretic utility at the PNE with respect to the increasing number of users. As the competition for MEC server's computing increases (i.e., increased number of users), the MEC server becomes more congested, and this results in a higher probability of failure, as is depicted by the contained sub-figure. In this case, the offloading action for each user becomes less beneficial, and thus each user offloads a reduced amount of data while executing a bigger portion of its task locally.



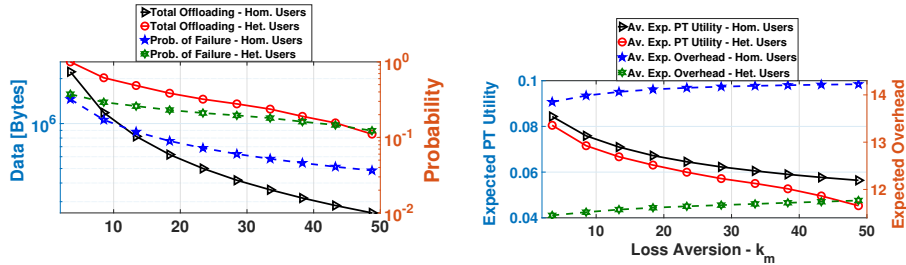
(a) Execution time and total offloaded data

(b) Average expected overhead and PT utility

Figure 3.15: Scalability Evaluation - Number of Users

### Heterogeneous Devices - Loss Aversion

In this section, the impact of the users' heterogeneous loss aversion prospect-theoretic behavior on the achievable performance is studied. Specifically, the results presented in Figure 3.16a and Figure 3.16b compare a scenario of homogeneous users (i.e., same loss aversion parameter  $k_m$  for all users) against a heterogeneous scenario, where each user  $i, i \in \mathbb{N}$ , is associated with a different personalized loss aversion index  $k_i$ . For a fair comparison, for all the users of the homogeneous group the considered loss aversion parameter,  $k_m$  is equal to the average loss aversion parameter value of all the members of the heterogeneous group. That is,  $k_m = \frac{\sum_{i=1}^N k_i}{N}$ . The results reveal that the heterogeneous environment leads to higher congestion levels of the MEC server, as both the total amount of offloaded data and the MEC server's probability of failure, reach higher values (Figure 3.16a), when compared to the corresponding ones of the homogeneous scenario. Furthermore, it is observed that the expected prospect-theoretic utility (Equation 3.39) is decreasing with respect to the total amount of offloaded data  $b_T$ , and as expected the case of the heterogeneous users achieves a lower average expected prospect-theoretic utility (Figure 3.16b), due to the higher congestion levels of the MEC server. Furthermore, from Figure 3.16b we note that the heterogeneous users, by offloading a greater amount of their tasks to the MEC server, experience lower average expected overhead.



(a) Total offloaded data and probability of failure (b) Average expected overhead and PT utility

Figure 3.16: Heterogeneous users - loss aversion parameter  $k_m$  impact study

### Comparative Analysis

Considering the basic setting of homogeneous devices, a comparative study of the proposed optimal approach and solution demonstrates its superiority and benefits over alternative strategies. The comparative evaluation is performed concerning the following metrics: achievable average expected overhead, MEC server's probability of failure, and the total amount of offloaded data. Specifically, we compare our approach, which assumes prospect-theoretic users (PT), to three different approaches, that assume the following users' behaviors: (a) overhead minimizers (OM) users, who selfishly select their offloaded data to minimize their expected overhead, (b) only offloading (OO) users, who are risk-seeking and offload their whole task to the MEC server, and (c) only local (OL) computing users, who are risk-averse and keep the task execution locally, to obtain the "safe" and guaranteed performance provided by their own devices.

Table 3.3 summarizes the corresponding results. Based on the fourth column of Table 3.3, we confirm that the OL users (last row) do not offload any data to the MEC server as expected, the OO users offload all of their data, the OM users offload a significant (but not the whole) amount of their data aiming at minimizing their overhead, while the PT users consider the server's probability of failure and accordingly offload a moderate amount of data. Specifically, when we consider risk-seeking (OO) or risk-averse (OL) users, they experience the worst overhead, as either, they lead the MEC server to failure with probability 1 (case of OO users), or they do not exploit its high computation capability by not offloading any part of their task (case of OL users), respectively. Also, the overhead of the OO users is greater than the one of the OL users, as the first ones have an extra overhead owing to their transmissions. On the other hand, concerning the case of the OM users, even though lower overhead is achieved compared to the previous cases since the users inherently aim at neutral overhead minimization, the selfish users' behavior does not consider the MEC server's failure probability, and eventually leads the MEC server



to overloaded status and high probability of failure. Finally, the PT users achieve the lowest average expected overhead compared to all the other approaches, while at the same time MEC server's probability of failure remains at significantly lower values.

Table 3.3: Comparative Evaluation

User's Nature	Average Expected Overhead	Probability of Failure	Total Offloaded Data [Bytes]
PT	13.8927	0.2546	1762475
OM	14.2385	0.9438	6534778
OO	14.8586	1	76932000
OL	14.3116	0	0

### 3.5.5 Summary

A device-centric risk-based distributed approach was proposed to determine the users' IoT devices' computation offloading volume in a wireless MEC environment, taking into consideration the loss-averse and gain-seeking behavior of the users following the properties of Prospect Theory. The proposed model and approach, is enabled by and enables cognition, and comes in contrast to the majority of existing methods in the literature, that adopt the expected utility maximization theory, where the users are assumed as risk-neutral.

In our setting the MEC server acts as a common pool of resources - CPR (Chapter 2.3) with uncertain payoff returned to the devices, due to its shared nature and the corresponding interdependence among the users' devices, while the choice of computation executed at the local IoT device was assumed to be a safe computation option. Exploiting the local IoT device computing and total offloading overhead, while taking into account each user's cognitive biases and behavior, the optimal amount of each user's offloaded data to the MEC server was obtained as the outcome of a non-cooperative game among the users.

The existence and uniqueness of a PNE point were shown, and an algorithm that converges to the optimal values of offloaded data for each user in a distributed manner was designed. Detailed numerical results were obtained via modeling and simulation, that demonstrated the operation features and superiority of the proposed cognitive-enabled framework, under both cases of homogeneous and heterogeneous users.

### **3.6 Risk-aware Data Offloading in Multi-access Mobile Edge Computing**

Multi-access Mobile Edge Computing (MEC) enables resource-constrained mobile devices to offload, either partially or completely, computationally intensive tasks to a set of servers at the edge of the network. Given that the shared nature of the servers' resources introduces high computation and communication uncertainty, here we extend the research work in Chapter 3.5 by considering users' risk-seeking or loss-aversion behavior in their final decisions regarding the portion of their computing tasks to be offloaded at each server in a multi-MEC server environment while executing the rest locally. This is achieved by capitalizing on the power and principles of Prospect Theory (Chapter 2.3) and Tragedy of the Commons, treating each MEC server as a Common Pool of Resources available to all the users, while being rivalrous and subtractable, thus may potentially fail if over-exploited by the users. The goal of each user becomes to maximize its perceived satisfaction, as expressed through a properly formulated prospect-theoretic utility function, by offloading a portion of its computing tasks to the different MEC servers. To address this problem and conclude to the optimal allocation strategy, a non-cooperative game among the users is formulated and the corresponding Pure Nash Equilibrium (PNE), i.e., optimal data offloading, is determined, while a distributed low-complexity algorithm that converges to the PNE is introduced. The performance and key principles of

the proposed framework are demonstrated through modeling and simulation, while useful insights about the users' data offloading decisions under realistic conditions and behaviors are presented.

### 3.6.1 System Model

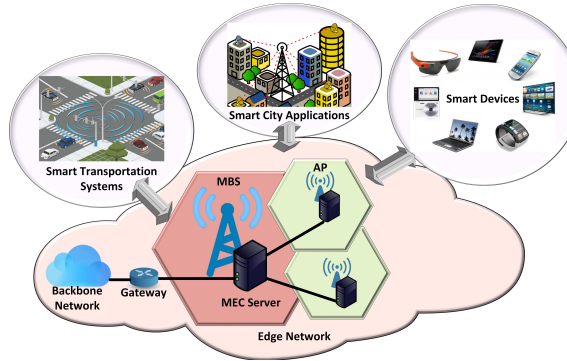


Figure 3.17: Multi-MEC servers environment

A multi-access edge computing (MEC) system with *multi-MEC servers*, as shown in Figure 3.17, is considered, where the users can offload part of their application to the MEC system through a 5G heterogeneous network. The MEC servers can be small data centers at the edge of the network and possibly managed by different Wireless Internet Service Providers (WISPs). Following the existing literature in the field of multi-access edge computing, the MEC servers reside at the Macro Base Stations (MBSs) of the macrocells or the Access Points (APs) of the small cells, e.g., femtocells [179, 176]. Considering that typically the small cells and the macrocell are overlapping, each user is assumed capable of potentially offloading part of its data to all the MEC servers in the examined scenario. By offloading a portion of the user application data to the MEC system, the computing performance and energy consumption that the users observe, are significantly improved and reduced respectively, thus overall enhancing user experience. Data offloading decision is a dynamic process that depends not only on the communication and computing environment

### Chapter 3. Mobile Edge Computing

but also on the type of users' requested services. These services may impose different time and energy constraints in the data offloading problem based on their real and non-real-time nature. Accordingly, the users may request either elastic or inelastic services, such as executing a machine learning algorithm for data analytics purposes or online gaming, respectively.

In our model, we denote by  $\mathbb{U} = \{1, \dots, i, \dots, U\}$  the set of users, and with  $\mathbb{S} = \{1, \dots, s, \dots, S\}$  the set of MEC servers in the system. Furthermore, each user  $i \in \mathbb{U}$  has a computing application to be completed, with a certain affordable delay and energy consumption, related to the user's energy availability. Specifically, we denote by  $A_i = (B_i, C_i, \phi_i, t_i, e_i)$  the user's  $i$  computing application, which is characterized by specific features and requirements. In particular, let  $B_i$  denote the total input bits and  $C_i$  the number of CPU cycles required for the execution of the requested computing application. We set  $C_i = \phi_i \cdot B_i$ , where  $\phi_i$ ,  $\phi_i > 0$  describes the application's intensity, e.g., a higher value of  $\phi_i$  expresses a more computing demanding application. For application  $A_i$ ,  $t_i$  is the time constraint that the user  $i$  requests regarding its completion, and  $e_i$  denotes user's  $i$  energy availability at its own device.

In this research work, we assume that for each user  $i \in \mathbb{U}$ , the application  $A_i$  can be arbitrarily partitioned into subsets of any size, which can be offloaded to any of the available MEC servers. We denote by  $\mathbf{b}_i = (b_{i,1}, \dots, b_{i,S})$  the user's  $i$  offloading vector and  $b_{i,s}$  is the amount of data that user  $i$  offloads to the MEC server  $s$ . Thus,  $b_{i,s} \in [0, B_i]$  and  $\sum_{s \in \mathbb{S}} b_{i,s} \leq B_i, \forall i \in \mathbb{U}$ . It is noted that each user transmits sequentially its data  $b_{i,s}, \forall s \in \mathbb{S}$  via exploiting its single-interface communication capabilities and each MEC server's wireless channel. Consequently, the amount of data that will be executed locally at the device is:  $(B_i - \sum_{s \in \mathbb{S}} b_{i,s})$ .

In a realistic multi-MEC servers system, the end-users sense their environment and available options of the MEC servers. The user devices are sufficiently intelligent and make optimal data offloading decisions in an autonomous and distributed

manner while expressing and considering the users' risk-aware behavioral characteristics. On the other hand, having a centralized load balancer to control the users' optimal data offloading, introduces several drawbacks in the system design and efficiency, which our proposed solution bypasses. *First*, the load balancer is a centralized decision-making entity, which is prone to be a single point of failure that can be attacked, e.g., distributed denial of service (DDoS) attacks, and the system can miss operate. *Second*, it is assumed that all the service operators owning the various MEC servers will accept and trust the centralized load-balancer to control the data offloading to them. *Third*, even in the simple case of considering risk-neutral rational users, the users are burdened by signaling overhead to report their characteristics to the centralized load-balancer. *Fourth*, in the case of risk-aware users, as considered in the proposed framework, the centralized load-balancer has no feasible way to know the user's behavioral characteristics and the users are reluctant to reveal them due to privacy concerns. Based on the above description, we evangelize that a distributed risk-aware data offloading in multi-MEC server environments is a more realistic framework compared to a centralized approach.

### Communication Model

Each AP/MBS operates and receives data over a dedicated communication link, i.e., frequency band, thus, each user, while transmitting part of its data to a MEC server, senses the interference from the rest of the users transmitting only to the same MEC server, i.e.,  $\sum_{j \in N_s, j \neq i} p_{j,s} \cdot g_{j,s}$ , where the communication channel gain between the user  $j$  and the MEC server  $s$  is denoted by  $g_{j,s}$ ,  $N_s = \{j \in \mathbb{U} : b_{j,s} \neq 0\}$  is the set of users that offload part of their application to server  $s$ , and  $p_{i,s}$  is the user's  $i$  transmission power to offload part of the data to server  $s$ . The signal-to-interference-plus-noise-ratio (SINR) measured at the receiver side, i.e., MEC servers, with respect to the transmission of user  $i$  is  $\gamma_{i,s} = \frac{p_{i,s} \cdot g_{i,s}}{\sigma_0^2 + \sum_{j \in N_s, j \neq i} p_{j,s} \cdot g_{j,s}}$ , and given that the bandwidth allocated to each communication link is  $W$ , the corresponding user's

achievable data rate, while communicating with server  $s$ , is [223]:

$$R_{i,s} = W \cdot \log\left(1 + \frac{p_{i,s} \cdot g_{i,s}}{\sigma_0^2 + \sum_{j \in N_s, j \neq i} p_{j,s} \cdot g_{j,s}}\right) \quad (3.47)$$

where  $\sigma_0^2$  indicates the variance of the Additive White Gaussian Noise (AWGN) of the server  $s$ . The user  $i$ , by offloading  $b_{i,s}$  amount of data to the MEC server  $s$ , experiences an overhead consisting of: a) the transmission time [sec] of the data

$$O_{i,s}^{m,t} = \frac{b_{i,s}}{R_{i,s}} \quad (3.48)$$

and b) the transmission energy consumption [Joules]

$$O_{i,s}^{m,e} = \frac{b_{i,s} \cdot p_{i,s}}{R_{i,s}} \quad (3.49)$$

### Computing Model

1) *Multi-access Edge Computing Model*: We assume that a strong computing resource (e.g., a high-speed CPU) is available at each MEC server, while the computing capability of each server is limited by the total amount of data  $\tilde{b}_s$  that can process at the same time, e.g., due to either limited memory storage or finite multi-core architecture of the MEC server. The total computing capability of each MEC server  $s$ , which is denoted by  $F_s$  [Cycles/sec], is shared among the users that select to offload  $b_{i,s}$  amount of data to the MEC server  $s$ . Thus, the computing capability that is assigned to user  $i$  (e.g., through a virtual machine) in order to remotely execute part of its application is expressed via user's  $i$  return function  $F_{i,s}(\bar{b}_s)$  that is given as follows:

$$F_{i,s}(\bar{b}_s) = \frac{\phi_i}{\sum_{j \in N_s} \phi_j} \cdot f_s(\bar{b}_s) \quad (3.50)$$

where  $\bar{b}_s = \sum_{j \in N_s} b_{j,s}$  is the total amount of offloaded data to MEC server  $s$ , and  $f_s$  defines the server's  $s$  production function expressing users' perceived computing satisfaction from the MEC server  $s$ , and is given as follows:

$$f_s(\bar{b}_s) = \begin{cases} (1 - \frac{\bar{b}_s}{\tilde{b}_s}) \cdot F_s & , \text{ if } \bar{b}_s \leq \tilde{b}_s \\ 0 & , \text{ otherwise} \end{cases} \quad (3.51)$$

where  $\tilde{b}_s$  denotes the received bytes threshold value that the MEC server can process without failing its operation.

**Proposition 3** *Each MEC server's  $s$ ,  $s \in \mathbb{S}$ , production function  $f_s(\bar{b}_s)$ , and each user's  $i$ ,  $i \in \mathbb{U}$  return function  $F_{i,s}(\bar{b}_s)$ , are strictly decreasing with respect to the total offloaded data  $\bar{b}_s$  at the MEC server  $s$ .*

The return function of user  $i$  (Equation 3.50) is personalized based on the computing demand  $\phi_i$  of its application, and due to Equation 3.51 decreases as the total computing offloading  $\bar{b}_s$  increases.

User  $i$  can execute remotely its offloaded data by receiving a computing capability  $F_{i,s}$  from the server, and the corresponding execution time is  $\frac{\phi_i \cdot b_{i,s}}{F_{i,s}}$ . As a result, based on Equation 3.48 user's  $i$  total time overhead is calculated as follows:

$$O_{i,s}^{m,t}|_{total} = \frac{b_{i,s}}{R_{i,s}} + \frac{\phi_i \cdot b_{i,s}}{F_{i,s}} \quad (3.52)$$

Based on Equation 3.49 and Equation 3.52 the relative MEC overhead that user  $i$  experiences by deciding to offload part of its application to the server  $s$ , considering both the user's application time constraints and the user's energy availability, is formulated as follows:

$$O_{i,s}^m(b_{i,s}) = \frac{\frac{b_{i,s}}{R_{i,s}} + \frac{\phi_i \cdot b_{i,s}}{F_{i,s}}}{t_i} + \frac{\frac{b_{i,s} \cdot p_{i,s}}{R_{i,s}}}{e_i} \quad (3.53)$$

and the overall multi-access edge computing overhead  $O_i^m = \sum_{s \in \mathbb{S}} O_{i,s}^m$ , is given as

$$O_i^m(\mathbf{b}_i) = \sum_{s \in \mathbb{S}} b_{i,s} \cdot \left( \frac{1}{R_{i,s} \cdot t_i} + \frac{\phi_i}{F_{i,s} \cdot t_i} + \frac{p_{i,s}}{R_{i,s} \cdot e_i} \right) \quad (3.54)$$

2) *Local Computing Model*: For the local computing model, user  $i \in \mathbb{U}$  executes  $L_i = B_i - \sum_{s \in \mathbb{S}} b_{i,s}$  amount of data locally at its device. By denoting with  $lc_i$  [Cycles/sec] user's  $i$  local computing capability and with  $le_i$  [Joules/Cycle] user's  $i$  energy consumption to process locally the data, the local computing execution time is given as follows:

$$O_i^{l,t} = \frac{\phi_i \cdot L_i}{lc_i} \quad (3.55)$$

while user's local energy consumption to process the data is determined as follows:

$$O_i^{l,e} = \phi_i \cdot L_i \cdot le_i \quad (3.56)$$

Based on Equation 3.55 and Equation 3.56, the user's relative overhead regarding the local computing approach considering both the computing time and the energy consumption overhead, is given as follows:

$$O_i^l(L_i) = \frac{O_i^{l,t}}{t_i} + \frac{O_i^{l,e}}{e_i} = \phi_i \cdot L_i \cdot \left( \frac{1}{t_i \cdot lc_i} + \frac{le_i}{e_i} \right) \quad (3.57)$$

### Actual Total Overhead

Based on Equation 3.54 and Equation 3.57, user's  $i$  total overhead is given as follows:

$$O_i = \sum_{s \in \mathbb{S}} b_{i,s} \cdot \left( \frac{1}{R_{i,s} \cdot t_i} + \frac{\phi_i}{F_{i,s} \cdot t_i} + \frac{p_{i,s}}{R_{i,s} \cdot e_i} \right) + \phi_i \cdot L_i \cdot \left( \frac{1}{t_i \cdot lc_i} + \frac{le_i}{e_i} \right) \quad (3.58)$$

Note that each user  $i \in \mathbb{U}$  with strategy  $\mathbf{b}_i = (b_{i,1}, \dots, b_{i,S})$  can evaluate its experienced total overhead by receiving from each server  $s$  (via server's broadcasting), the total interference  $\sum_{j \in N_s, j \neq i} p_{j,s} \cdot g_{j,s}$ , the overall applications' levels of computing intensity  $\sum_{j \in N_s} \phi_j$ , and the total amount of offloaded data  $\bar{b}_s$ , without requiring any additional information of the individual users.

## 3.6.2 The Prospect of Data Offloading

### Risk-aware Behavior: The Tragedy of the Commons

In the *multi-MEC servers'* environment, each MEC server constitutes a Common Pool of Resources (CPR), since all the users can arbitrarily offload part of their applications to the MEC servers for remote execution. Due to Equation 3.50 and Equation 3.51 each MEC server  $s$  is rivalrous and subtractable, as the MEC server's computing capability is a shared resource among the users. Specifically, Equation 3.51 denotes that the  $\tilde{b}_s$  is the received bytes threshold value for each MEC server  $s$ ,



thus if  $\bar{b}_s \geq \tilde{b}_s$  the MEC server  $s$  is considered unable to execute the receiving amount of applications at the same time, so it "fails". This phenomenon is well known in the literature as the Tragedy of the Commons [233]. As a result, in the case of the CPR's failure, it is more beneficial for the user  $i$  either to offload the  $b_{i,s}$  amount of data to another MEC server, or process the data locally.

Towards minimizing the perceived overhead, the user's  $i$  goal is to determine in an autonomous and distributed manner the offloading amount of data  $b_{i,s}$  to each MEC server  $s$  by accounting for the uncertainty of the expected outcome. The uncertainty introduced by the shared computing environment drives the users to exhibit a risk-aware behavior. Based on this uncertainty, we introduce the probability of failure of each MEC server  $s$ , which is denoted by  $p_s(\bar{b}_s)$ . The probability of failure characterizes each MEC server and represents its probability to fail to serve the end-users' computing requests due to the over-exploitation of its computing capabilities.

**Assumption 1** *Each MEC server's  $s$  (CPR) probability of failure  $p_s(\bar{b}_s)$  is strictly increasing, convex and twice continuously differentiable with respect to  $\bar{b}_s \in [0, \tilde{b}_s)$ , with  $p_s(\bar{b}_s) = 1, \forall \bar{b}_s \geq \tilde{b}_s$ .*

We consider a linear probability of failure function for each MEC server  $s$ , thus  $p_s(\bar{b}_s) = \frac{\bar{b}_s}{\tilde{b}_s}, \forall \bar{b}_s < \tilde{b}_s$ , while  $p_s(\bar{b}_s) = 1, \forall \bar{b}_s \geq \tilde{b}_s$  in order to represent a smooth potential failure of the MEC server, if its computing capabilities become over-exploited by the users. Each user  $i$  with strategy  $\mathbf{b}_i$ , by offloading  $b_{i,s}$  amount of data to the MEC server  $s$ , should consider the server's probability of failure  $p_s$ , since in the case that the server  $s$  "fails" to execute user's  $i$  amount of offloaded data, the user has to process the data locally. Given that the MEC server's  $s$  probability of failure is  $p_s$ , then the probability that the server survives and executes successfully the total received amount of offloaded data is accordingly  $(1 - p_s)$ . Based on Equation 3.53, Equation 3.57, and given  $p_s$ , user's  $i$  expected MEC overhead from the server  $s$ , is

formulated as follows:

$$\begin{aligned} \mathbb{E}(O_{i,s}^m(b_{i,s})) &= (1 - p_s(\bar{b}_s)) \cdot O_{i,s}^m(b_{i,s}) \\ &\quad + p_s(\bar{b}_s) \cdot (O_i^l(b_{i,s}) + \frac{b_{i,s}}{R_{i,s} \cdot t_i} + \frac{b_{i,s} \cdot p_{i,s}}{R_{i,s} \cdot e_i}) \end{aligned} \quad (3.59)$$

where the last two factors in the second term refer to the additional communication overhead in the case of the MEC server's failure, as the user offloads its data to the server, and then the server's failure is observed.

Following the same reasoning as in Equation 3.54, and applying the operation of expectation, the overall expected multi-access edge computing overhead that user  $i$  experiences is  $\mathbb{E}(O_i^m(\mathbf{b}_i)) = \sum_{s \in \mathbb{S}} \mathbb{E}(O_{i,s}^m(b_{i,s}))$ . Consequently, taking the expectation of the overall MEC overhead, where the first term of Equation 3.58 becomes  $\mathbb{E}(O_i^m(\mathbf{b}_i))$ , the user's  $i$  overall perceived expected overhead is given as follows:

$$\mathbb{E}(O_i) = \sum_{s \in \mathbb{S}} \mathbb{E}(O_{i,s}^m(b_{i,s})) + \phi_i \cdot L_i \cdot \left( \frac{1}{t_i \cdot l c_i} + \frac{l e_i}{e_i} \right) \quad (3.60)$$

where the overall local computing overhead remains the same (i.e., second term of both Equation 3.58 and Equation 3.60).

### Offloading Decision under Prospect Theory

To address the users' subjectivity in decision-making under uncertainty, as they tend to exhibit different decisions under losses or gains with respect to their actual satisfaction, Prospect Theory (Chapter 2.3) has been adopted. Here, the reference point for each user is the guaranteed overhead  $O_i^l(b_{i,s})$  (Equation 3.57) that the user  $i$  can obtain by processing the  $b_{i,s}$  amount of data locally instead of offloading them to the server  $s$ . To this end, the user's  $i$  prospect-theoretic utility, when offloading  $b_{i,s}$  amount of data to the MEC server  $s$  is formulated as follows:

$$u_{i,s}(q_{i,s}) = \begin{cases} (q_{i,r} - q_{i,s})^{\alpha_i} & , \text{ if } q_{i,s} \leq q_{i,r} \\ -k_i \cdot (q_{i,s} - q_{i,r})^{\gamma_i} & , \text{ if } q_{i,s} > q_{i,r} \end{cases} \quad (3.61)$$

### Chapter 3. Mobile Edge Computing

where  $q_{i,s} = O_{i,s}^m(b_{i,s})$  is the user's  $i$  actual perceived MEC overhead by offloading  $b_{i,s}$  amount of data to the MEC server  $s$ , as defined in Equation 3.53, and  $q_{i,r} = O_i^l(b_{i,s})$  denotes the reference point of the user's  $i$  prospect-theoretic utility.

Considering the case that the MEC server  $s$  does not fail due to the users' offloaded amount of data, the user's  $i$  MEC overhead  $q_{i,s}$  is calculated by Equation 3.53, and in this case, it is lower than the corresponding local computing overhead  $q_{i,r}$  (reference point), thus  $q_{i,s} \leq q_{i,r}$ . As a result, based on user's  $i$  first branch of its prospect-theoretic utility (Equation 3.61), via subtracting  $q_{i,s}$  from the reference point  $q_{i,r}$ , we have  $u_{i,s} = [b_{i,s}(\frac{\phi_i}{t_i l c_i} + \frac{l e_i \phi_i}{e_i} - \frac{1}{t_i R_{i,s}} - \frac{\phi_i}{t_i F_{i,s}} - \frac{p_{i,s}}{e_i R_{i,s}})]^{\alpha_i}$ . On the other hand, if the MEC server  $s$  fails to execute the received amount of offloaded data due to the fact that it is overloaded, then the user  $i$  has to process the  $b_{i,s}$  amount of data locally, thus its experienced overhead is given by Equation 3.57, while it has an extra communication overhead, as user  $i$  at first had to offload the  $b_{i,s}$  amount of data to the MEC server  $s$ . As a result, user's  $i$  actual experienced overhead in the case of the MEC server's  $s$  failure consists of the local computing overhead  $q_{i,r}$  (reference point) and the extra communication overhead, thus  $q_{i,s} = q_{i,r} + \frac{b_{i,s}}{R_{i,s} \cdot t_i} + \frac{b_{i,s} \cdot p_{i,s}}{R_{i,s} \cdot e_i}$ , and is greater than the reference point  $q_{i,r}$ . Therefore, based on the second branch of Equation 3.61, by subtracting the reference point  $q_{i,r}$  from user's  $i$  actual multi-access edge computing overhead  $q_{i,s}$ , its prospect-theoretic utility becomes  $u_{i,s} = -k_i \cdot [b_{i,s} \cdot (\frac{1}{R_{i,s} \cdot t_i} + \frac{p_{i,s}}{R_{i,s} \cdot e_i})]^{\alpha_i}$ .

Furthermore, for notational convenience we define  $\epsilon_i = (\frac{1}{R_{i,s} \cdot t_i} + \frac{p_{i,s}}{R_{i,s} \cdot e_i})^{\alpha_i}$ , and  $h_{i,s}(\bar{b}_s) = (\frac{\phi_i}{t_i l c_i} + \frac{l e_i \phi_i}{e_i} - \frac{1}{t_i R_{i,s}} - \frac{\phi_i}{t_i F_{i,s}} - \frac{p_{i,s}}{e_i R_{i,s}})^{\alpha_i}$ , considering that  $h_{i,s} > 0$  if the MEC server  $s$  does not fail. Considering the probability of failure  $p_s(\bar{b}_s)$ , the user's  $i$  prospect-theoretic utility can be written as:

$$u_{i,s} = \begin{cases} b_{i,s}^{\alpha_i} \cdot h_{i,s}(\bar{b}_s) & , \text{ with probabil. } 1 - p_s(\bar{b}_s) \\ -k_i \cdot \epsilon_i \cdot b_{i,s}^{\alpha_i} & , \text{ with probabil. } p_s(\bar{b}_s) \end{cases} \quad (3.62)$$

Based on Equation 3.62, and as per Equation 2.37, each user's  $i$  expected prospect-

theoretic utility regarding MEC server  $s$  is formulated as follows:

$$\begin{aligned}\mathbb{E}(u_{i,s}) &= b_{i,s}^{\alpha_i} \cdot h_{i,s}(\bar{b}_s)(1 - p_s(\bar{b}_s)) - k_i \epsilon_i b_{i,s}^{\alpha_i} p_s(\bar{b}_s) \\ &\triangleq b_{i,s}^{\alpha_i} \cdot \text{ert}_{i,s}(\bar{b}_s)\end{aligned}\tag{3.63}$$

where  $\text{ert}_{i,s}(\bar{b}_s) = h_{i,s}(\bar{b}_s)(1 - p_s(\bar{b}_s)) - k_i \epsilon_i p_s(\bar{b}_s)$  is the effective rate of return of the MEC server  $s$  for the user  $i$ .

### 3.6.3 Prospect-Theoretic Partial Offloading

Each user  $i$  has to sophisticatedly and selfishly determine its best offloading strategy in order to maximize its overall perceived expected prospect-theoretic utility, i.e.,  $\sum_{s \in \mathbb{S}} \mathbb{E}(u_{i,s})$ . In this process, there is a natural tradeoff between the user's  $i$  overall MEC overhead and its overall local computing overhead. To capture this tradeoff, we introduce each user's  $i$  satisfaction utility, which is expressed by its overall expected prospect-theoretic utility subtracting its overall local computing overhead as follows.

$$s_i(\mathbf{b}_i, \mathbf{b}_{-i}) = \sum_{s \in \mathbb{S}} \mathbb{E}(u_{i,s}) - O_i^l(L_i)\tag{3.64}$$

where  $\mathbf{b}_{-i} = [\mathbf{b}_1 \dots, \mathbf{b}_{i-1}, \mathbf{b}_{i+1}, \dots, \mathbf{b}_U]$  is the users' offloading strategies' vector except for the user  $i$ ,  $\sum_{s \in \mathbb{S}} \mathbb{E}(u_{i,s})$  is the overall expected prospect-theoretic utility that user  $i$  obtains, and  $O_i^l(L_i)$  is given by Equation 3.57, where  $L_i = B_i - \sum_{s \in \mathbb{S}} b_{i,s}$  is the amount of locally processed data.

Therefore, the ultimate goal of each user  $i$  is to maximize its perceived satisfaction utility  $s_i$  by determining its data offloading strategy  $\mathbf{b}_i$ . This problem can be formulated as a maximization problem of user's  $i$  satisfaction utility, and based on Equation 3.57 and Equation 3.63 can be expressed as follows.

$$\max_{\mathbf{b}_i \in \Gamma_i} s_i(\mathbf{b}_i, \mathbf{b}_{-i}) = \sum_{s \in \mathbb{S}} b_{i,s}^{\alpha_i} \cdot \text{ert}_{i,s}(\bar{b}_s) - \phi_i L_i \left( \frac{1}{t_i \cdot l c_i} + \frac{l e_i}{e_i} \right)\tag{3.65}$$

where  $\Gamma_i = \overbrace{[0, \dots, B_i] \times \dots \times [0, \dots, B_i]}^{S \text{ - times}}$  is the strategy set of user  $i$ .

### Chapter 3. Mobile Edge Computing

Due to the non-cooperative and distributed nature of the above maximization problem, it can be treated as a non-cooperative game among the users who act as players making the optimal decisions about themselves in a selfish and distributed manner. Let  $G = [\mathbb{U}, \{\Gamma_i\}_{i \in \mathbb{U}}, \{s_i\}_{i \in \mathbb{U}}]$  denote the non-cooperative game among the users which set is  $\mathbb{U}$ , where each user's strategy space is  $\Gamma_i$ , and its payoff is the satisfaction utility  $s_i$  (Equation 3.64). Towards solving the non-cooperative game  $G$ , the concept of Nash equilibrium is adopted (Chapter 2.1). The Nash equilibrium (NE) of the non-cooperative game  $G$  is the strategy vector which consists of users' offloading vectors,  $\mathbf{b}^* = [\mathbf{b}_1^*, \dots, \mathbf{b}_i^*, \dots, \mathbf{b}_U^*]$ , where no user has the incentive to change its own strategy (i.e., at least the amount of offloading data at one MEC server  $s$ ) given the strategies of the rest of the users. We denote as  $\mathbf{b}_{-i}^* = [\mathbf{b}_1^*, \dots, \mathbf{b}_{i-1}^*, \mathbf{b}_{i+1}^*, \dots, \mathbf{b}_D^*]$  the users' offloading strategies vector except for user  $i$  at the NE point.

$$\begin{aligned} \mathbb{E}(O_i)|_t &= \sum_{s \in \mathbb{S}} [b_{i,s}(\frac{1}{R_{i,s}} + \frac{\phi_i}{F_{i,s}})(1 - p_s(\bar{b}_s)) + b_{i,s}(\frac{1}{R_{i,s}} + \frac{\phi_i}{lc_i})p_s(\bar{b}_s)] + \frac{\phi_i L_i}{lc_i} \\ &= \frac{F_{i,s} = \sum_{j \in N_s} \phi_j (1 - \frac{\bar{b}_s}{b_s}) F_s}{p_s = \bar{b}_s / \tilde{b}_s, \bar{b}_s = b_{i,s} + b_{-i,s}} \sum_{s \in \mathbb{S}} [b_{i,s}(\frac{1}{R_{i,s}} + \frac{\sum_{j \in N_s} \phi_j}{F_s} + \frac{\phi_i}{lc_i}(\frac{b_{i,s} + b_{-i,s}}{\tilde{b}_s}))] + \frac{\phi_i L_i}{lc_i} \end{aligned} \quad (3.66)$$

$$\begin{aligned} \mathbb{E}(O_i)|_e &= \sum_{s \in \mathbb{S}} [b_{i,s} \frac{p_{i,s}}{R_{i,s}} (1 - p_s(\bar{b}_s)) + b_{i,s}(\frac{p_{i,s}}{R_{i,s}} + \phi_i l e_i) p_s(\bar{b}_s)] + \phi_i L_i l e_i \\ &= \frac{F_{i,s} = \sum_{j \in N_s} \phi_j (1 - \frac{\bar{b}_s}{b_s}) F_s}{p_s = \bar{b}_s / \tilde{b}_s, \bar{b}_s = b_{i,s} + b_{-i,s}} \sum_{s \in \mathbb{S}} [b_{i,s}(\frac{p_{i,s}}{R_{i,s}} + \phi_i l e_i (\frac{b_{i,s} + b_{-i,s}}{\tilde{b}_s}))] + \phi_i L_i l e_i \end{aligned} \quad (3.67)$$

### Problem Formulation

Each user  $i$  aims at maximizing its satisfaction utility  $s_i$ , while at the same time experiencing a non-negative expected prospect-theoretic utility  $\mathbb{E}(u_{i,s}) \geq 0$ . If  $\mathbb{E}(u_{i,s}) < 0$ , then the  $b_{i,s}$  amount of data that the user  $i$  offloads to the MEC server  $s$ , drives the latter to a high probability of failure  $p_s$ , thus the user's offloading is not beneficial.

Additionally, each user aims at satisfying its time  $t_i$  and energy  $e_i$  constraints, as follows:  $\mathbb{E}(O_i)|_t \leq t_i$  and  $\mathbb{E}(O_i)|_e \leq e_i$ , where  $\mathbb{E}(O_i)|_t$  and  $\mathbb{E}(O_i)|_e$  are given by

Equation 3.66 and Equation 3.67, respectively. Therefore, the maximization problem of user's  $i$  satisfaction utility (Equation 3.65-3.67) can be formulated as follows:

$$\begin{aligned} & \underset{\mathbf{b}_i \in \Gamma_i}{\text{maximize}} && s_i(\mathbf{b}_i, \mathbf{b}_{-i}) \\ & \text{subject to} && \left. \begin{aligned} & \sum_{s \in \mathbb{S}} b_{i,s} \leq B_i, \\ & \mathbb{E}(u_{i,s}) \geq 0, \quad \forall s \in \mathbb{S}, \\ & \mathbb{E}(O_i)|_t \leq t_i, \\ & \mathbb{E}(O_i)|_e \leq e_i \end{aligned} \right\} (C_i) \end{aligned} \quad (3.68)$$

where  $(C_i)$  denotes the group of the constraints that user's  $i$  offloading strategy  $\mathbf{b}_i$  should satisfy.

### Existence, Uniqueness and Convergence of PNE

Let us denote as  $\mathbb{A}_i$  the set of each user's  $i$  strategy space, where  $\mathbb{A}_i = \Gamma_i \cap \mathbb{C}_i$ ,  $\mathbb{C}_i = \{\mathbf{b}_i \in \Gamma_i : \mathbf{b}_i \text{ satisfies } (C_i)\}$ . Thus, the non-cooperative game  $G$  is transformed to  $\mathbb{G} = [\mathbb{U}, \{\mathbb{A}_i\}_{i \in \mathbb{U}}, \{s_i\}_{i \in \mathbb{U}}]$ .

**Theorem 10** *The non-cooperative game  $\mathbb{G}$  among the users is an  $n$ -person concave game, where  $n = U$ .*

In order to prove the above theorem, we first state the following Lemmas 5 - 8

**Lemma 5** *For each user  $i$  and each MEC server  $s$  there exists a value  $b_{i,s}^{th} \geq 0$  such that  $ert_{i,s}(b_{i,s}^{th}) = 0$  and  $\mathbb{E}(u_{i,s}) \geq 0, \forall b_{i,s} \leq b_{i,s}^{th}$ , while  $\mathbb{E}(u_{i,s}) < 0, \forall b_{i,s} > b_{i,s}^{th}$ .*

*Proof:* The first order derivative of the effective rate of return  $ert_{i,s}(\bar{b}_s)$  of the MEC server  $s$ ,  $s \in \mathbb{S}$  regarding user  $i$ ,  $i \in \mathbb{U}$  is given as follows:

$$\frac{\partial ert_{i,s}(\bar{b}_s)}{\partial \bar{b}_s} = \frac{\partial h_{i,s}(\bar{b}_s)}{\partial \bar{b}_s} (1 - p_s(\bar{b}_s)) - \frac{\partial p_s(\bar{b}_s)}{\partial \bar{b}_s} (h_{i,s}(\bar{b}_s) + k_i \epsilon_i) \quad (3.69)$$

It is obvious that  $\frac{\partial p_s(\bar{b}_s)}{\partial \bar{b}_s} > 0$ , since the MEC server's  $s$ ,  $s \in \mathbb{S}$  probability of failure  $p_s(\bar{b}_s)$  is strictly increasing with respect to  $\bar{b}_s$ . Also  $(1 - p_s(\bar{b}_s)) > 0$ , and  $\frac{\partial h_{i,s}(\bar{b}_s)}{\partial \bar{b}_s} < 0$ ,

### Chapter 3. Mobile Edge Computing

since the  $h_{i,s}(\bar{b}_s)$  is strictly decreasing with respect to  $\bar{b}_s$  based on Proposition 3. As a result  $\frac{\partial \text{ert}_{i,s}(\bar{b}_s)}{\partial \bar{b}_s} < 0$ , thus  $\text{ert}_{i,s}(\bar{b}_s)$  is strictly decreasing with respect to  $\bar{b}_s$ . CASE A: If  $\text{ert}_{i,s}(0) \leq 0$ , then,  $\text{ert}_{i,s}(b_{i,s}) \leq \text{ert}_{i,s}(0) \leq 0$ ,  $\forall b_{i,s} \leq B_i$  and  $\mathbb{E}(u_{i,s}) \leq 0$  (Equation 3.63). So, in this case either  $b_{i,s}^{th} = 0$  or it is not defined since  $\mathbb{E}(u_{i,s}) < 0$ ,  $\forall b_{i,s} \leq B_i$ . CASE B: If  $\text{ert}_{i,s}(0) > 0$ , then since  $\text{ert}_{i,s}(\tilde{b}_s) = -k_i \epsilon_i < 0$  due to  $p_s(\tilde{b}_s) = 1$ , then  $\exists b_{i,s}^{th} \in [0, \tilde{b}_s]$ , such that  $\text{ert}_{i,s}(b_{i,s}^{th}) = 0$  based on the Intermediate Value Theorem [227].

Based on Lemma 5, the maximization problem in Equation 3.68 can be rewritten as follows.

$$\begin{aligned} & \underset{\mathbf{b}_i \in A_i}{\text{maximize}} && s_i(\mathbf{b}_i, \mathbf{b}_{-i}) \\ & \text{subject to} && \left. \begin{aligned} & \sum_{s \in \mathbb{S}} b_{i,s} \leq B_i, \\ & 0 \leq b_{i,s} \leq b_{i,s}^{th}, \quad \forall s \in \mathbb{S}, \\ & \mathbb{E}(O_i)|_t \leq t_i, \\ & \mathbb{E}(O_i)|_e \leq e_i \end{aligned} \right\} (C_i) \end{aligned} \quad (3.70)$$

where the second constraint in  $(C_i)$  was replaced by  $0 \leq b_{i,s} \leq b_{i,s}^{th}$ .

**Lemma 6** *For each user  $i$  and each MEC server  $s$ , the expected prospect-theoretic utility  $\mathbb{E}(u_{i,s})$  is strictly concave  $\forall b_{i,s} \in (0, b_{i,s}^{th})$ .*

*Proof:* The expected prospect-theoretic utility's first order derivative is given as follows:

$$\begin{aligned} \frac{\partial \mathbb{E}(u_{i,s})}{\partial b_{i,s}} &= [b_{i,s}^{a_i} \frac{\partial h_{i,s}(\bar{b}_s)}{\partial \bar{b}_s} + a_i b_{i,s}^{a_i-1} h_{i,s}(\bar{b}_s)](1 - p_s(\bar{b}_s)) \\ &\quad - b_{i,s}^{a_i} h_{i,s}(\bar{b}_s) \frac{\partial p_s(\bar{b}_s)}{\partial \bar{b}_s} \\ &\quad - k_i \epsilon_i [a_i b_{i,s}^{a_i-1} p_s(\bar{b}_s) + b_{i,s}^{a_i} \frac{\partial p_s(\bar{b}_s)}{\partial \bar{b}_s}] \end{aligned} \quad (3.71)$$

It holds true that  $\frac{\partial \mathbb{E}(u_{i,s})}{\partial b_{i,s}} = \frac{\partial \mathbb{E}(u_{i,s})}{\partial \bar{b}_s}$ , since  $\bar{b}_s = b_{i,s} + b_{-i,s}$ , where  $b_{-i,s} = \sum_{j \in N_s, j \neq i} b_{j,s}$ . Also, for the second term in Equation 3.71 holds true that  $-b_{i,s}^{a_i} h_{i,s}(\bar{b}_s) \frac{\partial p_s(\bar{b}_s)}{\partial \bar{b}_s} < 0$ , and for the last term holds true that  $-k_i [a_i b_{i,s}^{a_i-1} p_s(\bar{b}_s) + b_{i,s}^{a_i} \frac{\partial p_s(\bar{b}_s)}{\partial \bar{b}_s}] < 0$ , since  $\frac{\partial p_s(\bar{b}_s)}{\partial \bar{b}_s} > 0$ .

### Chapter 3. Mobile Edge Computing

As a result, for the root of Equation 3.71, i.e.,  $\frac{\partial \mathbb{E}(u_{i,s})}{\partial b_{i,s}} = 0$ , since the last two terms in Equation 3.71) are negative, it should hold true that:

$$[b_{i,s}^{a_i} \frac{\partial h_{i,s}(\bar{b}_s)}{\partial \bar{b}_s} + a_i b_{i,s}^{a_i-1} h_{i,s}(\bar{b}_s)] > 0 \quad (3.72)$$

as  $(1 - p_s(\bar{b}_s)) > 0$ . The expected prospect-theoretic utility's second derivative is formulated as follows:

$$\begin{aligned} \frac{\partial^2 \mathbb{E}(u_{i,s})}{\partial b_{i,s}^2} &= [b_{i,s}^{a_i} \frac{\partial^2 h_{i,s}(\bar{b}_s)}{\partial \bar{b}_s^2} + 2a_i b_{i,s}^{a_i-1} \frac{\partial h_{i,s}(\bar{b}_s)}{\partial \bar{b}_s}](1 - p_s(\bar{b}_s)) \\ &\quad - 2b_{i,s}^{a_i-1} [b_{i,s} \frac{\partial h_{i,s}(\bar{b}_s)}{\partial \bar{b}_s} + a_i h_{i,s}(\bar{b}_s)] \frac{\partial p_s(\bar{b}_s)}{\partial \bar{b}_s} \\ &\quad - b_{i,s}^{a_i} h_{i,s}(\bar{b}_s) \frac{\partial^2 p_s(\bar{b}_s)}{\partial \bar{b}_s^2} \\ &\quad - k_i \epsilon_i [2a_i b_{i,s}^{a_i-1} \frac{\partial p_s(\bar{b}_s)}{\partial \bar{b}_s} + b_{i,s}^{a_i} \frac{\partial^2 p_s(\bar{b}_s)}{\partial \bar{b}_s^2}] \\ &\quad + a_i(a_i - 1)b_{i,s}^{a_i-2} [h_{i,s}(\bar{b}_s)(1 - p_s(\bar{b}_s)) - k_i \epsilon_i p_s(\bar{b}_s)] \end{aligned} \quad (3.73)$$

Based on Equation 3.73, since  $b_{i,s}$  satisfies Equation 3.72,  $\frac{\partial^2 h_{i,s}(\bar{b}_s)}{\partial \bar{b}_s^2} < 0$ ,  $\frac{\partial h_{i,s}(\bar{b}_s)}{\partial \bar{b}_s} < 0$ ,  $\frac{\partial p_s(\bar{b}_s)}{\partial \bar{b}_s} > 0$  and  $\frac{\partial^2 p_s(\bar{b}_s)}{\partial \bar{b}_s^2} = 0$ , it is true that  $\frac{\partial^2 \mathbb{E}(u_{i,s})}{\partial b_{i,s}^2} < 0, \forall b_{i,s} \in (0, b_{i,s}^{th})$ , thus  $\mathbb{E}(u_{i,s})$  is strictly concave.

In the following Lemma, we prove that  $\mathbb{C}_i = \{\mathbf{b}_i \in \Gamma_i : \mathbf{b}_i \text{ satisfies } (C_i)\}$  is a convex set, due to the fact that the group of constraints  $(C_i)$  is a set of convex functions.

**Lemma 7** *For each user  $i$ , its group of constraints  $(C_i)$  is a set of convex functions.*

$$\begin{aligned} g_i^{(1)} &= \sum_{s \in \mathbb{S}} b_{i,s} - B_i \\ g_{i,s}^{(2)} &= b_{i,s} - b_{i,s}^{th}, \quad \forall s \in \mathbb{S} \\ g_{i,s}^{(3)} &= -b_{i,s}, \quad \forall s \in \mathbb{S} \\ g_i^{(4)} &= \mathbb{E}(O_i)|_t - t_i \\ g_i^{(5)} &= \mathbb{E}(O_i)|_e - e_i \end{aligned} \quad (3.74)$$



### Chapter 3. Mobile Edge Computing

*Proof:* It is obvious that the functions  $g_{i,s}^{(2)}$  and  $g_{i,s}^{(3)}$  are convex functions  $\forall s \in \mathbb{S}$ , as linear functions with respect to  $b_{i,s}$ . Moreover from [234], we know that each function  $f$  of  $n$ -variables, with continuous partial derivatives and cross partial derivatives on a convex set  $S$ , is convex if and only if its Hessian matrix  $H(\mathbf{x})$ , where  $\mathbf{x} \in S$ , is positive semidefinite over the set  $S$ . Also, for each function  $f$  of  $n$ -variables, its Hessian matrix  $H(x)$  is formulated as follows:

$$\begin{bmatrix} f_{11} & f_{12} & \cdots & f_{1n} \\ f_{21} & f_{22} & \cdots & f_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ f_{n1} & f_{n2} & \cdots & f_{nn} \end{bmatrix} \quad (3.75)$$

where  $f_{ij}$  is the cross partial derivative of  $f$ , with respect to its  $j$ -th and its  $i$ -th argument.

Furthermore, from [235] we know that an  $n \times n$  matrix is positive semidefinite, if and only if all its principal minors  $\Delta_k$ ,  $\forall k = 1, \dots, n$ , are non-negative. In our case  $S = \Gamma_i$ , since  $\mathbf{b}_i \in \Gamma_i$ , and  $\Gamma_i$  is a convex set as a cartesian product of convex sets. Moreover, each function in Equation 24 has continuous partial derivatives, and each function's cross partial derivatives exist. For all the functions in Equation 3.74, thus  $\forall n_1 \in \{1, 4, 5\}$  and  $\forall n_2 \in \{2, 3\}$ , it holds true that,  $\frac{\partial g_i^{(n_1)}}{\partial b_{i,s} \partial b_{i,s'}} = 0$ ,  $\frac{\partial g_{i,s}^{(n_2)}}{\partial b_{i,s} \partial b_{i,s'}} = 0$ ,  $\forall s, s' \in \mathbb{S}$ ,  $s \neq s'$ , and  $\frac{\partial^2 g_i^{(n_1)}}{\partial b_{i,s}^2} \neq 0$ ,  $\frac{\partial^2 g_{i,s}^{(n_2)}}{\partial b_{i,s}^2} \neq 0$ . Let us examine the convexity of  $g_i^{(n_1)}$ ,  $\forall n_1 \in \{1, 4, 5\}$  in Equation 3.74

$g_i^{(1)}$ : Based on Equation 3.74 and the definition of an  $n$ -variables function's Hessian Matrix, i.e., Equation 3.75, it is obvious that the Hessian matrix  $H_{g_i^{(1)}} = \mathbf{0}_{S \times S}$ . As a result all the principal minors of the  $H_{g_i^{(1)}}$  are non-negative, thus the function  $g_i^{(1)}$  is convex.

$g_i^{(4)}$ : Based on Equation 3.66, and given that  $p_s(\bar{b}_s) = \frac{\bar{b}_s}{b_s}$  and  $L_i = B_i - \sum_{s \in \mathbb{S}} b_{i,s}$ ,

### Chapter 3. Mobile Edge Computing

the  $g_i^{(4)}$  is given as follows:

$$\begin{aligned}
 g_i^{(4)} = & \sum_{s \in \mathbb{S}} b_{i,s} \left( \frac{1}{R_{i,s}} + \frac{\sum_{j \in N_s} \phi_j}{F_s} \right) \\
 & + \sum_{s \in \mathbb{S}} \frac{\phi_i}{lc_i} \left( \frac{b_{i,s}^2}{\tilde{b}_s} + b_{i,s} \frac{b_{-i,s}}{\tilde{b}_s} \right) \\
 & + \frac{\phi_i}{lc_i} (B_i - \sum_{s \in \mathbb{S}} b_{i,s}) - t_i
 \end{aligned} \tag{3.76}$$

As a result, since  $\frac{\partial g_i^{(4)}}{\partial b_{i,s} \partial b_{i,s'}} = 0$ ,  $\forall s, s' \in \mathbb{S}$  with  $s \neq s'$ , the function's  $g_i^{(4)}$  Hessian matrix is formulated as:

$$H_{g_i^{(4)}} = \begin{bmatrix} \frac{\partial^2 g_i^{(4)}}{\partial b_{i,1}^2} & 0 & \dots & 0 \\ 0 & \frac{\partial^2 g_i^{(4)}}{\partial b_{i,2}^2} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \frac{\partial^2 g_i^{(4)}}{\partial b_{i,S}^2} \end{bmatrix} \tag{3.77}$$

where based on Equation 3.76,  $\frac{\partial^2 g_i^{(4)}}{\partial b_{i,s}^2} = \frac{2\phi_i}{lc_i \tilde{b}_s} > 0$ ,  $\forall s \in \mathbb{S}$ , thus the  $H_{g_i^{(4)}}$  is a diagonal matrix, and all its elements are non negative. Therefore, the  $H_{g_i^{(4)}}$  is positive semidefinite, so the  $g_i^{(4)}$  is a convex function.

$g_i^{(5)}$ : Based on Equation 3.67, and given that  $p_s(\bar{b}_s) = \frac{\bar{b}_s}{b_s}$  and  $L_i = B_i - \sum_{s \in \mathbb{S}} b_{i,s}$ , the  $g_i^{(5)}$  is given as follows:

$$\begin{aligned}
 g_i^{(5)} = & \sum_{s \in \mathbb{S}} \frac{\phi_i}{le_i} \left( \frac{b_{i,s}^2}{\tilde{b}_s} + b_{i,s} \frac{b_{-i,s}}{\tilde{b}_s} \right) \\
 & + \sum_{s \in \mathbb{S}} b_{i,s} \frac{p_{i,s}}{R_{i,s}} + \phi_i le_i (B_i - \sum_{s \in \mathbb{S}} b_{i,s}) - e_i
 \end{aligned} \tag{3.78}$$

As a result,  $\frac{\partial^2 g_i^{(5)}}{\partial b_{i,s}^2} = \frac{2\phi_i}{le_i \tilde{b}_s} > 0$ ,  $\forall s \in \mathbb{S}$ . Therefore, similarly with  $H_{g_i^{(4)}}$ , the  $H_{g_i^{(5)}}$  is a diagonal matrix with non negative elements, therefore it is positive semidefinite, and the function  $g_i^{(5)}$  is convex.

Based on Lemma 7, for each user  $i$ , the set  $\mathbb{C}_i = \Gamma_i \cap (\bigcap_{n_1 \in \{1,4,5\}} Lev(g_i^{(n_1)}, 0)) \cap (\bigcap_{n_2 \in \{2,3\}} Lev(g_i^{(n_2)}, 0))$ ,  $\forall s \in \mathbb{S}$  is a convex set as an intersection of a convex set  $\Gamma_i$

### Chapter 3. Mobile Edge Computing

and level sets of convex functions, which are necessarily convex sets (see Section 3.1.6 of [236]). Therefore, each user's  $i$  strategy space  $\mathbb{A}_i = \Gamma_i \cap \mathbb{C}_i$  in the non-cooperative game  $\mathbb{G}$ , is a convex set as an intersection of convex sets.

**Lemma 8** *Each user's  $i$ ,  $i \in \mathbb{U}$  satisfaction utility  $s_i$ , is a concave function over the strategy space  $\mathbb{A}_i$ .*

*Proof:* From [234], the satisfaction utility  $s_i$  is a concave function if and only if its Hessian matrix  $H_{s_i}$  is negative semidefinite over the convex set  $\mathbb{A}_i$ . Based on Equation 18, since  $\frac{\partial^2 s_i}{\partial b_{i,s} \partial b_{i,s'}} = \frac{\partial^2 \mathbb{E}(u_{i,s'})}{\partial b_{i,s} \partial b_{i,s'}}$ , then  $\frac{\partial^2 s_i}{\partial b_{i,s}^2} = \frac{\partial^2 \mathbb{E}(u_{i,s})}{\partial b_{i,s}^2}$  and it holds true that  $\frac{\partial^2 s_i}{\partial b_{i,s} \partial b_{i,s'}} = 0$ ,  $\forall s, s' \in \mathbb{S}$ , with  $s \neq s'$ , as  $\frac{\partial^2 \mathbb{E}(u_{i,s'})}{\partial b_{i,s} \partial b_{i,s'}} = 0$ . As a result, similarly with the  $H_{g_i^{(4)}}$  and  $H_{g_i^{(5)}}$ , user's  $i$  satisfaction utility's Hessian matrix is given as follows:

$$H_{s_i} = \begin{bmatrix} \frac{\partial^2 \mathbb{E}(u_{i,1})}{\partial b_{i,1}^2} & 0 & \dots & 0 \\ 0 & \frac{\partial^2 \mathbb{E}(u_{i,2})}{\partial b_{i,2}^2} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \frac{\partial^2 \mathbb{E}(u_{i,S})}{\partial b_{i,S}^2} \end{bmatrix} \quad (3.79)$$

In other words, the Hessian matrix  $H_{s_i}$  is a diagonal matrix and from Lemma 2, each element of the matrix is negative. Therefore, for each leading principal minor  $D_k$ ,  $k = 1, \dots, S$  of the  $H_{s_i}$ , it holds true that  $D_k = \prod_{j=1}^k h_{jj}$ , where  $h_{jj} = \frac{\partial^2 \mathbb{E}(u_{i,j})}{\partial b_{i,j}^2}$ . Moreover, for  $k = 2\rho$ ,  $(-1)^k D_k > 0$ , and for  $k = 2\rho+1$  again  $(-1)^k D_k > 0$ , thus based on [234] the Hessian matrix  $H_{s_i}$  is strictly negative definite. Thus, the satisfaction utility  $s_i$  is concave over the convex set  $\mathbb{A}_i$ .

Based on Lemmas 5-8, each user's  $i$  strategy space  $\mathbb{A}_i$  is a convex set, and its satisfaction utility  $s_i(\mathbf{b}_i, \mathbf{b}_{-i})$  is concave over the set  $\mathbb{A}_i$ . Thus, the non-cooperative game  $\mathbb{G}$  is an  $n$ -person concave game, where  $n = U$ , so the Theorem 1 holds true. An  $n$ -person concave game has at least one Pure Nash Equilibrium (PNE) [237], thus the existence of at least one PNE point for the non-cooperative game  $\mathbb{G}$  holds true.

Finally, based on Theorem 10, Lemma 8 and [237], the following Theorem proves the convergence of the users' strategies to the PNE.

**Theorem 11** *Consider the user  $i$  and an  $S \times S$  matrix function  $\mathbb{X}_i$  in which  $(\mathbb{X}_i)_{ss'} = \lambda_i \frac{\partial^2 s_i}{\partial s \partial s'}$ ,  $\forall s, s' \in \mathbb{S}$ , and the constant choices  $\lambda_i > 0$ . Then, if  $\mathbb{X}_i + \mathbb{X}_i^T$  is strictly negative definite, then the PNE of the game  $\mathbb{G}$  is unique. Starting from any initial offloading strategy  $\mathbf{b}_0 = [\mathbf{b}_{1,0}, \dots, \mathbf{b}_{U,0}]$ ,  $\mathbf{b}_{i,0} \in \mathbb{A}_i$ , the continuous Best Response (BR) dynamics converge to the unique PNE.*

*Proof:* As we mentioned in Lemma 8, since  $\frac{\partial^2 s_i}{\partial b_{i,s} \partial b_{i,s'}} = \frac{\partial^2 \mathbb{E}(u_{i,s'})}{\partial b_{i,s} \partial b_{i,s'}}$ , and  $\frac{\partial^2 s_i}{\partial b_{i,s} \partial b_{i,s'}} = 0$ ,  $\forall s, s' \in \mathbb{S}$ , with  $s \neq s'$ , as  $\frac{\partial^2 \mathbb{E}(u_{i,s'})}{\partial b_{i,s} \partial b_{i,s'}} = 0$ , the matrix  $\mathbb{X}_i$  is given as follows:

$$\mathbb{X}_i = \begin{bmatrix} \lambda_i \frac{\partial^2 \mathbb{E}(u_{i,1})}{\partial b_{i,1}^2} & 0 & \dots & 0 \\ 0 & \lambda_i \frac{\partial^2 \mathbb{E}(u_{i,2})}{\partial b_{i,2}^2} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \lambda_i \frac{\partial^2 \mathbb{E}(u_{i,S})}{\partial b_{i,S}^2} \end{bmatrix} \quad (3.80)$$

thus,  $\mathbb{X}_i = \lambda_i \cdot H_{s_i}$ . Also, due to the fact that  $\mathbb{X}_i$  is diagonal it holds true that  $\mathbb{X}_i = \mathbb{X}_i^T$ , so  $\mathbb{X}_i + \mathbb{X}_i^T = 2 \cdot \mathbb{X}_i$ . Furthermore, in Lemma 4 we proved that for each user  $i$ ,  $i \in \mathbb{U}$ , its satisfaction utility's Hessian matrix  $H_{s_i}$  is strictly negative definite, so since  $\lambda_i > 0$  and  $\mathbb{X}_i + \mathbb{X}_i^T = 2 \cdot \mathbb{X}_i = 2\lambda_i H_{s_i}$ , the  $\mathbb{X}_i + \mathbb{X}_i^T$  matrix is strictly negative definite. Therefore, the non-cooperative game  $\mathbb{G}$  has a unique NE point, and starting from any initial point  $\mathbf{b}_0$  the BR dynamics converge to the unique PNE point.

### 3.6.4 Towards Determining the Equilibrium

#### A Convex Optimization Approach

Each user's  $i$  satisfaction utility  $s_i$  is a concave function over  $\mathbb{A}_i$  (Lemma 8), thus the function  $z_i(\mathbf{b}_i, \mathbf{b}_{-i}) = -s_i(\mathbf{b}_i, \mathbf{b}_{-i})$  is a convex function over the same space. Let us denote each user's  $i$  best response strategy  $\mathbf{b}_i^*(\mathbf{b}_{-i}) : \mathbb{A}_{-i} \rightrightarrows \mathbb{A}_i$  considering the other

### Chapter 3. Mobile Edge Computing

users' strategies, as follows:

$$\mathbf{b}_i^*(\mathbf{b}_{-i}) = \arg \max_{\mathbf{b}_i \in \mathbb{A}_i} (s_i(\mathbf{b}_i, \mathbf{b}_{-i})), \mathbf{b}_{-i} \in \mathbb{A}_{-i} \quad (3.81)$$

where  $\mathbf{b}_{-i}$  is the vector of the offloading strategies of all the users except user  $i$ , as it was defined in Section IV, and  $\mathbb{A}_{-i} = \mathbb{A}_1 \times \cdots \times \mathbb{A}_{i-1} \times \mathbb{A}_{i+1} \times \cdots \times \mathbb{A}_U$  is the corresponding strategy space, thus  $\forall i \in \mathbb{U}$ ,  $\mathbf{b}_{-i} \in \mathbb{A}_{-i}$ . Each user's  $i$  best response strategy  $\mathbf{b}_i \in \mathbb{A}_i$  should satisfy the group of constraints  $(C_i)$  (Equation 3.70). Furthermore, considering that the function  $z_i$  is a convex function over the convex set  $\mathbb{A}_i$ , each user's  $i$  best response strategy can be formulated as follows:

$$\mathbf{b}_i^*(\mathbf{b}_{-i}) = \arg \min_{\mathbf{b}_i \in \mathbb{A}_i} (z_i(\mathbf{b}_i, \mathbf{b}_{-i})), \mathbf{b}_{-i} \in \mathbb{A}_{-i} \quad (3.82)$$

Each user  $i$  in order to maximize its satisfaction utility  $s_i$  (Equation 3.64), should equivalently minimize the convex function  $z_i$  over the convex set  $\mathbb{A}_i$ . Thus, each user  $i$  during the continues BR dynamics solves the following optimization problem to determine its best response strategy  $\mathbf{b}_i^*$ .

$$\left. \begin{array}{ll} \underset{\mathbf{b}_i \in \Gamma_i}{\text{minimize}} & z_i(\mathbf{b}_i, \mathbf{b}_{-i}) \\ \text{subject to} & \left. \begin{array}{l} \sum_{s \in \mathbb{S}} b_{i,s} \leq B_i, \\ 0 \leq b_{i,s} \leq b_{i,s}^{th}, \forall s \in \mathbb{S}, \\ \mathbb{E}(O_i)|_t \leq t_i, \\ \mathbb{E}(O_i)|_e \leq e_i \end{array} \right\} (C_i) \end{array} \right\} \quad (3.83)$$

Moreover, assuming that each user  $i$  is able to satisfy its time and energy constraint in  $(C_i)$  by executing its whole application locally, i.e.,  $\mathbf{b}_i = \mathbf{0} \in \mathbb{A}_i$ , the set  $\mathbb{A}_i$  is non-empty, and the above minimization problem is a nonlinear convex optimization problem, where the function  $z_i(\mathbf{b}_i, \mathbf{b}_{-i})$  is the objective function, and the  $g_i^{(n_1)}$ ,  $n_1 \in \{1, 4, 5\}$ ,  $g_{i,s}^{(n_2)}$ ,  $n_2 \in \{2, 3\}$  (Equation 3.74) are the inequality constraints.

### Algorithm and Complexity Analysis

In this section, the Distributed Algorithm for Convergence to the PNE (DACP) of the non-cooperative game  $\mathbb{G}$  is presented. The DACP algorithm is a decision-making tool that runs at the beginning of the data offloading process and after it converges to  $\mathbf{b}^* = [\mathbf{b}_1^*, \dots, \mathbf{b}_1^*, \dots, \mathbf{b}_U^*]$  the users know the data that should be offloaded to each MEC server and the ones that should be processed locally. The DACP algorithm is an iterative distributed sequential algorithm, where at each iteration only one randomly selected user plays an action. At the first iteration ( $ite = 0$ ), each user selects randomly a feasible data offloading vector  $\mathbf{b}_i^*, \forall i \in U$ . Then, this is reported to the MEC servers by a user's broadcasting signal and each MEC server calculates the  $\bar{b}_s, \forall s \in S$ , which then is broadcasted to all the users. At the next iteration of the DACP algorithm, one user is randomly selected to make an action  $\mathbf{b}_i^*$  given the values  $\bar{b}_s, \forall s \in S$ . The user makes action and broadcasts its decision to all the MEC servers in order for the latter to recalculate the new values  $\bar{b}_s, \forall s \in S$ . The same procedure is followed iteratively until the DACP algorithm converges (Line 15 of the DACP algorithm). After the DACP algorithm converges, then each user has decided its data offloading vector  $\mathbf{b}_i^*$  and performs the data offloading.

Specifically, each user, in order to compute its best response  $\mathbf{b}_i^*$ , first receives from each MEC server the total amount of offloaded data  $\bar{b}_s$ , the interference  $\sum_{j \in N_s} p_{j,s} g_{j,s}$ , and the overall applications' levels of computing intensity  $\sum_{j \in N_s, j \neq i} \phi_j$  that have been offloaded to this MEC server  $s$ . Then, each user, in order to construct its second constraint in  $(C_i)$ , determines the  $b_{i,s}^{th}$ , such that  $ert_{i,s}(b_{i,s}^{th}) = 0, \forall s \in S$ . From Lemma 1 the root  $r_{i,s}^*$  of the  $ert_{i,s} = 0$  exists, and given that the  $ert_{i,s}$  is strictly decreasing, the  $r_{i,s}^*$  is unique and can be found via Binary Search into  $[0, \tilde{b}_s]$  with an approximation error  $\epsilon \rightarrow 0$ , thus  $b_{i,s}^{th} = \min(r_{i,s}^*, B_i)$ .

Each user has to solve the nonlinear optimization problem given in Equation 3.83 to determine its best response strategy. Since, as we have already proven, the problem in Equation 3.83 is a convex optimization problem, the constrained local minimum

is also a constrained global minimum. Thus, each user may apply any of well known existing methods for solving constrained nonlinear optimization problems [238], and conclude to the global minimum of  $z_i(\mathbf{b}_i, \mathbf{b}_{-i})$  (Equation 3.83), while determining its best response strategy  $\mathbf{b}_i^*$ . For demonstration purposes, we consider the sequential quadratic programming (SQP) [239] method.

Regarding the Algorithm's DACP complexity, each user  $i$ , is required to determine the  $b_{i,s}^{th}$ ,  $\forall s \in \mathbb{S}$ . Given that the complexity of the Binary Search into the interval  $[0, \tilde{b}_s]$ ,  $s \in \mathbb{S}$ , is  $\mathcal{O}(\log_2 \tilde{b}_s)$  [229], the complexity of the user  $i$  to determine all the  $b_{i,s}^{th}$  is  $\mathcal{O}(S \cdot \log_2(\max_{s \in \mathbb{S}}(\tilde{b}_s)))$ . Also, by denoting as  $\mathcal{O}(\Delta)$  the complexity of solving the convex the optimization problem through sequential quadratic programming, and since the rest operations involve only algebraic calculations, the complexity of each user  $i$  to determine its best response  $\mathbf{b}_i^*$  at each iteration  $ite$  is  $\mathcal{O}(\Delta + S \cdot \log_2(\max_{s \in \mathbb{S}}(\tilde{b}_s)))$ . Considering that  $U$  users execute the Algorithm DACP and given that  $Ite$  iterations are needed for convergence to the PNE, the total complexity of the distributed Algorithm DACP for all the users is  $\mathcal{O}(U \cdot Itc \cdot (\Delta + S \cdot \log_2(\max_{s \in \mathbb{S}}(\tilde{b}_s))))$ . Finally, the complexity of the optimization problem  $\mathcal{O}(\Delta)$  can be considered significantly greater than the complexity  $\mathcal{O}(S \cdot \log_2(\max_{s \in \mathbb{S}}(\tilde{b}_s)))$ , therefore the complexity of the Algorithm DACP is  $\mathcal{O}(U \cdot Itc \cdot \Delta)$ .

In a nutshell, the DACP algorithm is a decision-making tool enabling the users to determine their optimal data offloading satisfying their personal constraints ( $C_i$ ), as presented in Equation 3.83, before they actually perform it. Also the DACP algorithm needs only few iterations (i.e., less than five iterations) in order to converge, thus the signaling overhead added to the end-users is rather limited, and in most cases practically insignificant.

### 3.6.5 Empirical Evaluation

In this section, we provide a detailed numerical performance evaluation of the proposed prospect-theoretic framework, through modeling and simulation, illustrating

---

**Algorithm 3** DACP: Distributed Algorithm for Convergence to PNE

---

```

1: Input:
2: Set of users:  $\mathbb{U} = [1, \dots, i, \dots, U]$ 
3: Set of MEC Servers:  $\mathbb{S} = [1, \dots, s, \dots, S]$ 
4: Output:
5: Profile Strategy at PNE:  $\mathbf{b}^* = [\mathbf{b}_1^*, \dots, \mathbf{b}_i^*, \dots, \mathbf{b}_U^*]$ 
6: Initialization:
7:  $\mathbf{b}_i = [b_{i,1}, \dots, b_{i,s}, \dots, b_{i,S}]$ 
8:  $ite = 0, Convergence = 0$ 
9: Iterative Procedure:
10: while  $Convergence == 0$  do
11:    $ite = ite + 1;$ 
12:    $flag = 0;$ 
13:   for  $i = 1$  to  $U$  do
14:     for  $s = 1$  to  $S$  do
15:       user  $i$  calculates the transmission uplink rate  $R_{i,s}$ 
16:        $r_{i,s}^* = BinarySearch([0, \tilde{b}_s], \epsilon);$ 
17:        $b_{i,s}^{th} = min(r_{i,s}^*, B_i);$ 
18:     end for
19:      $\mathbf{b}_i^* = fmincon();$ 
20:     if  $(|b_{i,s}^* - b_{i,s}| \leq \epsilon', \forall s \in \mathbb{S})$  then
21:        $flag = flag + 1;$ 
22:     end if
23:      $\mathbf{b}_i = \mathbf{b}_i^*$ 
24:   end for
25:   if  $(flag == U)$  then
26:      $Convergence = 1, Ite = ite;$ 
27:   end if
28: end while

```

---

the operation, features, and benefits of our approach. We focus on the pure operational characteristics of our prospect-theoretic framework, in terms of efficiently



controlling the users' offloaded data with respect to the heterogeneous multiple MEC server environment. Furthermore, we provide a detailed study of our framework's operation under heterogeneous users regarding their loss-aversion characteristics, and a scalability and fragility evaluation study is shown concerning an increasing number of users and MEC servers. Finally, a comparative evaluation of our approach against alternative approaches and offloading strategies is provided.

In our study, we consider a set of  $S = 3$  heterogeneous MEC servers, with each MEC server  $s$ ,  $s \in \mathbb{S}$  having a coverage area of radius  $R_s = 100m$ , and  $U = 50$  users in total. Each user's  $i$ ,  $i \in \mathbb{U}$  channel gain is modeled as  $g_{i,s} = \frac{1}{d_{i,s}^\theta}$ , where  $d_{i,s}$  is the user's  $i$  distance from the MEC server  $s$ , i.e.,  $d_{i,s} \leq R_s$ , and  $\theta$  is the distance loss exponent, e.g.,  $\theta = 2$ . The system's transmission bandwidth is considered  $W = 5MHz$ , while a representative value of the service uplink rate for video conference application is  $R_{fix} = 128$  kbps. Each user  $i$  transmits to the MEC server  $s$  with power  $p_{i,s} = \frac{d_{i,s}^2}{R_s^2}$ , thus each user's  $i$  transmission power is normalized and proportional to its distance from the corresponding MEC server.

Moreover, for each user  $i$  we consider  $lc_i \in [0.1, 1]$  GHz and  $le_i = 10^{-9} \frac{Joules}{CPU-Cycle}$ ,  $\forall i \in \mathbb{U}$  [230]. The considered application characteristics (e.g., face recognition application) are  $B_i \in [1000, 5000]$  KB and  $C_i \in [1000, 5000]$  Mega-Cycles. In the following, unless otherwise explicitly stated, we assume homogeneous users with prospect-theoretic parameters  $\alpha_i = 0.2$  and  $k_i = 5$ ,  $\forall i \in \mathbb{U}$ . Finally, for each MEC server  $s$ ,  $s \in \mathbb{S}$  we consider that  $F_s \in [1, 4] \cdot 10^3$  GHz and  $\tilde{b}_s \in [30, 70]\% \cdot \sum_{i=1}^{50} B_i$ .

### Pure Operation of the Algorithm

Figure 3.18a presents the evolution of a specific user's offloading strategy ( $b_{i,s}$ ,  $\forall s \in \mathbb{S}$ ) at each MEC server  $s$ , as a function of the number of iterations and actual execution time needed for the Algorithm DACP to converge at the PNE point. Firstly, we observe that the user by starting from randomly selected feasible initial values, as the amounts of the data offloading at each MEC server, converges in few iterations,

### Chapter 3. Mobile Edge Computing

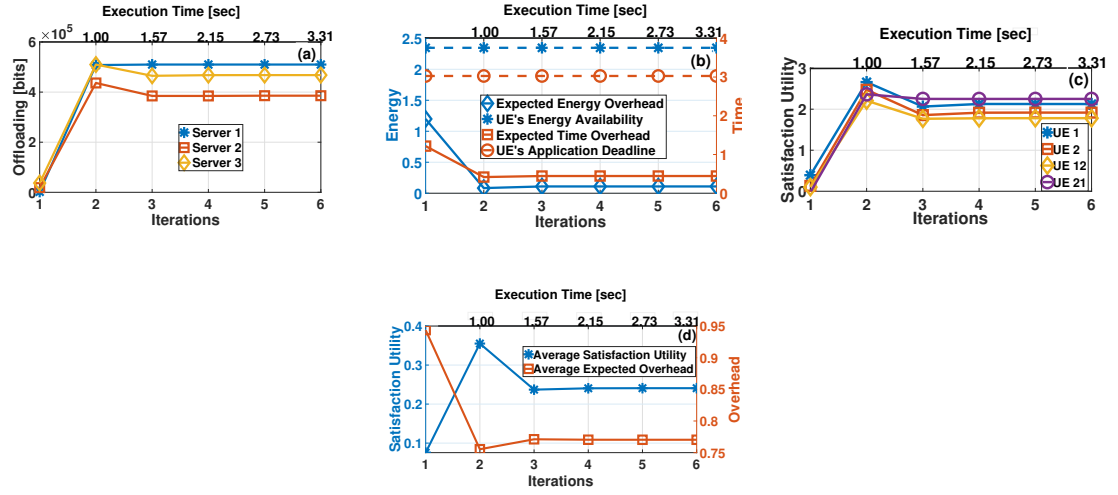


Figure 3.18: Pure operation of the proposed framework - Users' Perspective

i.e., less than five, at the unique PNE point. Indicatively we note that the DACP algorithm needs approximately 1.5sec to converge to the Pure Nash Equilibrium considering that 3 MEC servers and 50 users reside in the network, while significantly smaller times are observed if smaller-scale systems are considered or enhanced devices are utilized. Also, in our simulated scenarios, we have considered an indicative

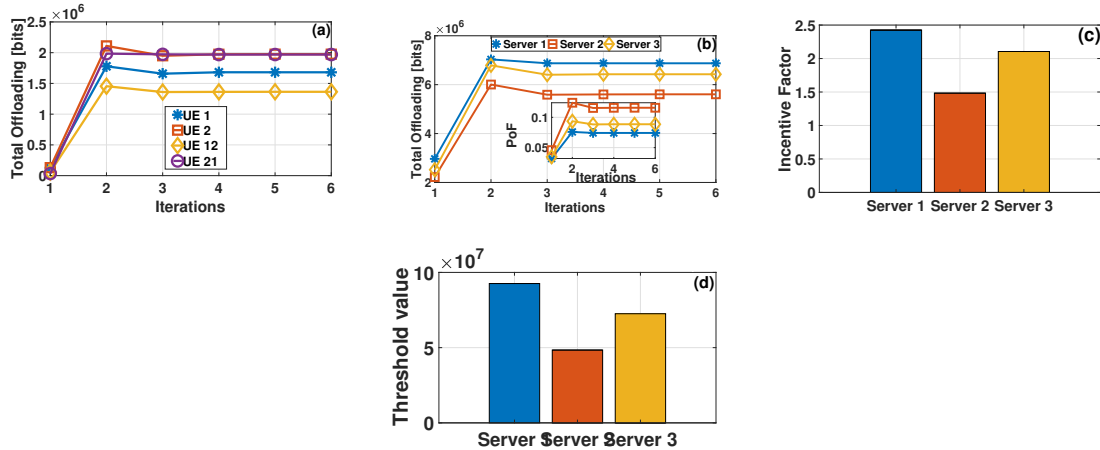


Figure 3.19: Pure operation of the proposed framework - MEC Servers' Perspective

application with latency constraints  $3sec$  (Figure 3.18b) and based on the decision of the DACP algorithm a time overhead  $\mathbb{E}(O_i)|_t = 50msec$  is achieved for the execution of the application.

As we also see in Figure 3.19a, where four different users are considered, each user's total amount of offloaded data converges to a stable point, while the difference in the values of these points is due to the users' heterogeneous characteristics, e.g., users' application characteristics, users' location inside the system. Furthermore, as Figure 3.18b illustrates, the examined user determines its best response strategy (in accordance to Equation 3.82) by satisfying its energy and time constraint at every iteration, while at the same time its satisfaction utility converges to a stable point (Figure 3.18c), as the user's data offloading strategy at each MEC server converges (Figure 3.18a). Also, the propagation time is negligible in our presented numerical results, as the maximum distance of each user from each MEC server is  $100m$ . Moreover, the convergence of the users' average satisfaction utility and overhead are presented in Figure 3.18d.

Figure 3.19b presents the total amount of offloaded data that each MEC server collects by the users. MEC servers' heterogeneous characteristics, in terms of incentivizing the users to offload part of their applications to the MEC servers, are better captured by the incentive factor  $\frac{\frac{\tilde{b}_s}{\sum_{j \in \mathcal{S}} \tilde{b}_j} + \frac{F_s}{\sum_{j \in \mathcal{S}} F_j}}{\frac{\sum_{i \in \mathcal{U}} d_{i,s}}{\sum_{j \in \mathcal{S}} \sum_{i \in \mathcal{U}} d_{i,j}}}$  of each MEC server, which is presented in Figure 3.19c. The MEC server's incentive factor indicates that the higher the ability of a MEC server to process bigger amounts of data, i.e.,  $\tilde{b}_s$ , or the higher the MEC server's computation capability, i.e.,  $F_s$ , then the higher is the incentive of a user to offload part of its application to this MEC server to obtain an increased satisfaction utility. Consequently, the higher is the MEC server's incentive factor, the greater is the amount of data that it gathers by the users (Figure 3.19b). Also note, that each MEC server's incentive factor is being influenced by the average distance of the users from the server since for small distances the users will experience less communication overhead during their data offloading at this MEC server. Al-

though MEC servers 1 and 3 collect a higher amount of data compared to server 2, the latter one concludes to a higher probability of failure (sub-figure within Figure 3.19b). This phenomenon is observed since MEC servers 1 and 3 are assumed to have a significantly higher threshold value  $\tilde{b}_s$  than MEC server 2 (Figure 3.19d), which enables them to process a higher amount of data.

In the following, we present some indicative results to study the tradeoffs in the users' offloading decisions concerning the MEC servers' characteristics, i.e., threshold value  $\tilde{b}_s$ , computation capability  $F_s$ , and the average distance of the users from each MEC server. In particular, we assume a scenario where each user initially has the same distance from each MEC server, while the MEC server 3 has improved computation capabilities compared to the rest of the servers, in terms of its threshold value  $\tilde{b}_s$  and its computation capability  $F_s$ . Then, the distance of the MEC server 3 from each user increases, thus, the users reduce their amount of data that they offload to the MEC server 3 (Figure 3.20a). Specifically, the MEC server 3, due to its improved computation capability gathered a greater amount of data at the initial point, while then the amount of collected data decreases, as each user experiences a greater communication overhead due to the increase of its transmission power and time, which overturns the obtained computation benefit. Moreover, the users to reduce their additional local amount of data due to the decrease of their offloaded amount of data at the MEC server 3, they increase their corresponding amounts to the rest of MEC servers, and this leads the MEC servers 1 & 2 to receive a greater amount of data compared to the MEC server 3 (Figure 3.20a), after a specific point. As a result, the values of the probabilities of failure for the MEC servers 1 & 2 increase, while MEC server's 3 corresponding value decreases (Figure 3.20b). Furthermore, due to the increase of the MEC server's 3 distance from the users, each user's offloading is becoming less beneficial, and as a result, its satisfaction utility decreases as it experiences a greater expected overhead (Figure 3.20c).

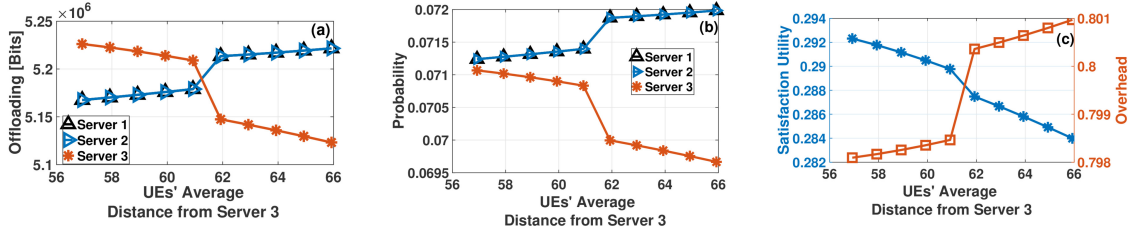


Figure 3.20: Computation vs Communication Overhead

### Scalability & Fragility Evaluation

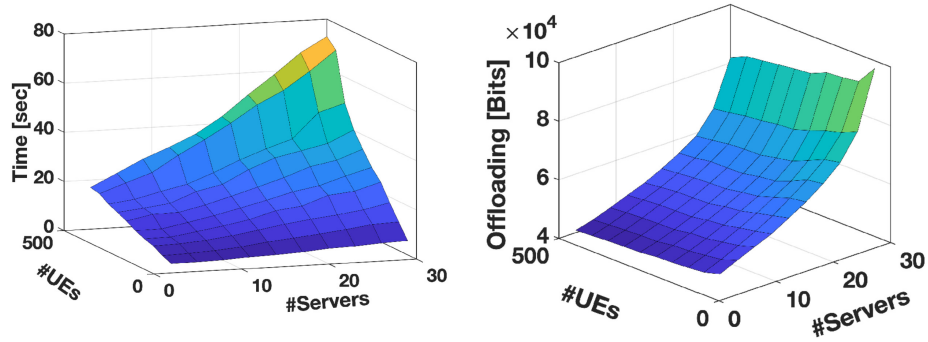


Figure 3.21: Time & Offloaded Data - Scalability Evaluation

Figure 3.21 illustrates the necessary time for convergence to PNE (left-most sub-figure), both for an increasing number of users and an increasing number of MEC servers. It is observed that our prospect-theoretic framework scales very well with respect to the increasing number of MEC servers, since the required execution time

has a smaller increase rate when compared to the corresponding increase rate on the number of servers. Moreover, as the number of the users increases, the framework's execution time follows almost a linear increasing trend with respect to the number of users, and this indicates that the factor  $\Delta \cdot Ite$  in our framework's complexity  $\mathcal{O}(U \cdot Ite \cdot \Delta)$  increases with a significantly lower rate compared to the increase of the factor  $U$ . It should be clarified that the DACP algorithm is a decision-making tool enabling the users to determine their optimal data offloading satisfying their personal constraints ( $C_i$ ), as presented in Equation 3.83, before they actually perform it.

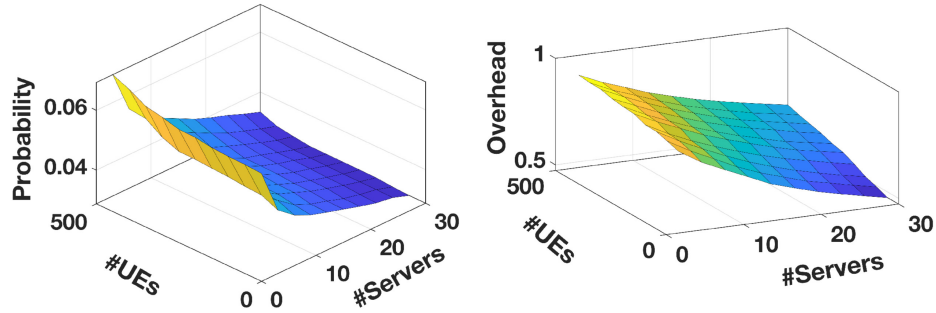


Figure 3.22: Probability & Overhead - Scalability Evaluation

Furthermore, in Figure 3.22 our framework's performance in terms of the MEC servers' probability of failure, and users' experienced expected overhead is studied. The results reveal that by keeping the number of users constant and increasing the number of MEC servers, the performance of the system improves since the average amount of data that each MEC server receives from the users is reduced (Figure 3.21b), as the users have more choices/MEC servers to offload their data. Thus, the average probability of the MEC servers decreases (Figure 3.22 left-most figure) ( $p_s$  is decreasing with respect to  $\bar{b}_s$ ), and the users experience lower expected overhead

(Figure 3.22 right-most figure).

On the other hand, by keeping the number of MEC servers constant and increasing the number of users, the exact opposite phenomenon is observed. In particular, the MEC servers become more congested as the servers' average received amount of data increases (Figure 3.21 right-most figure), and as a result the average probability of failure of the MEC servers also increases (Figure 3.22 left-most figure). Moreover, since each MEC server is overloaded, the computation capability portion that each user obtains ( $F_{i,s}$ , Equation 3.50) from each MEC server decreases, while at the same time the communication overhead increases. As a result, each user experiences a greater expected overhead (Figure 3.22 right-most figure).

In order to further study the effect of competition on the fragility of each MEC server (i.e., treated as CPR) between a single and several self-interested users, we use the Fragility under Competition (FuC) metric, which is defined as the ratio of the fragility of a MEC server when there are several users to the fragility of the server when there is only one user [130]. The fragility of the MEC server  $s$  is expressed by the failure probability function,  $p_s(\bar{b}_s)$  which steadily increases as users' total offloaded bits (i.e., investment)  $\bar{b}_s$  increases. The fragility of MEC server  $s$  is expressed by the failure probability function, which steadily increases as users' total investment increases. Specifically, the Fragility under Competition for each MEC server  $s$  is given by  $FuC_s = \frac{p_s(\bar{b}_s^*)}{p_s(b_{s,i}^*)}$ , where the numerator  $p_s(\bar{b}_s^*)$  is the probability of failure function when the total investment in the server  $s$  at the Pure Nash Equilibrium (PNE) point of  $N$ ,  $N \geq 2$  homogeneous visitors is  $\bar{b}_s^*$ , whereas the denominator  $p_s(b_{s,i}^*)$  is the probability of failure function when considering a single user  $i$  (i.e.,  $N = 1$ ) who has the same risk preferences as the homogeneous group and its optimal investment in CPR is  $b_{i,s}^*$ .

Figure 3.23 depicts the MEC server average FuC value as a function of increasing number of users, for different values of sensitivity parameter  $a$ . In particular, we observe that the FuC value of a MEC server rises as the number of users grows,

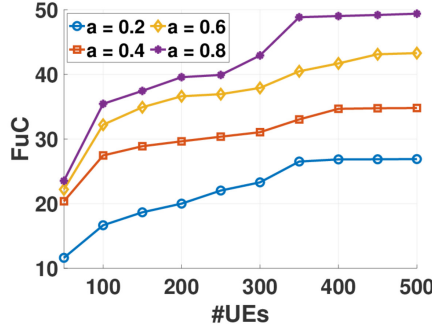


Figure 3.23: Fragility under Competition

then depending on the sensitivity parameter  $a$ , it reaches a peak and after that remains stable, regardless of the number of users. Based on Theorem 1 and Lemma 1 the total investment in a MEC server  $s$  at the PNE is smaller than  $b_{i,s}^{\text{th}}$ , while Assumption 1 states that probability of failure is an increasing function of  $\bar{b}_s$ , and thus  $p_s(\bar{b}_s) < p_s(b_{i,s}^{\text{th}})$ . As a consequence, FuC is upper bounded which is confirmed by our numerical evaluation results. Figure 9 also illustrates that the FuC bound decreases when visitors have a smaller sensitivity parameter  $a$ , thus they become more risk-averse. This confirms that the bounds are influenced by the sensitivity parameter and the specific CPR characteristics.

### Comparative Evaluation

Considering the basic scenario of homogeneous users, a comprehensive comparative study of the proposed optimal approach, against several other alternatives is presented. The comparative evaluation is performed concerning the following metrics: users' achievable average expected overhead and MEC servers' probability of failure. We compare our approach to six other approaches that differ with respect to the users' behaviors, as follows:



- (a) non-prospect-theoretic (NPT) users, but expected overhead minimizers instead. Each user  $i$  determines its best response  $\mathbf{b}_i^*$  that minimizes its perceived expected overhead (Equation 3.59),
- (b) a full game-theoretic offloading approach (only non-partial offloading is permitted) (GOFF) [175], where a non-cooperative game is formulated among the users. Each user  $i$  determines its best choice  $ch_i^* \in \{0, 1, \dots, S\}$ , in terms of which MEC server to select to offload its whole application (data), ( $ch_i = 0$ , if the user  $i$  keeps its application for local execution), that minimizes its perceived expected overhead (Equation 3.60),
- (c) a single MEC server environment (SMEC), where a single only MEC server with the average capabilities of the three MEC servers of the basic setting is placed, instead of a *multi-MEC servers* environment,
- (d) only local (LOC) computing users, who are risk-averse and keep the task's execution locally to obtain the guaranteed, though limited, the performance provided by their device,
- (e) full offloading users (FULL), who are risk-seeking, and offload their whole task to the *multi-MEC servers* environment, by choosing randomly a MEC server  $s$ , thus  $\mathbf{b}_i^* = B_i$ ,
- (f) each user  $i$  determines its best response  $\mathbf{b}_i^* = [b_{i,1}^*, \dots, b_{i,S}^*]$  randomly (RAND), such that  $\sum_{s \in \mathbb{S}} b_{i,s} \leq B_i$ .

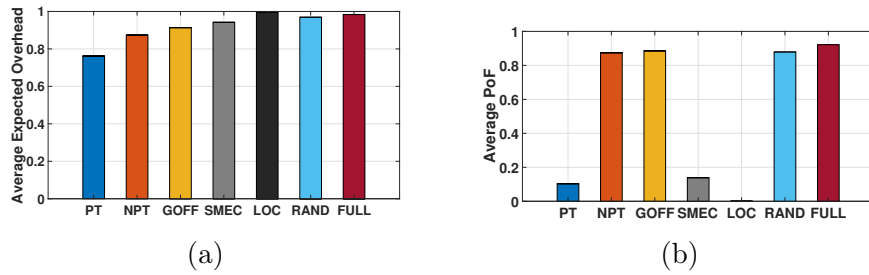


Figure 3.24: Comparative Evaluation

Specifically, Figure 3.24a illustrates the users' average experienced overhead, and

Figure 3.24b indicates the MEC servers' average probability of failure for each different approach. The results reveal that our proposed approach achieves the best performance in terms of both experienced overhead and probability of failure, while the LOC, RAND, and FULL alternatives achieve the worst performance. In particular, in the LOC approach, the users perceive the highest expected overhead, since they keep the whole application for local execution, and thus they obtain the worst performance in terms of time and energy overhead due to the limited local computing characteristics of the devices. On the other hand, under the RAND and FULL approaches which offload either part (RAND) or the whole application (FULL), respectively, the users experience lower overhead. However, under the FULL approach, the MEC servers become overloaded, and thus the highest average probability of failure is observed.

Furthermore, as Figure 3.24a presents, the NPT approach achieves the second-best performance after our approach. This is due to the pure benefits stemming from the optimization of the partial offloading, while under the GOFF approach the users offload their whole application without taking advantage of the potential for partial offloading. This leads the MEC servers to higher levels of congestion, with a higher probability of failure (Figure 3.24b), and as a result since the uncertainty of the MEC servers' successful operation increases, it is expected that the users will execute greater amounts of data locally, and the expected overhead (Equation 3.60) increases accordingly.

On the other hand, under the NPT approach, the users make their offloading decisions to simply minimize their perceived expected overhead, without evaluating however their perceived overhead regarding the guaranteed performance that they would obtain if executed the offloading amount of data locally. Consequently, the MEC servers conclude to significantly higher probability of failure (Figure 3.24b), while the users obtain the worst performance compared to our prospect-theoretic approach, where the users' decisions offloading strategies are based on the tradeoff

between the perceived performance and the one that they would experience from their local device (Figure 3.24a). Finally, the SMEC prospect-theoretic strategy results in a relatively good performance in terms of MEC servers' average probability of failure owing to the consideration of the risk-based behavior modeling, however, since there is only one single MEC server, the users enjoy limited computation capabilities, while the communication overhead increases and the MEC server's computation capability is shared among all the users. As a result, the SMEC strategy results in relatively higher expected overhead compared to the NPT and GOFF.

### **3.6.6 Summary**

A novel approach towards establishing each user's optimal data offloading decision-making process within a multiple MEC servers environment is introduced. The users' risk-aware attitudes are considered due to the computing uncertainty imposed by the multi-MEC system. The users can offload part of their computing tasks to the MEC servers and execute the rest locally. Each MEC server is considered as a Common Pool of Resources (Chapter 2.3), serving the users' computing requests, and can potentially fail due to over-exploitation. The users demonstrate different risk-aware data offloading behaviors, which are captured in a holistic prospect-theoretic utility function, following the principles of Prospect Theory. The goal of each user is to maximize its perceived satisfaction, as expressed by the prospect-theoretic utility function, by offloading its computing tasks to the MEC servers.

A non-cooperative game among the users is formulated and the corresponding Pure Nash Equilibrium is determined, while a distributed low-complexity decision-making algorithm that converges to the PNE is also introduced. Detailed numerical results were presented highlighting the operation and superiority of the proposed framework with respect to the users' resource-constrained computing systems' perceived overall overhead.

## Chapter 4

# Redesigning Resource Management in Wireless Networks

The rise in popularity of smartphones, along with the need for personalized services with different Quality of Service (QoS) requirements, has created an increased interest in energy-efficient resource management frameworks in wireless networks, where user actions and decisions are interdependent. Our focus is placed on the transformation and treatment of the uplink power control problem under the perspective of game theory in satisfaction form. The novel concept of Minimum Efficient Satisfaction Equilibrium (MESE) is introduced and its properties are thoroughly investigated. In particular, considering that each user is associated with a cost function concerning its actions, the MESE point defines each user's transmission power that satisfies its QoS prerequisites with the lowest cost (Chapter 2.1.2). We prove that at the MESE point, not only the system achieves the lowest possible cumulative cost, but also each user individually is penalized with the minimum cost compared to the corresponding cost of any Efficient Satisfaction Equilibrium (ESE) point. The existence, uniqueness, and benefits of the MESE are thoroughly studied, while a distributed and low complexity algorithm based on the Best Response Dynamics that converges to the MESE point is proposed. Through modeling and simulation,

the performance and inherent attributes of the proposed novel resource management framework are evaluated, and its benefits are revealed.

## 4.1 Related Work

The massive expansion of mobile devices and traffic is creating competitive communication environments that induce constraints regarding the availability of their resources, thus making the problem of resource management even more challenging and demanding. In such competitive and distributed communications systems, users evolve with others, while their decisions and actions are interdependent. Thus, given this setting and responding to the need for distributed solutions, Game Theory (Chapter 2.1) arises as a natural choice and a powerful tool to cope with users' selfish and competitive behavior regarding the resource orchestration process within the emerging 5G networks.

The majority of the existing literature, regarding resource management in wireless networks, focuses on the *Expected Utility Maximization* problems, where the users aim at maximizing their utility in a selfish manner targeting the highest possible performance [240]. Various resource management problems have been considered in the recent literature, dealing with: (1) a single resource to be allocated to the users, e.g., power control [241, 242], rate control [243], (2) the allocation of multiple resources, e.g., power and rate control [244] or sub-channel, rate, and power allocation [245, 246]. In those approaches, the non-cooperative game theory is adopted to formulate the resource management problems and their solutions conclude to Nash equilibrium points which are stable operational points for all the users in the network.

However, it is well-known that the Nash equilibrium points stemming from users' selfish decision-making are generally inefficient. Thus, a first step towards addressing this challenge and guide the selfish users towards a more efficient operating point was the introduction of the pricing mechanisms, where the users are penalized concerning

their resources' consumption [247, 248]. Despite the relative improvements that have been obtained by adopting the concept of resource pricing, still, those approaches could not address the main disadvantage of the Nash equilibrium points holistically. In particular, customized heuristic pricing mechanisms are required each time to treat different resource types and address different networking environments. Furthermore, even when pricing is considered, in principle each user still aims at maximizing his own perceived Quality of Service (QoS). Thus the realization of the aforementioned maximization goal ultimately does not offer a notifying difference to the experienced satisfaction.

Therefore, the unprecedented need of rethinking the resource orchestration process arises in 5G wireless networks towards accommodating and successfully addressing the QoS demands of the significantly increasing number of users. Towards this direction, a new concept of equilibrium is introduced: *Satisfaction Equilibrium*, where the users aim to satisfy their minimum QoS prerequisites instead of targeting at QoS maximization [249, 250]. Perlaza et al. in [85] and [251] have discussed in detail the definition of the Satisfaction Equilibrium (SE) and the general conditions for examining its existence. Also, they have introduced the concept of users' effort to achieve the SE, thus proposing a refinement of the SE, namely the Efficient SE (ESE). At the ESE point, all the users conclude to a resource allocation point, i.e., strategy, which requires the lowest effort to satisfy their minimum QoS prerequisites. In [252] and [253], the concepts of SE and ESE are applied in a simplified uplink power control problem considering interference channels in a single-cell wireless communication environment. Additionally, reinforcement learning algorithms have been introduced in [254] and [255] to determine the SEs and ESEs under different conditions. Nevertheless, the aforementioned approaches are still primitive concerning the resource allocation problem within the uplink transmission power environment. As a result, many interesting properties that emerge when the satisfaction equilibrium framework is applied to this setting have not been revealed yet. Our work aims at

filling this gap, and focuses on the transformation and treatment of the uplink power control problem under the perspective of game theory in satisfaction form.

## **4.2 Contributions**

We study in detail the satisfaction equilibrium points (Chapter 2.1.2) for QoS provisioning in wireless networks, where user actions and decisions are interdependent. This is achieved via examining the uplink power control problem for a general set of users' realistic utility functions, which are increasing with respect to the user's uplink transmission power and decreasing with respect to the intracell interference. A representative example (but not limited to this one) is the Shannon's formula. The novel concept of Minimum Efficient Satisfaction Equilibrium (MESE) is introduced, which is shown to be of special interest among the satisfaction equilibrium points that have already been proposed in the literature, i.e., Satisfaction Equilibrium (SE) and Efficient Satisfaction Equilibrium (ESE).

Assuming that each user is associated with a cost function of arbitrary form with respect to its actions, at the MESE point each user transmits at a power level that satisfies its QoS prerequisites with the lowest cost. It is worthwhile noting that we prove that at the MESE point, not only the system achieves the lowest possible cumulative cost, but also each user is penalized with the minimum cost compared to the corresponding cost at every other ESE point. The existence and uniqueness of the MESE point is thoroughly studied.

A distributed and low complexity Minimum Effort Best Response Dynamics algorithm is proposed, which is based on the best response dynamics behavioral rule and converges to the MESE point that is also the most energy-efficient from all the existing ESEs. A series of experiments are performed to evaluate the performance and attributes of the proposed novel resource management framework which is based on games in satisfaction form. A basic comparative study demonstrates its superi-

ority and benefits in terms of power savings and improved network capacity, against approaches targeting energy-efficiency and/or utility maximization.

### 4.3 Rethinking Uplink Power Control

The introduced definitions and concepts in Chapter 2.1, along with the pressing need for cost-efficient (i.e., energy-efficient) solutions in the era of wireless communications, motivate and support the rethinking and redefinition of the power control problem in wireless networks. Let us consider  $K$  transmitter/receiver pairs denoted by index  $k \in K$ . For all  $k \in K$ , transmitter  $k$  uses power level  $p_k \in A_k$ , with  $A_k$  generally defined as a compact sub-lattice. For each player  $k \in K$ , we denote  $p_k^{min}$  and  $p_k^{max}$  the minimum and maximum power levels in  $A_k$ , respectively. For every pair of devices  $(i, j) \in K^2$ ,  $g_{ij}$  is the channel gain coefficient between transmitter  $i$  and receiver  $j$ .

As it was introduced in Chapter 2.1, a game in satisfaction form is defined as [253]  $\hat{G} = (K, \{A_k\}_{k \in K}, \{f_k\}_{k \in K})$ , where  $K = \{1, \dots, |K|\}$  represents the set of players,  $A_k$  is the strategy set of player  $k \in K$ ,  $u_k(a_k, \mathbf{a}_{-k})$  represents player's  $k$  payoff (i.e., utility function), and  $f_k(\mathbf{a}_{-k}) = \{a_k \in A_k : u_k(a_k, \mathbf{a}_{-k}) \geq u_{thr}\}$  determines the set of actions of player  $k$  that allows its satisfaction, that is its payoff to be above a threshold value  $u_{thr}$ , given the actions  $\mathbf{a}_{-k}$  played by all the other players. A strategy profile is denoted by a vector  $\mathbf{a} = (a_1, \dots, a_{|K|}) \in A$ ,  $A = A_1 \times \dots \times A_k \times \dots \times A_{|K|}$ .

For the rest of this section, we will assume and study uplink power control games in which each player has a utility function that is increasing with respect to its transmission power and decreasing concerning the total summation over the powers of the rest of the players, as the latter quantity acts as interference to the examined player's transmission. One representative example of such utility function that satisfies the aforementioned realistic assumption, is the commonly adopted Shannon capacity which is given by:



$$u_k(p_k, \mathbf{p}_{-k}) = \log_2 \left( 1 + \frac{W}{R_{fixed}} \frac{p_k g_{kk}}{\sigma_k^2 + \sum_{j \neq k} p_j g_{jk}} \right) \left[ \frac{bps}{Hz} \right] \quad (4.1)$$

where  $\sigma_k^2$  is the noise variance at receiver  $k$ ,  $R_{fixed}$  is the fixed requested service data rate and  $W$  the system's bandwidth.

The considered QoS requirement for each user  $k$  is to have a channel capacity  $u_k(p_k, \mathbf{p}_{-k})$  higher than a given threshold  $u_{thr} \left[ \frac{bps}{Hz} \right]$ . The satisfaction correspondence of user  $k$  is subsequently expressed as:

$$\begin{aligned} f_k(\mathbf{p}_{-k}) &= \{p_k \in A_k \mid u_k(p_k, \mathbf{p}_{-k}) \geq u_{thr}\} \\ &= \{p_k \in A_k \mid p_k \geq \frac{R_{fixed}}{W} (2^{u_{thr}} - 1) \frac{\sigma_k^2 + \sum_{j \neq k} p_j g_{jk}}{g_{kk}}\} \end{aligned} \quad (4.2)$$

In the above inequality, note that if a user raises its transmission power then some other users may also have to increase their transmission powers as well to get satisfied. Also, given the strategy profile of the other users  $\mathbf{p}_{-k}$ , the following statement is valid for each user  $k$ :

$$p \in f_k(\mathbf{p}_{-k}) \Rightarrow \forall p^* \in A_k : p^* \geq p, p^* \in f_k(\mathbf{p}_{-k}). \quad (4.3)$$

Thus, given the strategies of the other users, i.e.,  $\mathbf{p}_{-k}$ , there is a transmission power  $p_k^{MSP}$  which on one hand satisfies the QoS prerequisites of the examined user  $k$ , but on the other hand playing with a lower transmission will leave the user unsatisfied. Contrary, if the user transmits with a greater power, then the user will remain satisfied. We will refer to that power  $p_k^{MSP}$  as the *Minimum Satisfying Power* (MSP) of user  $k$  given  $\mathbf{p}_{-k}$ . Note, that under the assumption we made about the monotonicities of the utility functions, the inequality 4.3 will hold true for the rest of our analysis.

As we can observe from the definition of the games in satisfaction form, the existence of an SE depends on the feasibility of the constraints imposed on the player's utility functions. In the following subsection, we examine the existence of an ESE in the uplink power control game  $\hat{G} = (K, \{A_k\}_{k \in K}, \{f_k\}_{k \in K})$  with cost functions  $\{c_k\}_{k \in K}$  and payoff/utility functions  $\{u_k\}_{k \in K}$  (Equation 4.1).

### 4.3.1 Existence of ESE

To prove the existence of at least one ESE point in the uplink power control game  $\hat{G}$  in our setting we first mention the Tarski and Knaster's fixed point theorem [256].

**Theorem 12 (Fixed point theorem)** *Let  $\mathcal{L}$  be a complete lattice and let  $f : \mathcal{L} \rightarrow \mathcal{L}$  be an order-preserving function. Then, the set of fixed points of  $f$  in  $\mathcal{L}$  is also a complete lattice.*

Let  $A$  be the set of the strategy space of the game  $\hat{G}$  as defined above. Let us also define the lattice  $\mathcal{L} = \langle A, \preceq \rangle$ , where  $\preceq$  is the component-wise less or equal. Note that  $\mathcal{L}$  is a complete lattice as all of its subsets have both a supremum and an infimum. The next step is to construct an appropriate function  $g : \mathcal{L} \rightarrow \mathcal{L}$ . For that purpose, we will use the notation  $BR_k(\mathbf{p}_{-k})$  as the best response function of a user  $k$ , while the strategies of the rest of the users are  $\mathbf{p}_{-k}$ . That is, the transmission power  $p_k \in A_k : p_k = \arg \min_{p_k \in f_k(\mathbf{p}_{-k})} c(p_k)$ . Let us define the function  $g : \mathcal{L} \rightarrow \mathcal{L}$  as follows:

$$g(\mathbf{p}) = (BR_1(\mathbf{p}_{-1}), \dots, BR_{|K|}(\mathbf{p}_{-|K|})) \quad \forall \mathbf{p} \in A$$

Note that if  $f_k(\cdot) \neq \emptyset$  for every user  $k$ , then  $BR_k(\mathbf{p}_{-k}) \in A_k, \forall \mathbf{p}_{-k} \in A_{-k}, \forall k \in K$ . Following those definitions we conclude to the following proposition.

**Proposition 4** *If an uplink power control game in satisfaction form  $\hat{G}$  with cost function  $\{c_k\}_{k \in K}$  and utility function  $\{u_k\}_{k \in K}$  (Equation 4), has the  $f_k$  functions for every user  $k$  non empty for every input then it possesses at least one ESE.*

*Proof:* The proof comes from the Theorem 12. As mentioned,  $\mathcal{L}$  is a complete lattice. We can also note that  $\forall \mathbf{p}, \mathbf{p}' \in A : \mathbf{p} \preceq \mathbf{p}'$  it holds that:

$$\begin{aligned} (BR_1(\mathbf{p}_{-1}), \dots, BR_{|K|}(\mathbf{p}_{-|K|})) &\preceq \\ (BR_1(\mathbf{p}'_{-1}), \dots, BR_{|K|}(\mathbf{p}'_{-|K|})) \end{aligned}$$

Or equivalently  $g(\mathbf{p}) \preceq g(\mathbf{p}')$ . That means that for every user  $k$ , when the rest of the users have played  $\mathbf{p}_{-k}$  and then they increase their powers,  $k$ 's best response will be a greater or equal transmission power than it was. The latter holds true because of the monotonicity we assumed on the utility functions, and thus either  $k$ 's best response will still satisfy user's  $k$  QoS prerequisites or user  $k$  should increase its transmission power in order to be satisfied, thus, inevitably playing an action (transmission power) that is related to a greater cost than before. In that fashion, we also proved that  $g$  is an order-preserving function. Following the previous analysis, Tarski-Kraskel's theorem ensures the existence of a fixed point of function  $g$ . That is,  $\exists \mathbf{p} \in A$  :

$$\mathbf{p} = g(\mathbf{p}) \Leftrightarrow (p_1, \dots, p_{|K|}) = (BR_1(\mathbf{p}_{-1}), \dots, BR_{|K|}(\mathbf{p}_{-|K|}))$$

### 4.3.2 Existence of MESE

In the previous subsections, we proved that under some assumptions the set of all the ESEs  $\{E\}$  of the game  $\hat{G}$  is not empty. Let us now define a binary relation  $\preceq_c$  in  $\{E\}$  such that:

$$\mathbf{a} \preceq_c \mathbf{b} \Leftrightarrow \sum_{k=1}^{|K|} c_k(\mathbf{a}_{-k}) \leq \sum_{k=1}^{|K|} c_k(\mathbf{b}_{-k})$$

Note that  $\preceq_c$  is a total ordering for the set  $\{E\}$ . Thus,  $\{E\}$  possesses at least one minimum value. That means that:

$$\exists \mathbf{e}^* \in \{E\} : \forall \mathbf{e} \in \{E\}, \mathbf{e}^* \preceq_c \mathbf{e}$$

Note that  $\mathbf{e}^*$  is an MESE for the game  $\hat{G}$ . So, under the same assumptions as before, we can easily prove the existence of at least one MESE.

### 4.3.3 Uniqueness and Benefits of MESE

In the followings, we provide some propositions that hold in the considered uplink power control game, to show the main benefits of the MESE point and the specific conditions under which it is unique. We will assume that the  $f_k$  functions are

non empty for every input and every player  $k$ , where this assumption ensures the possession of at least one ESE (Proposition 4).

**Proposition 5** *In an uplink power control game  $\hat{G}$  as mentioned above, if an action profile  $\mathbf{p}^+$  is an SE of the game and it holds that  $\forall k \in K, \forall p_k \in A_k : p_k \geq p_k^+, c_k(p_k) \geq c_k(p_k^+)$  there exists one action profile  $\mathbf{p}^*$  that is an ESE in which it holds that  $c_k(p_k^+) \geq c_k(p_k^*), \forall k \in K$ .*

*Proof:* For the proof we exclude the powers  $p_d : p_d > p_k^+, \forall k \in K$ . Thus, the modified strategy space is denoted by  $A'_k$ , and the corresponding game is  $\hat{G}'$ . In the game  $\hat{G}'$ , we know that the strategy  $p_k^+$  will satisfy the user  $k$ ,  $\forall k \in K$ , regardless the strategies of the rest of the users (Equation 4.3). We can now apply proposition 1, which proves the existence of an action profile  $\mathbf{p}^*$  that is an ESE for  $\hat{G}'$ . As it is an ESE in that game we have that:

$$\forall k \in K, \forall p_k \in A'_k : p_k \in f_k(\mathbf{p}_{-k}^*), \quad c_k(p_k) \geq c_k(p_k^*) \quad (4.4)$$

Because by default  $p_k^+$  is the maximum transmission power of the set  $A'_k$  of the  $k^{th}$  user in  $\hat{G}'$ , it means that  $p_k^+ \geq p_k^*$  and consequently  $c_k(p_k^+) \geq c_k(p_k^*)$  based on Equation 4.4. So, the above statement combined with our assumption regarding the monotonicity of the utility function, enables us to conclude to the following statement regarding the initial game  $\hat{G}$ :

$$\forall k \in K, \quad \forall p \in A_k : p \in f_k(\mathbf{p}_{-k}^*), \quad c_k(p) \geq c_k(p_k^*)$$

Due to the above statement and given that  $\mathbf{p}^*$  is certainly an SE in  $\hat{G}$ , we conclude that  $\mathbf{p}^*$  is also an ESE in  $\hat{G}$ . By proving that, we have also proven implicitly that  $\sum_{k \in K} c_k(p_k^+) \geq \sum_{k \in K} c_k(p_k^*)$ .

Note that by assuming increasing utility function with respect to user's  $k$  transmission power and decreasing with respect to the total power of the rest of the users, we gain an ascending monotonicity of the cost functions in the set of ESEs in a way that is described by the following proposition.

**Proposition 6** *In an uplink power control game  $\hat{G}$  let two action profiles  $\mathbf{p}^{*(1)}$ ,  $\mathbf{p}^{*(2)}$  be ESEs. Then for each user  $k$  it holds that  $c_k(p_k^{*(1)}) > c_k(p_k^{*(2)})$  iff  $p_k^{*(1)} > p_k^{*(2)}$ .*

*Proof:* This proposition is proven via the reductio ad absurdum, as follows. If  $c_k(p_k^{*(1)}) > c_k(p_k^{*(2)})$  and it was  $p_k^{*(1)} < p_k^{*(2)}$ , it would mean that in the strategy profile  $\mathbf{p}^{*(1)}$  user  $k$  would remain satisfied if it played  $p_k^{*(2)}$  (Equation 4.3), which reduces its cost. That is a contradiction because  $\mathbf{p}^{*(1)}$  is an ESE. In addition,  $p_k^{*(1)} = p_k^{*(2)}$  can't hold because  $c_k(p_k^{*(1)}) \neq c_k(p_k^{*(2)})$ . Now, assume that  $p_k^{*(1)} > p_k^{*(2)}$ . That would mean that in the strategy profile  $\mathbf{p}^{*(2)}$  user  $k$  would remain satisfied if it played  $p_k^{*(1)}$ . Thus, because those action profiles are ESEs, it should hold that  $c_k(p_k^{*(1)}) > c_k(p_k^{*(2)})$ .

Let us now use those propositions in order to prove the following statement that enables us to study the plurality of the set of all the MESE points.

**Proposition 7** *For any two MESEs  $\mathbf{p}^{\dagger(1)}$ ,  $\mathbf{p}^{\dagger(2)}$  it holds that:*

$$c_k(p_k^{\dagger(1)}) = c_k(p_k^{\dagger(2)}), \quad \forall k \in K$$

*Proof:* Let  $\{E\}$  be the set of action profiles that are ESEs. Let us now denote two MESEs of the game,  $\mathbf{p}^{\dagger(1)}$  and  $\mathbf{p}^{\dagger(2)}$  such that for one user  $k$  it holds that  $c_k(p_k^{\dagger(1)}) \neq c_k(p_k^{\dagger(2)})$ . In order for them to be MESEs the following should hold:

$$\forall \mathbf{p}^* \in E, \sum_{k \in K} c_k(p_k^*) \geq \sum_{k \in K} c_k(p_k^{\dagger(1)}) = \sum_{k \in K} c_k(p_k^{\dagger(2)}) \quad (4.5)$$

As assumed, there is one user  $k$  that  $c_k(p_k^{\dagger(1)}) \neq c_k(p_k^{\dagger(2)})$  and consequently  $p_k^{\dagger(1)} \neq p_k^{\dagger(2)}$ . Without loss of generality, we assume that  $c_k(p_k^{\dagger(1)}) < c_k(p_k^{\dagger(2)})$ . Because of the fact that  $\mathbf{p}^{\dagger(1)}$  and  $\mathbf{p}^{\dagger(2)}$  are ESEs, from Proposition 6 we get  $p_k^{\dagger(1)} < p_k^{\dagger(2)}$ . Thus, the total summation over the costs of all users in  $\mathbf{p}^{\dagger(1)}$  would be lower than the one of  $\mathbf{p}^{\dagger(2)}$  if they do not differentiate in any other strategy. This, denotes that there should be one other user  $j$  ( $j \neq k$ ) that  $c_j(p_j^{\dagger(1)}) > c_j(p_j^{\dagger(2)})$ . With the same argument it holds that  $p_j^{\dagger(1)} > p_j^{\dagger(2)}$ . Let  $\mathbf{p}^+$  be an action profile with  $p_k^+ = p_k^{\dagger(1)}$  and  $p_j^+ = p_j^{\dagger(2)}$ . Note that  $\mathbf{p}^+$  has lower summation over the costs of users  $k, j$  from both

$\mathbf{p}^{\dagger(1)}$  and  $\mathbf{p}^{\dagger(2)}$ . Continuing in that fashion,  $\mathbf{p}^+$  strategy profile picks for every user  $k$  the power that gives  $k$  the lower cost over  $p_k^{\dagger(1)}$  and  $p_k^{\dagger(2)}$ . Because of Proposition 6, the transmission power would always be the lower of the two. If  $c_k(p_k^{\dagger(1)}) = c_k(p_k^{\dagger(2)})$ , let  $p_k^+$  be the lower transmission power of the two. Note that  $\mathbf{p}^+$  is an SE as each user  $k$  was satisfied by playing  $p_k^+$  either at  $\mathbf{p}^{\dagger(1)}$  or at  $\mathbf{p}^{\dagger(2)}$  while all of the other users have played greater or equal transmission powers (Equation 4.3). So, at  $\mathbf{p}^+$  it holds that:

$$\sum_{k \in K} c_k(p_k^+) < \sum_{k \in K} c_k(p_k^{\dagger(1)}) = \sum_{k \in K} c_k(p_k^{\dagger(2)}) \quad (4.6)$$

Note that in order to construct  $\mathbf{p}^+$  we chose strategies between two ESEs. Thus, because of Equation 4.3 and Equation 2.11 we get that  $\forall k \in K, \forall p \in A_k : p \geq p_k^+, c_k(p) \geq c_k(p_k^+)$ . Thus, applying Proposition 5 on  $\mathbf{p}^+$  gives us an ESE  $\mathbf{p}^\dagger$  with

$$\sum_{k \in K} c_k(p_k^+) \geq \sum_{k \in K} c_k(p_k^\dagger) \quad (4.7)$$

Combining inequalities 4.6 and 4.7 we conclude:

$$\sum_{k \in K} c_k(p_k^\dagger) \leq \sum_{k \in K} c_k(p_k^+) < \sum_{k \in K} c_k(p_k^{\dagger(1)}) = \sum_{k \in K} c_k(p_k^{\dagger(2)})$$

which leads to contradiction with Equation 4.5 as  $\mathbf{p}^\dagger$  is an ESE. So,  $c_k(p_k^{\dagger(1)}) = c_k(p_k^{\dagger(2)})$ ,  $\forall k \in K$  which completes the proof.

The above proposition, shows that every MESE point gives the same cost to a given user. Consequently, if  $\forall k \in K, \forall p_1, p_2 \in A_k : (p_1 \neq p_2), c_k(p_1) \neq c_k(p_2)$  (which is a common case in the uplink power control), then the MESE point is unique. The following proposition shows that each user achieves the minimum cost at a MESE point compared to the experienced cost at any ESE point.

**Proposition 8** *In the considered uplink power control game, let  $\mathbf{p}^\dagger$  be a MESE of the game and  $\{E\}$  the set of ESEs, it holds that  $c_k(p_k^\dagger) \leq c_k(p_k^*), \forall k \in K, \forall \mathbf{p}^* \in E$ .*

*Proof:* For the proof let us study the strategy profile  $\mathbf{p}$  that:

$$\forall k \in K, \quad \forall \mathbf{p}^* \in E, \quad p_k = \arg \min_{p_k^*} c_k(p_k^*) \quad (4.8)$$

Thus, the strategy profile  $\mathbf{p}$  picks for each user the power that gives the lowest cost for the user over all its strategies that belong to the set of ESEs. In case of a tie (more than one strategies that minimize the cost), it picks the lower strategy from those. Let us focus on a random user  $k$ . Let  $\mathbf{p}^*$  be one ESE such that  $p_k = p_k^*$ . So, from all the ESEs of the game,  $\mathbf{p}^*$  gives the lowest cost to user  $k$ ,  $c_k(p_k^*)$ . Because of Equation 4.8:

$$\forall i \in K, c_i(p_i) \leq c_i(p_i^*)$$

For the users  $i'$  that holds  $c_{i'}(p_{i'}) < c_{i'}(p_{i'}^*)$ , Proposition 3 gives  $p_{i'} < p_{i'}^*$ . For all users  $i''$  that holds  $c_{i''}(p_{i''}) = c_{i''}(p_{i''}^*)$  the way that we broke the ties gives us  $p_{i''} \leq p_{i''}^*$ . So, combining the above statements we have proven that  $\forall i \in K, p_i \leq p_i^*$ .

Owing to the above, user  $k$  will certainly be satisfied in strategy profile  $\mathbf{p}$  because it was satisfied at the ESE  $\mathbf{p}^*$  in which the other users have played greater or equal transmission powers (Equation 4.3). The above analysis holds for every user  $k$ , thus every user in strategy profile  $\mathbf{p}$  is satisfied, thus  $\mathbf{p}$  is an SE. Note that in order to construct  $\mathbf{p}$ , we chose strategies between strategy profiles that are ESEs. Thus, based on of Equation 4.3 and Equation 2.11 we get that  $\forall k \in K, \forall p \in A_k : (p \geq p_k), c_k(p) \geq c_k(p_k)$ . Now, we can apply Proposition 5 that gives us an ESE  $\mathbf{p}^\dagger$  that:

$$\begin{aligned} \forall k \in K \quad c_k(p_k) &\geq c_k(p_k^\dagger) \\ \sum_{k \in K} c_k(p_k) &\geq \sum_{k \in K} c_k(p_k^\dagger) \end{aligned} \tag{4.9}$$

Taking into consideration Equation 4.8, we can note that only the equality can hold in inequalities 4.9 so

$$\begin{aligned} \forall k \in K \quad c_k(p_k) &= c_k(p_k^\dagger) \\ \sum_{k \in K} c_k(p_k) &= \sum_{k \in K} c_k(p_k^\dagger) \end{aligned}$$

Note that we cannot find an ESE that has lower total cost than  $\mathbf{p}$ . Thus,  $\mathbf{p}^\dagger$  is an MESE. Because of Proposition 4 every MESE assigns the same cost to a given user.

That means that every MESE allocates to each user the minimum cost that it could have in an ESE as exactly  $\mathbf{p}$  does.

One final observation is that if the cost function of every user is increasing concerning its transmission power, MESE would be unique and there would not exist any strategy profile that satisfies all of the users and simultaneously allocates on any user lower cost value than the MESE does. That can be easily concluded if one applies Proposition 5 on the unique MESE.

#### 4.3.4 Algorithm & Convergence

We present a distributed algorithm that converges at a Minimum Efficient Satisfaction Equilibrium (MESE) of the game  $\hat{G} = (K, \{A_k\}_{k \in K}, \{f_k\}_{k \in K})$ . For this purpose we first introduce the *Best Response Dynamics* (BRD) in the context of a game in satisfaction form.

##### Best Response Dynamics

Best Response Dynamics is defined as the behavioral rule in which each user always chooses its uplink transmission power to be its best response (BR) as defined earlier, depending on the uplink transmission power of the rest of the users. In the context of this thesis, the dynamics should not be sequential but asynchronous. This means, that each user can play a strategy whenever it has a response from the rest of the users, however the users should not choose their strategies simultaneously. As shown in [253], when the BRD start from an SE as an initial strategy profile, they converge monotonically to an ESE. The algorithm that is studied in the following section is the BRD starting with the action profile associated with the lowest effort for each user. We will show that the considered monotonicity of the utility function reduces the complexity for each user to determine its best response. Thus, an efficient implementation of BRD is achieved in our setting.



### Minimum Effort Best Response Dynamics

Initially, each user  $k$  pre-processes its data with the *Preparation Phase* of the Minimum Effort Best Response Dynamics (MEBRD) algorithm. Note that after executing the preparation phase of the MEBRD algorithm, each user  $k$  has computed the vector  $\mathcal{S}_k[\cdot]$ . The element  $\mathcal{S}_k[j]$  denotes the uplink transmission power that provides the minimum cost over the powers  $P_k[j], \dots, P_k[|A_k| - 1]$ . Therefore, if the Minimum Satisfying Power (MSP) is a power  $p_k^{MSP} = P_k[i] \in A_k, \forall k \in K$  given  $\mathbf{p}_{-k}$  then  $BR_k(\mathbf{p}_{-k}) = \mathcal{S}_k[i]$ .

---

#### Algorithm 4 Preparation Phase

---

```

1: Sort in ascending order all powers  $p_k$  in a vector  $P_k[\cdot]$ ;
2:  $power \leftarrow P_k[|A_k| - 1]$ ;
3:  $min \leftarrow c_k(power)$ ;
4:  $\mathcal{S}_k[|A_k| - 1] \leftarrow power$ ;
5: for  $p \leftarrow |A_k| - 2, 0$  do
6:   if  $c_k(P_k[p]) \leq min$  then
7:      $\mathcal{S}_k[p] \leftarrow P_k[p]$ ;
8:      $power \leftarrow P_k[p]$ ;
9:      $min \leftarrow c_k(P_k[p])$ ;
10:     $indexOfMin \leftarrow p$ ;
11:  else
12:     $\mathcal{S}_k[p] \leftarrow power$ ;
13:  end if
14: end for
15:  $Msp \leftarrow P_k[0]$ ;
16:  $IndexOfMsp \leftarrow 0$ 
17: play  $\mathcal{S}_k(0)$ ;
```

---

After the preparation phase, each user chooses the power that minimizes its

cost function. Therefore, the starting strategy profile of the dynamics will be:  $\mathbf{p}_{start} = (\mathcal{S}_1[0], \dots, \mathcal{S}_{|K|}[0])$ . Then, each user who is in turn to play executes the *Turn Phase* of the MEBRD algorithm. The auxiliary vectors  $\mathcal{S}_k[]$  will help each user to calculate its MSP in every turn with a binary search. Due to the monotonicity of the utility function and the fact that each user either does not change or increases its transmission power at each turn (as we will prove in the convergence section), user  $k$  should only do binary search from the MSP of its previous turn to its current  $p_k^{max}$ . With the binary search each user  $k$  is searching for the smallest value in  $P_k[]$  that satisfies  $u_k(p, \mathbf{p}_{-k}) \geq u_{thr}$ , which is the MSP of player  $k$  given  $\mathbf{p}_{-k}$ . The algorithm stops when no one user has a new best response strategy to play.

---

**Algorithm 5** Turn Phase

---

- 1: **if** still satisfied **then**
  - 2:   do not change transmission power;
  - 3: **else**
  - 4:    $[Msp, IndexOfMsp] \leftarrow BinarySearch(P_k[], Msp, |A_k|, u_k(), \mathbf{p}_{-k}); \{\text{Finds new lower limit (as the vector } \mathbf{p}_{-k} \text{ has changed) using binary search in } P_k[] \text{ from previous Msp to } P_{max} \text{ using the utility function of the player}\}$
  - 5:   **play**  $\mathcal{S}_k(IndexOfMsp); \{\text{When all of } P_k[indexOfMsp] \text{ to } P_k[|A_k| - 1] (= P_{max}) \text{ strategies satisfies you play the power that gives the lowest cost}\}$
  - 6: **end if**
- 

### Convergence

In this section we prove that the above algorithm converges, under finite number of steps, to an MESE of the considered uplink power control game. To conclude this, we prove a set of propositions as follows:

**Proposition 9** *The MEBRD algorithm monotonically converges to a strategy profile  $\mathbf{p}^\dagger \in A$ .*

*Proof:* In the first turn of each user  $k$ , it examines whether it is satisfied or should increase its power to get satisfied. Because of the dynamics, it will not choose a lower transmission power as everyone started with the one that is associated with the lowest cost,  $\mathcal{S}_k(0)$ . This fact shows that in the first round, each user  $k$  is either playing again  $\mathcal{S}_k(0)$  (if it is still satisfied) or increases its power.

Since the utility functions  $u_k$  are decreasing with respect to the total summation over the powers of the users, if a set of users increases its power levels and no user does the opposite, an individual that kept its power unchanged, is now satisfied by greater power levels than before (or with the same ones). Taking into account the above fact and that in the first round all users either raised their powers or held the same values, we can conclude to the following statement. In every turn all users will either keep the power levels of the previous turn (if they are still satisfied) or increase them (to get satisfied). It is noted that because of the assumption that  $f_k()$  will be not empty for every  $\mathbf{p}_{-k}$ , user  $k$  will always have a best response. Therefore, for each user  $k$  its sequence of played strategies  $\{p_k\}$  is increasing through a finite set, so, its strategy eventually converges (monotonically) to a strategy  $p_k^\dagger$ .

**Proposition 10**  $p^\dagger$  is an ESE.

*Proof:* In this algorithm, when the turn phase of a user  $k$  is running, it chooses the transmission power that satisfies it but also has the minimum cost. When the previous power that the user selected, is not still the one mentioned above it should change strategy when its turn comes. Due to proposition 6, all users will eventually converge to a transmission power that has those two properties. Thus, because of the dynamics and its eventual convergence,  $p^\dagger$  is an ESE.

**Proposition 11**  $p^\dagger$  is a MESE.

*Proof:* Let  $\mathbf{p}^* = (p_1^*, \dots, p_{|K|}^*)$  be an ESE of the game and  $\mathbf{p}^\dagger = (p_1^\dagger, \dots, p_{|K|}^\dagger)$  be the strategy profile that our algorithm converges. Before we continue note that based on

Equation 2b, 6 it holds that:

$$\forall k \in K, \quad \mathcal{S}_k(0) \leq p_k^* \quad (4.10)$$

*Is there a possibility for one player  $i$ , to hold  $p_i^\dagger > p_i^*$ ?*

Following Equation 4.10 and due to the fact that in every turn a user cannot decrease its power, in order for this to happen, a user  $j$  from all the users, has played a power that exceeds its  $p_j^*$  during one of its turns. Let  $\mathbf{p}$  be the strategy profile of the game right before this turn. Because of the above we have:

$$\forall k \in K, \quad p_k \leq p_k^* \quad (4.11)$$

Thus player  $j$  played a transmission power  $p_j^{exc} \in A_j$  that

$$p_j \leq p_j^* < p_j^{exc} \quad (4.12)$$

Since the strategy profile  $\mathbf{p}^* = (p_1^*, \dots, p_{|K|}^*)$  is an ESE, from Equation 2.11, Equation 4.3, we also get that:

$$\forall p \in A_j : (p \geq p_j^*), \quad c_j(p) \geq c_j(p_j^*) \quad (4.13)$$

Because of the fact that  $\mathbf{p}^*$  is an SE, we get that  $p_j^* \in f_j(\mathbf{p}_{-j}^*)$ . Because of Equation 4.11, and the fact that the utility functions are decreasing with respect to the total summation of the powers of the other users we obtain that:

$$p_j^* \in f_j(\mathbf{p}_{-j}). \quad (4.14)$$

Combining Equation 4.12, Equation 4.13, and Equation 4.14 user's  $j$  best response can be neither  $p_j^{exc}$ , nor any power that is strictly greater than  $p_j^*$ . Thus, user  $j$  will not play a power that is greater than  $p_j^*$ . So the answer to the previously stated question is negative.

Considering the above argument, it holds that,  $\forall k \in K, p_k^\dagger \leq p_k^*$ . Furthermore, since the action profiles  $\mathbf{p}^*$  and  $\mathbf{p}^\dagger$  are ESEs, it holds that for every user  $k$ :

$$\forall p \in A_k : p \geq p_k^\dagger, \quad c_k(p) \geq c_k(p_k^\dagger) \quad (4.15)$$

$$\forall p \in A_k : p \geq p_k^*, \quad c_k(p) \geq c_k(p_k^*) \quad (4.16)$$

But as we have proven for each user  $k$  it holds that  $p_k^* \geq p_k^\dagger$ , therefore based on Equation 18a we conclude:

$$\forall k \in K, \quad c_k(p_k^\dagger) \leq c_k(p_k^*)$$

Consequently,

$$\sum_{k \in K} c_k(p_k^\dagger) \leq \sum_{k \in K} c_k(p_k^*)$$

Given that the above analysis is valid for every ESE  $\mathbf{p}^*$ , it will also hold true for one MESE. So  $\mathbf{p}^\dagger$  is a MESE.

One corollary of the above is that since it holds that:

$$\forall \mathbf{p}^* \in E, \forall k \in K, \quad p_k^\dagger \leq p_k^* \quad (4.17)$$

we also proved that from all the ESEs,  $\mathbf{p}^\dagger$  is the most efficient one in terms of power.

## Complexity

In this subsection, the complexity of the algorithm is studied in the case of the users are playing sequentially in a given order. Let us concentrate on one user  $k$  to specify its CPU time complexity excluding the time that other users take to make their decisions. At first, user  $k$  should sort the array  $P_k[]$  so that would be a  $\mathcal{O}(|A_k| \cdot \log_2(|A_k|))$  time complexity. In every cycle of turns, someone should always increase its power, or else the algorithm converged to  $\mathbf{p}^\dagger$ . The worst case is bound by the case where the game would have  $\mathcal{C} = |A_1| + \dots + |A_{|K|}|$  cycles of turns. So, in  $\mathcal{C} - |A_k|$  cycles, user  $k$  will find out, in constant time, that it is satisfied. On the other hand, in  $|A_k|$  cycles of turns the user should do one binary search in  $P_k[]$  to find out its next move. Therefore, for all of the cycles it will totally spend  $\mathcal{O}((\mathcal{C} - |A_k|) + |A_k| \cdot \log_2(|A_k|))$ . Thus, the total time complexity is  $\mathcal{O}((\mathcal{C} - |A_k|) + |A_k| \cdot \log_2(|A_k|))$ . Note, that if each user has the same cardinality in its strategy space,  $N$ , the total complexity will be  $\mathcal{O}(|K| \cdot N + N \cdot \log_2(N))$ .

### 4.3.5 Empirical Evaluation

In this section, we provide some indicative numerical results to evaluate the performance of the MEBRD algorithm, while at the same time demonstrating the key benefits of the MESE point. For demonstration purposes, the utility function of each user is assumed to follow Equation 4. The distance  $d_k, \forall k \in K$  from the base station is randomly and uniformly distributed within the range of 1 to 50 *m*. The gain  $g_k$  of each user  $k$  is inversely proportional to the square of its distance  $d_k$ , i.e.,  $g_k = \frac{1}{d^2}$ . Each user is assumed to have a finite number of power levels, while its maximum transmission power is 1 *Watt*.

#### Pure Operation

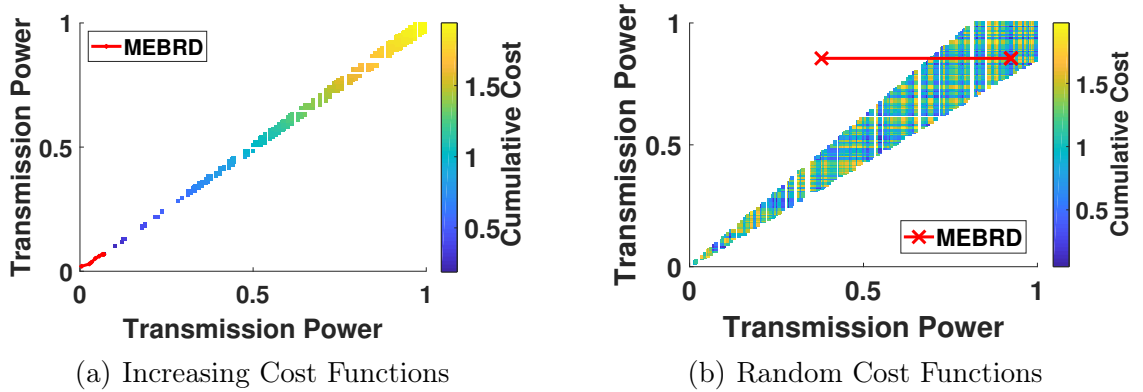


Figure 4.1: Convergence of the MEBRD algorithm in a 2-user uplink power control game

Figure 4.1 presents, for a two-user uplink power control game, the steps of the MEBRD to converge to an MESE point with respect to each user's transmission power, assuming either increasing cost function (Figure 4.1a) or arbitrarily randomly chosen cost function (Figure 4.1b). The colored region represents all the strategy profiles that are SEs and each point's color depends on the cumulative cost of the two users, where the light and dark color represents a high and low cost, respectively.

It is noted that in Figure 1a each user starts with its lowest transmission power and monotonically converges to the unique MESE which is also the SE that charges each user with the lowest possible cost and power. Please also note that in Figure 4.1a and Figure 4.1b, the linear trend of the satisfaction region stems from the selection of the  $u_{thr}$  value, which in our case represents the utility that each user would gain if it transmitted with an intermediate power, i.e.,  $0.5W$ , in its feasible power range. In the rest of this section, we consider increasing cost function with respect to user's transmission power.

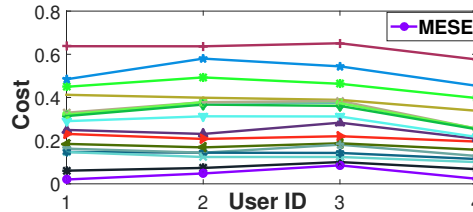


Figure 4.2: Cost of various ESE points vs user ID

Assuming that each user is capable of achieving its QoS prerequisites, it is highly possible for the game to possess multiple ESEs. In Figure 4.2, we compare the cost allocation of different ESEs (multiple curves) of an uplink power control game with four users. We confirm that the MESE achieves the lowest cumulative cost by assigning to each user the transmission power associated with the lowest effort compared to every ESE, as claimed in Proposition 8

### Complexity Evaluation

The time complexity of the MEBRD algorithm mainly depends on the number of users in the system and the number of the discrete transmission power levels that each user possesses. Consequently, Figure 4.3 presents the behavior of the execution time of the MEBRD algorithm with respect to the number of each user's discrete transmission power levels. Considering a system with 35 users we observe that the algorithm can handle efficiently (i.e., less than 10 sec runtime) a large number of

power levels (i.e., 65536), and thus it can be also used as an approximation for a continuous set of power strategies.

The time complexity of the MEBRD algorithm depends also in every parameter of the game, such as for instance each user's threshold  $u_{thr}$ , above which the user is satisfied. For example, if each user is satisfied by gaining a very low bit rate, most of the transmission powers will satisfy the user independently of the power of the others, hence the MESE that the MEBRD algorithm converges will consist of low transmission powers and it will converge fast. Specifically, Figure 4 presents the time needed for the MEBRD to converge to its MESE as a function of the number of users in the system, assuming 150 different transmission power levels available to each user. In particular two cases are studied with respect to each user's threshold parameter  $u_{thr}$ : (a) high threshold (Figure 4.4a) and (b) low threshold (Figure 4.4b). In the latter case, it is easier to satisfy the requirements and more SEs are expected to exist. Please also note that the low thresholds are selected such that the game even with 400 users in the system will possess at least one equilibrium. The top curve in each subgraph (referred to as Static) represents the time needed for the MEBRD algorithm to converge. For example, when the game consists of a lower number of users (e.g., 100) each user is satisfied easily, thus experiencing a fast convergence. On the other hand, when the number of users is approaching 400, the algorithm needs more iterations to converge. After that number, the system does not possess any equilibrium points. In this case, the number of required iterations decreases since each user makes greater steps increasing its transmission power in order to meet its QoS prerequisites and eventually fail.

The second curve in the two subgraphs of Figure 4.4 (referred to as Dynamic Entrance) denotes the time needed in order the MEBRD algorithm to converge, when each run of the MEBRD is not independent as before. In contrary, *five* users every time enter the game, while the previous number of users had converged to the MESE and use this point in order to initialize their strategies in the new run. As the



algorithm does not need to be in sequential turns, it will still converge to the MESE for the new number of users, while lower convergence time than before (top curve) is observed. Last it is observed that in the latter case the time needed for convergence is not necessarily strictly increasing with respect to the number of users.

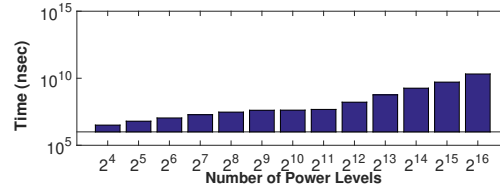


Figure 4.3: Execution time of the MEBRD algorithm as a function of the number of each user's transmission power levels

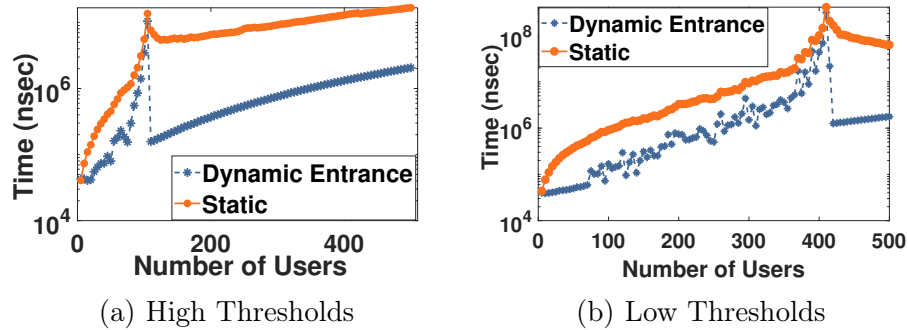


Figure 4.4: Execution time of the MEBRD algorithm as a function of the number of users in the system

## Comparative Evaluation

In this subsection, we compare the MESE as a strategy profile, with the corresponding ones that an Energy Efficiency Maximization or Shannon Maximization algorithm would obtain, with respect to different performance metrics. Specifically, in this scenario, *six* users are considered in the system that is located at decreasing distances from the base station. Therefore, users with lower ID have the highest distances from the base station, thus worse channel conditions. With reference to the energy

efficiency maximization case, a utility function that represents the achievable data rate over corresponding consumed power (expressed in bits/joule) was adopted, as typically defined and used in corresponding literature [241]. In both cases of Shannon and energy efficiency utilities we set  $R_{fixed} = 64Kbps$  and  $W = 10^6Hz$ . In particular, Figure 4.5a suggests that for the first 3 users (the 3 users that are the farthest from the base station) the energy efficiency maximization algorithm, as expected, scores higher in the energy efficiency metric. However, as shown in Figure 4.5b this happens at the cost that each of the three users transmits with a very high transmission power compared to the MESE, hence gaining a higher bit rate than their QoS prerequisites (Figure 4.5c). Thus, the MESE strategy profile converges to quite low transmission powers, while assigning to each user bit rate that is close to its threshold (green line in Figure 4.5c) and therefore satisfying each user's requirement. The other two strategy profiles do not properly adapt to the users' needs, since they either exceed the threshold forcing the users to transmit with high power, or leave the users unsatisfied.

### 4.3.6 Summary

We adopted the game theory in satisfaction form, towards properly re-defining and treating the uplink power control problem in wireless networks, for a general set of users' realistic utility functions, which are assumed to be increasing with respect to the user's uplink transmission power and decreasing with respect to the intra-cell interference. Given this setting, we initially defined and discussed the different equilibrium points, i.e., Satisfaction Equilibrium, Efficient Satisfaction Equilibrium, and Minimum Efficient Satisfaction Equilibrium. Among those, the MESE appeared to be of high theoretical and practical interest, and therefore its properties were thoroughly examined.

In particular, it is shown that at the MESE point, for any arbitrary cost function, each user transmits with a power level that satisfies its QoS prerequisites with the

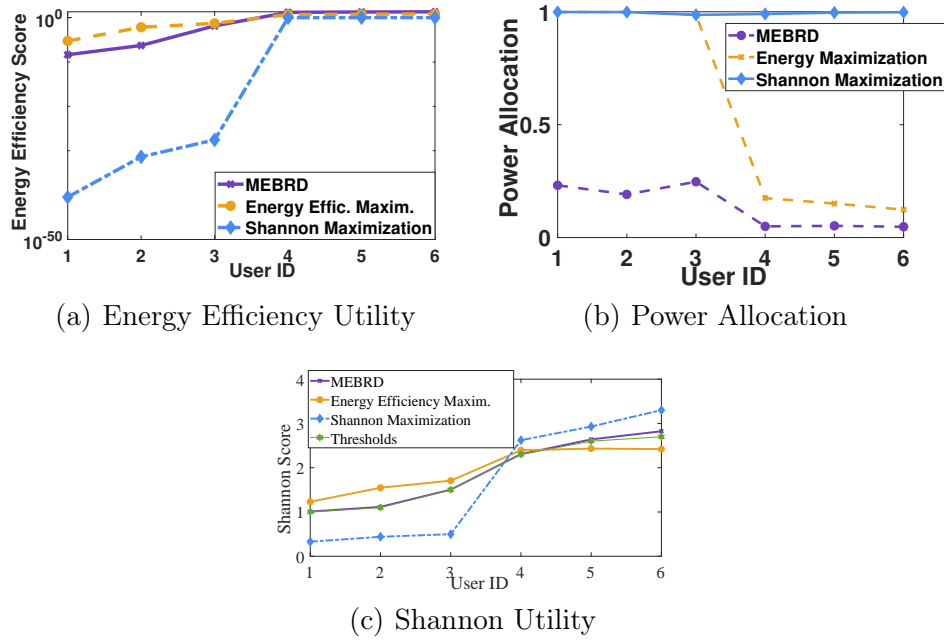


Figure 4.5: Comparison of strategy profiles for MEBRD, Energy-Efficiency Maximization and Shannon Maximization

lowest cost not only from its own perspective but from the whole system's perspective as well. The existence and uniqueness of the MESE point were shown and a distributed Minimum Effort Best Response Dynamics algorithm was introduced to determine the MESE point. Detailed numerical and comparative results were presented to highlight and reveal the benefits of the MESE point compared to other types of equilibrium (e.g., ESE point) and the superiority of the proposed novel resource management framework in terms of power savings and improved network capacity.

# Chapter 5

## Smart Technologies

### 5.1 Learning and Game theoretic Demand Response Management

The combined problem of power company selection and Demand Response Management in a Smart Grid Network consisting of multiple power companies and multiple customers is studied via adopting a distributed learning and game-theoretic technique. Each power company is characterized by its reputation and competitiveness. The customers who act as learning automata select the most appropriate power company to be served, in terms of price and electricity needs' fulfillment, via a distributed learning-based mechanism. Given customers' power company selection, the Demand Response Management problem is formulated as a two-stage game theoretic optimization framework, where at the first stage the optimal customers' electricity consumption is determined and at the second stage the optimal power companies' pricing is calculated. The output of the Demand Response Management problem feeds the learning system in order to build knowledge and conclude to the optimal power company selection. A two-stage Power Company learning selection and Demand Response Management (PC-DRM) iterative algorithm is proposed in or-

der to realize the distributed learning power company selection and the two-stage distributed Demand Response Management framework. The performance of the proposed approach is evaluated via modeling and simulation and its superiority against other state-of-the-art approaches is illustrated.

### **5.1.1 Related Work**

#### **Smart Grid Network**

The growing number of users and their demands, as well as the ever-increasing competitive environment in which electricity providers are called upon to coexist, testify that new smart distribution networks need to be studied and developed. The existing network is therefore under great pressure from the various challenges and needs arising from the environment, consumers, market, and infrastructure issues. These challenges and needs are more important and urgent than ever and have led the network to expand and enhance its functions to smarter features with the help of fast-growing technologies. The shift in the development of transmission networks to be smarter has been briefly defined as "Smart Network". Some of the key goals of these new smart energy distribution networks are to optimally serve the needs of consumers as well as the healthy profitability of electricity companies [257].

The term "Smart Network" has been in use since the end of 2003 and the first appearance of the term dates back much earlier. There are several definitions of the "Smart Network" that focus on either its operation or its technology. The common point of all, is the application of digital processing and communications to the electricity grid, with data flow and management being done by a centralized system called "Smart Grid" [258].

The idea of a smart grid is an electricity grid that can intelligently integrate the actions of all users connected to it, generators or consumers, in order to provide an efficient, economical and secure electricity supply. A smart network uses innovative

products and services, combined with intelligent monitoring of the network status [259].

The Smart Network connects supply and demand by enabling both producers and consumers to set their operating needs more flexible and sophisticated. For example, consumers are only able to consume at high prices for extremely important reasons and to shape their consumption according to the information they have about the present consumption price. On the other hand, producers with high flexibility can adjust their sales price to maximize their profits, while at the same time depending on their electricity generation costs, they can offer consumers discount periods, thereby expanding their advertising influence and gaining more users.

Coupled with the smart grid features offered, the liberalization of the electricity market that began last decade, or even earlier, especially in the United States of America, has led to the increasing establishment of smart grids. Consumers now have the option of choosing the company from which they purchase electricity. In Massachusetts, the electricity market was liberalized in 1997 [260], in Maryland in 1999 and in Texas in 2002 [261]. The liberalization of the electricity market forces to creation of a more effective, flexible, and reliable electricity system.

### **Demand Response Management**

With the increasingly demanding challenges of the growing electricity needs, aging infrastructure, and the integration of renewable green energy resources, a new way of addressing these demands will need to be developed by electricity distribution networks. As we have already mentioned, new smart electricity distribution networks face these challenges by managing the concept of demand response. Essentially, demand response management refers to the implementation of techniques to control energy consumption by consumers, improve energy efficiency and reduce the cost of electricity generation from electricity companies [262, 263, 264, 265]. One of the key objectives of demand response management is to reduce the differences between

electricity consumption and average consumption in the network so that there is a balance between demand and supply [266]. Modeling the problem of managing the demand response is very important for achieving the goals of the Smart Grid Network. Specifically, there are several different models of this problem, but the common point is the aim of balancing consumers' demand for electricity and determining the best plan for electricity supply and pricing from companies' side, in order to increase and reduce companies' profit and generation cost, respectively.

In [267] the authors study the demand response management problem in a centralized manner, by using a finite-horizon Markov decision process (MDP) and a linear programming technique, in order to maximize companies' profit and determine the energy load in a real-time electricity market. On the other hand, a decentralized approach of the demand response management problem is studied in [268], where the authors formulate the problem of managing the demand response as a non-convex optimization problem, where convex relaxation techniques are applied, and the companies' optimal pricing is determined. A different formulation of the demand response management problem is followed in [269], where the notion of micro-grids is developed in the electricity market in order to fulfill power demand in specific regions. The authors address the problem of demand response management by constructing a Stackelberg game with a unique equilibrium solution. The notion of micro-grids is also studied in [270], where the authors examine the demand response management problem for multiple energy resources (i.e., Fuel cells, PhotoVoltaic modules), and they propose a two-stage stochastic programming approach to minimize the operational cost in energy management.

In [271], a price prediction model with the use of an Artificial Neural Network is introduced by the authors, while the costumers adopt a Reinforcement Learning mechanism in order to deal with the uncertainty in the feature prices and make optimal decisions regarding their home appliances. A quite similar method, in terms of the construction of a predictive model, is followed in [272], where the customers use

the prediction control in order to manage in an autonomous manner their ON/OFF periods and determining their optimal decisions for the demand response management problem. A neural network is also used in [273], where the authors introduce a smart grid model that considers the power consumption and the customers' satisfaction, while a projection neural network is used for minimizing the electricity cost for all the users. Furthermore, the demand response management is studied also in [274], as the costumers utilize renewable energy resources, which are controlled by cloud servers, and the use of current security mechanisms (i.e., RSA, AES, ECC) is studied for security purposes.

An incentive-based demand response management optimization framework is introduced in [275], where the customers efficiently determine their optimal households' energy consumption during peak hours, while in [276] the authors address the peak loads in an electricity market by introducing quality of service metrics for the customers, and a data analytical management scheme. The proposed scheme is based on the analysis of consumers' consumption data gathered from smart homes. On the other hand, in [277] the authors implement a heuristic demand response technique for consumption scheduling of appliances, in order to decrease the peak to average ratio of power demand. The authors use stochastic programming, and communication requirements, in order to schedule customers' consumption in real-time. The authors in [278] highlight the importance of the use of auto-configured devices, and based on that they design an adaptable energy management system, in order to determine the customers' demand response. The Pareto optimal demand response management based on energy costs and load factor is studied in [279], where the authors introduce a multi-objective optimization problem and its Pareto optimality is determined. The demand response management problem has been studied also in the era of multiple data centers, wherein [280] the authors introduce an approach to dynamically adjust the datacenters' load to balance the unstable solar input into the energy grid.

Moreover, in [281] the authors implement a large-scale optimization approach



in a distributed manner, in order to control and support the demand response of residential appliances. This scheme is based on a hierarchical control and a coordination system, that enables the exchange of information between the utility and the management system. A hierarchical-based system is also used in [282], where the authors introduce a dynamic pricing response algorithm, that considers both the service providers' profit and customers' costs. The hierarchical decision-making is made based on a Reinforcement Learning mechanism, where the Q-learning algorithm is adopted to solve the decision-making problem. In [283] the authors examine and consider models from the marketplace in order to design demand response management to match power supply and meet customers' demands. The authors in [284] propose distributed algorithms for electricity companies and consumers, in order to maximize social welfare. [285] presents a new algorithm for finding the optimal time of use of electricity.

In addition, it is equally important to apply game theory for modeling the demand response management problem, as game theory is proved to be quite effective in dealing with complex interactions. The authors in [286] formulated the problem as a non-cooperative N-person game, and a distributed demand response management strategy is proposed in order to achieve the minimum energy cost. Network congestion is also studied in [287] and a load management strategy modeled as a "Smart Network" game is proposed. The authors in [288] studied the planning of home energy consumption through a Stackelberg game, in which the electricity companies are the leaders of the network and the consumers adjust their demands.

The key point of all the above research is that in smart electricity distribution networks, there is only one company that supplies electricity to consumers. However, as we have already pointed out, the liberalization of the electricity market now gives consumers the option to choose between many energy providers [289], [290], [291], which brings new challenges to the interaction between companies and consumers. It is therefore imperative to study the problem of managing the demand response

in an environment where many electricity companies coexist. The first survey in this multi-company and multi-consumer environment is presented in [285], but the authors do not take into account the power functions of the electricity companies.

To this end, the need to develop new smart energy distribution networks to meet all the growing demands has become an urgent need in modern society. The customers' demands are now directly linked to smart electricity distribution networks, where the use of utility functions and the demand response management with new theoretical models, demonstrate a vital role. In such smart grids, the characteristics of the Demand Response Management (DRM), Network Economics (NE), and electricity company choice shape the market [286]. The theory of Network Economics aims to determine the price of electricity, in order for a successful penetration on the electricity market to be achieved [292]. The process of selecting electricity companies aims to bridge the gap between the electricity companies and the customers, while at the same time enables consumers to make the best choice in terms of saving money, and the companies to meet the electricity demands of the network [293].

In [287], the problem of managing the demand response is dealt with only the customers' point of view, as the authors study the problem of the load control by applying a distributed energy consumption planning to customers and a dynamic pricing strategy to companies. Real-time power planning is calculated by adopting a Stackelberg game model, where the power company is the leader, setting real-time price and customers planning their devices' electricity consumption. A similar approach is discussed in [294]. The problem of load balancing and peaks avoidance is studied in [295], where an incentive-based algorithm for home load management is proposed, reducing overall energy costs and taking into account the satisfaction of the users. Also, aiming at load balancing, the authors in [296] propose an optimal game pricing strategy for smart grid networks, by optimizing the value per day time period, so that the electricity load of the network remains in an equilibrium state rather than in peak values.

The home demand response management problem is studied in [297], taking into account the underlying power distribution network and the associated constraints. The Demand Response Management problem is formulated as a flow power problem, and a distributed algorithm is proposed to determine the optimal demand planning, while allowing communications between the electricity supplier and the households. The direct interaction between the electricity company and the customer is studied in [298], where the problem of allocating a certain amount of load adjustment by the electricity company to the customers is examined, with the aim of minimizing the total loss of the consumer.

In [299], the authors study the interaction between an energy provider and multiple customers through a Stackelberg game approach, and propose an algorithm that aims to control the loads of the users' devices. A similar approach is being studied in [289] and [300], involving multiple electricity companies and multiple customers, where the aim of the Stackelberg game is to maximize the revenue of each electricity company and minimize the amount of payment that each customer makes.

### 5.1.2 Contributions

We jointly study the combined problem of optimal power company selection by the customers based on a reputation and competitiveness distributed learning framework, and the problem of demand response management based on a game theoretic approach. We assume the existence of an open electricity market, and we formulate it as a Smart Grid Network, which consists of multiple power companies and customers. Each power company is associated with a reputation and competitiveness factor per timeslot, while the customers acting as RL-agents adopt the Stochastic Learning Automata methodology (Section 2.2.3) in order to select the power company that they will be served from. The learning power company selection algorithm runs once at the beginning of each timeslot. To fully capture the interaction between the power companies and the customers in the Smart Grid Network, the demand

response management problem is modeled as a two-stage non-cooperative game.

At the first stage, the customers by considering the companies' pricing policies, determine their optimal electricity consumption that maximizes their utility, while at the second stage, given the optimal customers' consumption, the power companies evaluate their optimal pricing policies that maximize their profit. Moreover, in our work, the non-shiftable and shiftable customers' demands are treated with different priority. Following the proposed two-stage non-cooperative game-theoretic approach, the customers and the power companies can interact and finally reach the Nash Equilibrium point, if proper strategies are selected on both sides. It is noted also, that the demand response management optimization problem consists of multiple iterations at the beginning of each timeslot, thus it is of a different time scale compared to the distributed learning power company selection algorithm.

The following specific contributions and innovations of this work are described in detail, in order to achieve the aforementioned key objective.

1. A distributed learning framework is proposed towards implementing the customers' power company selection at the beginning of each timeslot. The selection probabilities of each customer are updated by considering power companies' reputation and competitiveness factors. The reputation and competitiveness factor of each power company reflects the provided discount, its achievable peak-to-average ratio, and its penetration to the electricity market.
2. Representative power companies' and customers' profit and utility functions, respectively, are introduced to capture their behavior within the electricity market. Specifically, power companies' profit function reflects the tradeoff between the company's revenue and its corresponding electricity generation cost. On the other hand, each customer's utility function reflects the tradeoff between the satisfaction of its electricity demands and its corresponding total cost based on a fair pricing policy by considering the electricity consumption of the rest

of the customers in the Smart Grid Network.

3. Following the distributed learning-based power company selection process by the customers, the optimization problem of maximizing customers' utility function and power companies' profit function, is formulated as a two-stage game. The Nash Equilibrium point of the two-stage game is achieved based on the selection of appropriate strategies from the customers and power companies, while a distributed algorithm that obtains the aforementioned equilibrium point, is proposed.

### 5.1.3 System Model

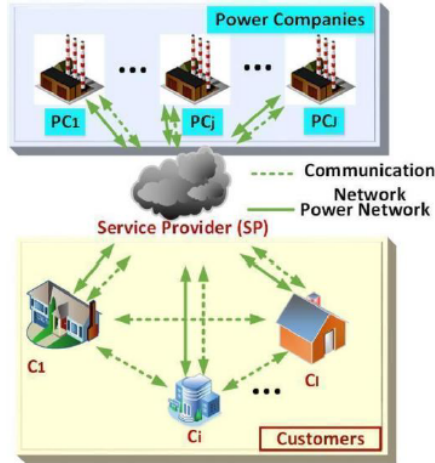


Figure 5.1: Smart Grid Network

Figure 5.1 shows a graphical representation of the considered Smart Grid Network, consisting of multiple users and multiple power companies. There is two-way communication between companies and consumers that is achieved through a centralized Service Provider (SP) management system. In essence, this centralized management system acts as an intermediary connection between power companies and customers, with which customers and companies are connected through power

connection (solid lines), while two-way communication connections (dotted lines) enable the connectivity between companies and customers. The centralized management system allows for the exchange of information, including the power companies' prices and customers' load demand. Each customer is equipped with an Energy Management Controller (EMC), which coordinates the power consumption among customer's smart appliances and is aware of appliances' shiftable and non-shiftable electricity demand and consumption.

A fundamental novelty that differentiates this work from the recent relevant literature, is that each consumer is informed through the centralized management system about the network's total energy consumption, and as a result, each customer's privacy is maintained. The information of the energy consumption, allows us to apply price fairness criteria regarding the consumption price of each customer and so in the Smart Grid Network can coexist harmonious consumers of different economic levels.

We define as  $\mathbb{J} = \{1, \dots, j, \dots, J\}$  the set of electricity power companies, and with  $\mathbb{I} = \{1, \dots, i, \dots, I\}$  the set of customers that exist in the Smart Grid Network. The whole operation time is divided in  $T$  timeslots, where  $\mathbb{T} = \{1, \dots, t, \dots, T\}$  denotes the corresponding set. Moreover,  $A_{s,i}$ ,  $A_{ns,i}$  denote the set of appliances characterized by shiftable and non-shiftable electricity consumption of customer  $i$ ,  $i \in \mathbb{I}$ , respectively, while customer's  $i$  overall set of appliances is denoted as  $A_i = A_{s,i} \cup A_{ns,i}$ .

### Utility and Customers' Characteristics

The considered Smart Grid Network consists of multiple customers and power companies. Each customer  $i$ ,  $i \in \mathbb{I}$  is characterized by its demand  $d_i^{(t)} [KWh]$  of electricity units per operation timeslot  $t$  towards meeting the needs of its appliances  $a$ ,  $a \in A_i$ . Based on the availability of the generated electricity by the power companies and its corresponding price, customer  $i$  consumes  $e_{i,j}^{(t)} [KWh]$  amount of electricity via selecting the power company  $j$ ,  $j \in \mathbb{J}$ . At each operation timeslot  $t$ , each customer  $i$  is

served exclusively from one company, while the power company selection of each customer  $i$  can vary for different timeslots. In this work, we assume that the power companies are able to cover customers' demands, thus  $e_{i,j}^{(t)} \leq d_i^{(t)}$ ,  $\forall i \in \mathbb{I}, \forall j \in \mathbb{J}, \forall t \in \mathbb{T}$ .

We denote as  $x_{a,i}^{(t)}$  the demand of customer's  $i$  appliance  $a \in A_i$  for the timeslot  $t$ , and  $x_{a,i,j}^{(t)}$  the corresponding electricity consumption of customer's  $i$  appliance  $a$ ,  $a \in A_i$  from the  $j^{th}$  power company. Then, the shiftable and non-shiftable electricity consumption of customer  $i$  in timeslot  $t$  from  $j^{th}$  power company are determined as  $X_{s,i,j}^{(t)} = \sum_{a \in A_{s,i}} x_{a,i,j}^{(t)}$  and  $X_{ns,i,j}^{(t)} = \sum_{a \in A_{ns,i}} x_{a,i,j}^{(t)}$ . Thus, it is concluded that  $e_{i,j}^{(t)} = X_{s,i,j}^{(t)} + X_{ns,i,j}^{(t)} \leq X_{s,i}^{(t)} + X_{ns,i}^{(t)} = d_i^{(t)}$ , where  $X_{s,i}^{(t)} = \sum_{a \in A_{s,i}} x_{a,i}^{(t)}$  and  $X_{ns,i}^{(t)} = \sum_{a \in A_{ns,i}} x_{a,i}^{(t)}$ .

At each operation timeslot  $t$ , every customer  $i$  aims at satisfying its needs for electricity consumption, while giving higher priority to its non-shiftable appliances' electricity needs  $X_{ns,i}^{(t)}$ . It is noted that in a competitive market, as the one assumed here, though the customer  $i$  requests and buys electricity from a power company  $j$ , it should also consider the total electricity consumption of the rest of the customers, i.e.,  $E_{-i}^{(t)} = \sum_{j \in \mathbb{J}} \sum_{i' \in \mathbb{I}, i' \neq i} e_{i',j}^{(t)}$ , in the current timeslot  $t$ , as the electricity consumption of the rest of the customers in the Smart Grid Network contributes to the configuration of the prices announced by the power companies, as it is presented in the following subsection. This key feature is one of the essentials elements of this work, which differentiate it from similar research work, where each customer's utility function has been formulated considering only its personal electricity consumption. Each customer  $i$  is informed about the total electricity consumption  $E^{(t)} = \sum_{j \in \mathbb{J}} \sum_{i \in \mathbb{I}} e_{i,j}^{(t)}$  in the Smart Grid Network via the centralized SP and through the communication network. As a result, each customer  $i$  is able to deduct its personal consumption  $e_{i,j}^{(t)}$ , i.e.,  $E_{-i}^{(t)} = E^{(t)} - e_{i,j}^{(t)}$ , and no privacy issues are related to this broadcasted information (i.e.,  $E^{(t)}$ ) by the SP, since each customer's consumption  $e_{i,j}^{(t)}$  is hidden with the total consumption.

Each customer's  $i$  satisfaction function is formulated as an increasing concave

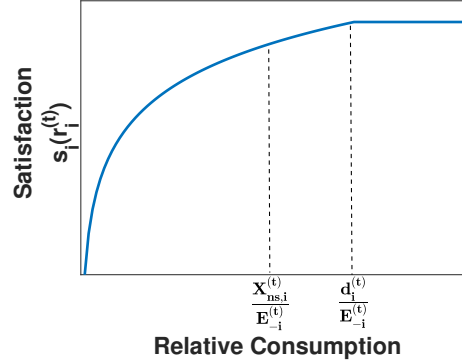


Figure 5.2: Customer's Satisfaction Function

function  $s_i(r_i^{(t)})$  with respect to the relative customer's consumption, i.e.,  $r_i^{(t)} = \frac{e_{i,j}^{(t)}}{E_{-i}^{(t)}}$ . As Figure 5.2 demonstrates, customer's  $i$  satisfaction increases rapidly till its relative non-shiftable consumption, i.e.,  $\frac{X_{ns,i}^{(t)}}{E_{-i}^{(t)}}$ , is satisfied, while after that point its satisfaction increases slowly till it fulfils its relative shiftable electricity needs, i.e.,  $\frac{X_{s,i}^{(t)}}{E_{-i}^{(t)}}$ . Also, for values greater than its overall relative consumption, i.e.,  $\frac{d_i^{(t)}}{E_{-i}^{(t)}}$ , its satisfaction is saturated, because there are no other real needs to cover via consuming additional electricity.

Here, without loss of generality, we adopt a logarithmic customer's satisfaction function with respect to its relative electricity consumption, as:

$$s_i(r_i^{(t)}) = s_i(e_{i,j}^{(t)}, \mathbf{e}_{-i}^{(t)}) = k \cdot \log(1 + \lambda \cdot r_i^{(t)}) \quad (5.1)$$

where  $\mathbf{e}_{-i}^{(t)}$  denotes the vector of all customers' electricity consumption excluding customer  $i$ , and the parameters  $k, \lambda \in R^+$  determine the slope of the concave function to reflect its priority to fulfill its relative non-shiftable consumption prerequisites.

Furthermore, another major novelty introduced in this work is the proposal of a relatively fair pricing policy for the customers that is applied by the power companies that exist in the Smart Grid Network. Specifically, the power companies charge each customer  $i$  based on its relative electricity consumption, i.e.,  $r_i^{(t)} = \frac{e_{i,j}^{(t)}}{E_{-i}^{(t)}}$ , and not based only on its overall consumption  $e_{i,j}^{(t)}$ . Based on this pricing policy, the



power companies provide the incentive even to the low budget customers to buy an affordable amount of electricity in terms of cost, thus still satisfying, while limiting the high budget customers' greedy behavior who aim to dominate the Smart Grid Network. As a result, the benefits of the proposed fair pricing policy are two-fold:

1. customers are satisfied due to fair charges of electricity consumption
2. the power companies attract more customers, thus increase their profit in a long-term period and improve their penetration in the market.

This fair pricing policy for each customer  $i$  based on its relative consumption  $r_i^{(t)}$ , is formulated as:

$$FPP_i(r_i^{(t)}) = FPP(e_{i,j}^{(t)}, \mathbf{e}_{-i}^{(t)}) = \gamma_i^{(t)} \cdot r_i^{(t)} \cdot p_j^{(t)} \quad (5.2)$$

where,  $p_j^{(t)}[\frac{\$}{KWh}]$  is the price that is announced by the power company  $j, j \in \mathbb{J}$  for the timeslot  $t, t \in \mathbb{T}$ , and  $\gamma_i^{(t)}$  is a time-varying parameter capturing the dynamics of customer's  $i$  behavior, i.e., smaller  $\gamma_i^{(t)}$  reflects customer's  $i$  dynamic behavior to spend money in order to buy more electricity.

Finally, each customer's  $i, i \in \mathbb{I}$  utility function is formulated via capturing its satisfaction, i.e.,  $s_i(r_i^{(t)})$  with respect to its relative electricity consumption, as well as its dissatisfaction due to the associated charges (i.e., pricing), as follows:

$$\begin{aligned} U_i^{(t)}(e_{i,j}^{(t)}, \mathbf{e}_{-i}^{(t)}, \mathbf{p}^{(t)}) &= s_i(r_i^{(t)}) - FPP_i(r_i^{(t)}) \\ &= s_i(e_{i,j}^{(t)}, \mathbf{e}_{-i}^{(t)}) - FPP_i(e_{i,j}^{(t)}, \mathbf{e}_{-i}^{(t)}) \\ &= k \cdot \log(1 + \lambda \cdot r_i^{(t)}) - \gamma_i^{(t)} \cdot r_i^{(t)} \cdot p_j^{(t)} \end{aligned} \quad (5.3)$$

where  $\mathbf{p}^{(t)} = (p_1^{(t)}, \dots, p_j^{(t)}, \dots, p_J^{(t)})$  denotes the vector of the announced prices by the power companies in timeslot  $t, t \in \mathbb{T}$ .

### Welfare and Characteristics of Power Companies

Each power company  $j, j \in \mathbb{J}$  generates an amount  $g_j^{(t)}[KWh]$  of electricity units per timeslot  $t$ , while the generation cost of each electricity unit from the  $j^{th}$  power

company in timeslot  $t$  is  $c_j^{(t)}[\frac{\$}{kWh}]$ . In this work, we assume that each power company  $j$  is able to generate the overall needed amount of electricity, thus  $g_j^{(t)} = E_j^{(t)} = \sum_{i \in \mathbb{I}} e_{i,j}^{(t)}$ . The peak customers' electricity consumption in the  $j^{th}$  power company is  $E_{P_j} = \max_{t \in \mathbb{T}} E_j^{(t)} = \max_{t \in \mathbb{T}} \sum_{i \in \mathbb{I}} e_{i,j}^{(t)}$ , while the corresponding average consumption over  $T$  operation timeslots in the  $j^{th}$  power company is  $E_{avg_j} = \frac{\sum_{t \in \mathbb{T}} E_j^{(t)}}{T} = \frac{\sum_{t \in \mathbb{T}} \sum_{i \in \mathbb{I}} e_{i,j}^{(t)}}{T}$ . Moreover, using the peak customers' electricity consumption  $E_{P_j}$  and the corresponding average consumption  $E_{avg_j}$ , we define the peak-to-average (PAR) ratio in customers' electricity consumption of the  $j^{th}$  power company as  $PAR_j = \frac{E_{P_j}}{E_{avg_j}}$ .

Each power company aiming to achieve a low peak-to-average ratio power consumption, so as to maintain the smooth electricity generation during the day. Also, customers prefer to be served by companies that maintain low peak-to-average ratios, as through this way they "feel" more "safe" that they will be satisfied effectively and fulfil their electricity requirements.

A fairly effective way for the power companies to maintain low peak-to-average ratio, is to provide incentives to the customers to shift their consumption from high-peak to off-peak for specific periods of the day. Moreover, the power companies could benefited by the policy of announcing discounts to the customers, regarding their billing prices. Through this way, it is able the electricity needs among the customers to be balanced, and as a result the power companies to maintain low peak-to-average ratio, and at the same time the announcements of discounts to provide incentives to the customers to select the power company, which will result in a long-term improvement of the power company's profit.

Discount strategy is a fairly common technique with which companies manage to win more customers and improve their profit in a long term period. The effectiveness of this technique has already been studied in literature and has been applied in several different fields, as in tourism. The power companies of the Smart Grid Network by studying and analyzing the consumption habits of the customers, are able to

determine the most appropriate and effective discounts for their electricity prices.

Each power company is interested to increase its reputation and competitiveness in the electricity market. In a nutshell, each power company's reputation increases as the total price discounts, i.e.,  $\sum_{t \in \mathbb{T}} f_j^{(t)}$ , offered to the customers increases, throughout the day, as well as if the company maintains low peak-to-average ratio. In this work, we formulate the competitiveness of each power company  $j, j \in \mathbb{J}$  via its penetration to the electricity market, which is translated to the electricity consumption served by the  $j^{th}$  power company over the total electricity consumption in the Smart Grid Network, i.e.,  $Comp_j = \frac{\sum_{t \in \mathbb{T}} E_j^{(t)}}{E^{(t)}} = \frac{\sum_{t \in \mathbb{T}} \sum_{i \in \mathbb{I}} e_{i,j}^{(t)}}{\sum_{j \in \mathbb{J}} \sum_{t \in \mathbb{T}} \sum_{i \in \mathbb{I}} e_{i,j}^{(t)}}$

Consequently, each power company  $j, j \in \mathbb{J}$  is characterized by a reputation and competitiveness score  $RC_j$ , which is considered by the customers throughout the power company selection process, and is formulated as follows:

$$RC_j = \sum_{t \in \mathbb{T}} f_j^{(t)} \cdot \frac{1}{PAR_j} \cdot Comp_j \quad (5.4)$$

where,  $f_j^{(t)}$  is the discount that is announced by the power company  $j, j \in \mathbb{J}$  to the customers during the timeslot  $t, t \in \mathbb{T}$ .

The profit of each power company is constructed by considering the revenue and the costs of the power company by billing its customers and generating the needed electricity, respectively. Specifically, each power company's  $j, j \in \mathbb{J}$  profit function is formulated as follows:

$$P_j^{(t)}(E_j^{(t)}, p_j^{(t)}) = R_j^{(t)}(E_j^{(t)}, p_j^{(t)}) - C_j^{(t)}(E_j^{(t)}) \quad (5.5)$$

where,  $R_j^{(t)}$  and  $C_j^{(t)}$  express the revenue and the generation cost of the  $j^{th}$  power company, respectively. The power company's  $j$  revenue  $R_j^{(t)}$  per timeslot  $t$  depends on the amount of sold electricity to the customers that selected to be served by the specific company, i.e.,  $E_j^{(t)} = \sum_{i \in \mathbb{I}} e_{i,j}^{(t)}$ , the company's price  $p_j^{(t)}$  per electricity unit, and the discount  $f_j^{(t)}$  that the company announces to the customers on that timeslot. As

a result, the power company's  $j$  revenue is formulated as:

$$\begin{aligned} R_j^{(t)}(E_j^{(t)}, p_j^{(t)}) &= (1 - f_j^{(t)}) \cdot p_j^{(t)} \cdot \sum_{i \in \mathbb{I}} e_{i,j}^{(t)} \\ &= (1 - f_j^{(t)}) \cdot p_j^{(t)} \cdot \sum_{i \in \mathbb{I}} \sum_{a \in A_i} x_{a,i,j}^{(t)} \end{aligned} \quad (5.6)$$

On the other hand, the power company's  $j$  cost for generating the overall amount of electricity that the customers demand, is expressed as:

$$C_j^{(t)}(E_j^{(t)}) = c_j^{(t)} \cdot E_j^{(t)} = c_j^{(t)} \cdot \sum_{i \in \mathbb{I}} e_{i,j}^{(t)} \quad (5.7)$$

where  $c_j^{(t)}$  denotes the power company's  $j$  electricity production cost per unit of electricity for the timeslot  $t$ .

#### 5.1.4 Modeling of the Smart Grid Network as a Distributed Learning System

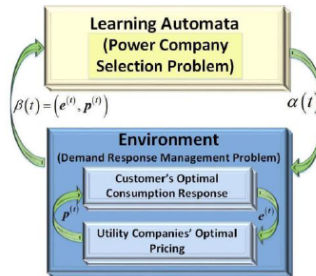


Figure 5.3: Smart Grid Network as a Learning System

Power companies build their reputation and competitiveness for a long time to attract more customers and increase their profits. On the other hand, each company's reputation and competitiveness factor contribute significantly to the customers' choices regarding the power company that they select to be served by. Consequently, the Smart Grid Network can be studied as a learning system, where the customers act as RL agents, i.e., stochastic learning automata that interact with

the environment and determine an optimal policy of selecting a power company. Figure 5.3 presents the Smart Grid Network as a learning system and the relationship between the learning automata and the environment. Specifically, each customer/learning automaton at each operation timeslot  $t$  has an action vector  $\alpha_i(\mathbf{t}) = (\alpha_i^1, \dots, \alpha_i^j, \dots, \alpha_i^J)$ , where  $\sum_{j \in \mathbb{J}} \alpha_i^j = 1$ , thus the action vector  $\alpha_i(\mathbf{t})$  represents the customer's  $i$  power company selection for the timeslot  $t$  (Section 2.2.3).

Towards making their decision, the learning automata consider the output set  $\beta(t) = (\mathbf{e}^{(t)}, \mathbf{p}^{(t)})$ , i.e.,  $\mathbf{e}^{(t)}$  is the vector of all customers' electricity consumption, and  $\mathbf{p}^{(t)}$  the pricing vector that contains the power companies' prices, as this is determined by solving the Demand Response Management problem, which is analyzed in Chapter 5.1.1. The solution of the Demand Response Management problem refers to customers' and companies' optimal electricity consumption and prices, respectively. Based on the learning automata chosen actions and the corresponding reaction of the environment, the reward probability  $r_j(t)$  that is associated with the power company that the customer selected to be served by, is obtained as  $r_j(t) = \frac{RC_j}{\sum_{j \in \mathbb{J}} RC_j}$ , thus  $0 \leq r_j(t) \leq 1, \forall j \in \mathbb{J}$ . Essentially, the reward probability  $r_j(t)$  updates with a higher or a lower probability the customer's selection, regarding the power company  $j$  that was selected and with which its reward probability  $r_j(t)$  is associated with. The action probability vector of customer  $i$  is defined as  $\mathbf{Pr}_i(\mathbf{t}) = (Pr_{i,1}(t), \dots, Pr_{i,j}(t), \dots, Pr_{i,J}(t))$ , where  $Pr_{i,j}$  represents the probability of the customer  $i$  to select the power company  $j$  for the timeslot  $t$ . Each customer's  $i$  probability vector is updated based on the concept of Stochastic Learning Automata, and the update rules, as these were introduced in Section 2.2.3, are formulated as follows:

$$Pr_{i,j}(t+1) = Pr_{i,j}(t) - b \cdot \frac{RC_j}{\sum_{j \in \mathbb{J}} RC_j} \cdot Pr_{i,j}(t), \text{ if } j^{(t+1)} \neq j^{(t)} \quad (5.8)$$

$$Pr_{i,j}(t+1) = Pr_{i,j}(t) + b \cdot \frac{RC_j}{\sum_{j \in \mathbb{J}} RC_j} \cdot (1 - Pr_{i,j}(t)), \text{ if } j^{(t+1)} = j^{(t)} \quad (5.9)$$

where  $0 < b < 1$  is the learning step parameter that controls the convergence and the complexity of the learning algorithm. Essentially, Equation 5.8 represents customer's selection probability update rule for the next timeslot for the company that was selected, while Equation 5.9 represents the update rule that is followed for the rest selection probabilities of the customer, thus for the ones that are associated with the rest power companies. In that way, the customer acting as a learning automaton, increases its probability of selecting the same power company  $j$  based on the achievable reward probability  $r_j(t)$  of that company, thus the customer explores its environment and converges to the power company that provides a good reward (i.e., reputation score).

It should be noted that initially, the overall Smart Grid Network needs no prior knowledge of the reward and action probabilities, and thus the initial power company selection by the users can be simply assumed as  $Pr_{i,j}(t) = \frac{1}{J}, \forall j \in \mathbb{J}$ . The customers, in a long-term period converge to the most cost-efficient solution of power company selection per operation timeslot  $t$ , given also that the overall policies of the power companies (i.e.,  $c_j^{(t)}, p_j^{(t)}, f_j^{(t)}, \forall j \in \mathbb{J}, \forall t \in \mathbb{T}$ ) do not change rapidly within a long time period. Finally, it is also highlighted that other learning techniques, such as exponential learning, Q-learning, etc., could be also adopted instead of the learning automata approach that was selected in that work due to the scalable and low-complexity nature.

### 5.1.5 Demand Response Management: Problem Formulation & Solution

The Demand Response Management (DRM) problem is formulated considering the iterations and interactions of both the power companies and the customers. Before the DRM problem, the customers have already selected the power companies that they want to serve by, based on their stochastic learning methodology described

in Section 5.1.4. Each power company  $j, j \in \mathbb{J}$  aims at maximizing its profit (i.e., Equation 5.5), by considering the customers' electricity consumption, and it aims to converge to the optimal announced price  $p_j^{(t)*}$  per timeslot  $t, t \in \mathbb{T}$ . On the other hand, each customer's  $i, i \in \mathbb{I}$  goal is to maximize its utility function (i.e., Equation 5.3), given the announced electricity prices by the power companies, and determine in a distributed manner its optimal electricity consumption  $e_{i,j}^{(t)*}$ . The distributed nature in determining both the optimal prices  $p_j^{(t)*}, \forall j \in \mathbb{J}$ , and each customer's optimal consumption  $e_{i,j}^{(t)*}, \forall i \in \mathbb{I}$  is a key component in the formulation and solution of the DRM problem to support the vision of independent and deregulated electricity markets, where no centralized entity is required, as both the customers and the power companies act as distributed decision-makers.

Each power company's  $j, j \in \mathbb{J}$ , and each customer's  $i, i \in \mathbb{I}$  DRM optimization problem, is formulated as follows:

$$p_j^{(t)*} = \arg \max_{p_j^{(t)}} \left[ \begin{aligned} &P_j^{(t)}(E_j^{(t)}, p_j^{(t)}) = R_j^{(t)}(E_j^{(t)}, p_j^{(t)}) - C_j^{(t)}(E_j^{(t)}) \\ &= (1 - f_j^{(t)}) \cdot p_j^{(t)} \cdot \sum_{i \in \mathbb{I}} e_{i,j}^{(t)} - c_j^{(t)} \cdot \sum_{i \in \mathbb{I}} e_{i,j}^{(t)} \end{aligned} \right] \quad (5.10)$$

$$e_{i,j}^{(t)*} = \arg \max_{e_{i,j}^{(t)}} \left[ \begin{aligned} &U_i^{(t)}(e_{i,j}^{(t)}, \mathbf{e}_{-\mathbf{i}}^{(t)}, \mathbf{p}^{(t)}) = s_i(r_i^{(t)}) - FPP_i(r_i^{(t)}) \\ &= s_i^{(t)}(e_{i,j}^{(t)}, \mathbf{e}_{-\mathbf{i}}^{(t)}) - FPP_i(e_{i,j}^{(t)}, \mathbf{e}_{-\mathbf{i}}^{(t)}) \\ &= k \cdot \log(1 + \lambda r_i^{(t)}) - \gamma_i^{(t)} \cdot r_i^{(t)} \cdot p_j^{(t)} \end{aligned} \right] \quad (5.11)$$

As Equation 5.10 and Equation 5.11 depict, the decisions about the optimal prices by the power companies and the optimal electricity consumption by the customers are interconnected problems, as the decision of the one (i.e., power companies) should act as an input to the other (i.e., customers) and vice versa. As a result, the DRM problem is studied as a two-stage game, where at the first stage, the optimal electricity consumption of the customers is determined via formulating the maximization problem of their utilities (i.e., Equation 5.11) as a non-cooperative game among the customers. At the second stage, each power company, given the optimal electricity consumption of the customers, determines its optimal pricing policy that maximizes

its profit (i.e., Equation 5.10). The interaction and feedback among power companies and customers endure until both conclude to their optimal decisions.

### Customers' Optimal Consumption Response

In the first stage of the DRM problem, each customer  $i, i \in \mathbb{I}$  determines its optimal electricity consumption for timeslot  $t, t \in \mathbb{T}$ , by considering its power company selection and the announced price by the corresponding company. We define as  $\mathbb{G} = [\mathbb{I}, \{S_i^{(t)}\}, \{U_i^{(t)}\}]$  the non-cooperative consumption response game among the customers, which consists of the infinite set of customers  $\mathbb{I} = \{1, \dots, i, \cdot, I\}$ , the strategy space  $S_i^{(t)} = [0, d_i^{(t)}]$  of each customer  $i, \forall i \in \mathbb{I}$  and its utility function  $U_i^{(t)}$ . The non-cooperative consumption response game  $\mathbb{G}$  can be expressed as follows:

$$\begin{aligned} \max_{e_{i,j}^{(t)} \in S_i} \quad & \begin{bmatrix} U_i^{(t)} = s_i(r_i^{(t)}) - FPP_i(r_i^{(t)}) \\ = k \cdot \log(1 + \lambda r_i^{(t)}) - \gamma_i^{(t)} \cdot r_i^{(t)} \cdot p_j^{(t)} \end{bmatrix} \\ \text{s.t.} \quad & 0 \leq e_{i,j}^{(t)} \leq d_i^{(t)} \end{aligned} \quad (5.12)$$

Here, we adopt the concept of the Nash Equilibrium (NE), as this was introduced in Section 2.1, at which no customer can improve its utility by unilaterally changing its electricity consumption. Towards proving the existence and uniqueness of the NE of the non-cooperative game  $\mathbb{G}$ , it suffices to show that for every timeslot  $t, t \in \mathbb{T}$ , each customer's  $i$  strategy space  $S_i^{(t)}$  is a non-empty, convex and compact subset of the Euclidean space  $R^I$ , and the utility function  $U_i^{(t)}(e_{i,j}^{(t)}, \mathbf{e}_{-i}^{(t)}, p_j^{(t)})$  is continuous in  $e_{i,j}^{(t)}$  and quasi-concave in  $S_i^{(t)}$  as explained in [301].

**Theorem 13** *In the non-cooperative consumption response game  $\mathbb{G}$ , customer's  $i$  best response strategy to a given electricity consumption vector  $\mathbf{e}_{-i}^{(t)}$  is  $BR_i(\mathbf{e}_{-i}^{(t)}) = e_{i,j}^{(t)*}$ , as provided in Equation 5.13, where  $s_i'^{-1}$  is the inverse function of the first derivative of the customer's  $i$  satisfaction function  $s_i$ , and  $\tau = \lim_{r_i^{(t)} \rightarrow \infty} s_i'^{-1}$*



$$BR_i(\mathbf{e}_{-i}^{(t)}) = e_{i,j}^{(t)*} = \begin{cases} d_i^{(t)} & \text{if } 0 \leq \gamma_i^{(t)} \cdot p_j^{(t)} \leq \tau \\ \min\{d_i^{(t)}, E_{-i}^{(t)} \cdot s_i'^{-1}(0)\} & \text{if } \tau < \gamma_i^{(t)} \cdot p_j^{(t)} \leq s_i'^{-1}(0) \\ 0 & \text{if } \gamma_i^{(t)} \cdot p_j^{(t)} > s_i'^{-1}(0) \end{cases} \quad (5.13)$$

*Proof:* Towards determining customer's best response strategy  $BR_i(\mathbf{e}_{-i}^{(t)}) = e_{i,j}^{(t)*}$ , the first and the second order derivatives of customer's utility function  $U_i^{(t)}$  with respect to  $e_{i,j}^{(t)}$  are used:

$$\frac{\partial U_i^{(t)}(e_{i,j}^{(t)}, \mathbf{e}_{-i}^{(t)}, p_j^{(t)})}{\partial e_{i,j}^{(t)}} = \frac{1}{E_{-i}^{(t)}} \cdot [s_i'(r_i^{(t)}) - \gamma_i^{(t)} \cdot p_j^{(t)}] \quad (5.14)$$

$$\frac{\partial^2 U_i^{(t)}(e_{i,j}^{(t)}, \mathbf{e}_{-i}^{(t)}, p_j^{(t)})}{\partial e_{i,j}^{(t)2}} = \frac{1}{(E_{-i}^{(t)})^2} \cdot s_i''(r_i^{(t)}) \quad (5.15)$$

Each customer's satisfaction function  $s_i^{(t)}(r_i^{(t)})$  is an increasing concave function with respect to  $r_i^{(t)}$ , thus  $s_i''(r_i^{(t)}) < 0$  and  $\frac{\partial^2 U_i^{(t)}(e_{i,j}^{(t)}, \mathbf{e}_{-i}^{(t)}, p_j^{(t)})}{\partial e_{i,j}^{(t)2}} < 0$ . We set  $\tau = \lim_{r_i^{(t)} \rightarrow \infty} s_i'^{-1}$ . Since  $s_i'(r_i^{(t)})$  is a strictly decreasing function (due to  $s_i''(r_i^{(t)}) < 0$ ) and as  $s_i'(r_i^{(t)}) > 0$ , we know that  $\tau < s_i'(r_i^{(t)}) \leq s_i'(0)$  and  $0 \leq \tau < s_i'(0)$ . Hence, for  $0 \leq \gamma_i^{(t)} \cdot p_j^{(t)} \leq \tau$ , we have  $\frac{\partial U_i^{(t)}(e_{i,j}^{(t)}, \mathbf{e}_{-i}^{(t)}, p_j^{(t)})}{\partial e_{i,j}^{(t)}} > 0$  and thus  $U_i^{(t)}$  is an increasing function of  $e_{i,j}^{(t)}$ . In this case, the best response strategy for customer  $i, i \in \mathbb{I}$  is to demand its maximum electricity consumption, i.e.,  $d_i^{(t)}$ . So, for  $0 \leq \gamma_i^{(t)} \cdot p_j^{(t)} \leq \tau$ , we have  $BR_i(\mathbf{e}_{-i}^{(t)}) = d_i^{(t)}$ ,  $\forall i \in \mathbb{I}$ . For  $\tau < \gamma_i^{(t)} \cdot p_j^{(t)} \leq s_i'(0)$ , the equation  $\frac{\partial U_i^{(t)}(e_{i,j}^{(t)}, \mathbf{e}_{-i}^{(t)}, p_j^{(t)})}{\partial e_{i,j}^{(t)}} = 0$  is equivalent to  $s_i'(r_i^{(t)}) = \gamma_i^{(t)} \cdot p_j^{(t)} \Leftrightarrow \hat{r}_i^{(t)} = s_i'^{-1}(\gamma_i^{(t)} \cdot p_j^{(t)})$ ,  $\forall i \in \mathbb{I}$ . Note that as  $s_i'(r_i^{(t)})$  is a strictly decreasing function, its inverse (i.e.,  $s_i'^{-1}$ ) exists, and that  $\hat{r}_i^{(t)}$  is a decreasing function of  $\gamma_i^{(t)} \cdot p_j^{(t)}$ . Since  $s_i''(r_i^{(t)}) < 0$  for all  $r_i^{(t)}$  and hence  $\frac{\partial^2 U_i^{(t)}(e_{i,j}^{(t)}, \mathbf{e}_{-i}^{(t)}, p_j^{(t)})}{\partial e_{i,j}^{(t)2}} < 0$ , the roots of  $\frac{\partial U_i^{(t)}(e_{i,j}^{(t)}, \mathbf{e}_{-i}^{(t)}, p_j^{(t)})}{\partial e_{i,j}^{(t)}} = 0$  maximize  $U_i^{(t)}(e_{i,j}^{(t)}, \mathbf{e}_{-i}^{(t)}, p_j^{(t)})$  for the given electricity consumption of the rest users, i.e.,  $\mathbf{e}_{-i}^{(t)}$ . An one-to-one relation exists between  $r_i^{(t)}$  and  $e_{i,j}^{(t)}$ , and thus the best response electricity consumption in response to  $\mathbf{e}_{-i}^{(t)}$  that maximizes  $U_i^{(t)}(e_{i,j}^{(t)}, \mathbf{e}_{-i}^{(t)}, p_j^{(t)})$  is also unique and is equal to  $e_{i,j}^{(t)*} = E_{-i}^{(t)} \cdot \hat{r}_i^{(t)} = E_{-i}^{(t)} s_i'^{-1}(\gamma_i^{(t)} \cdot p_j^{(t)})$ . If  $e_{i,j}^{(t)*} > d_i^{(t)}$  customer  $i, i \in \mathbb{I}$  does not request for  $e_{i,j}^{(t)*}$ . In this case, since  $e_{i,j}^{(t)*}$  is the

unique maximizer of  $U_i^{(t)}$ , then  $U_i^{(t)}$  is an increasing function of  $e_{i,j}^{(t)}$  in  $e_{i,j}^{(t)} \leq d_i^{(t)} \leq e_{i,j}^{(t)*}$  for fixed  $\mathbf{e}_{-i}^{(t)}$ . Therefore, the best response to  $\mathbf{e}_{-i}^{(t)}$  is the maximum value of customer's electricity consumption, i.e.,  $BR_i(\mathbf{e}_{-i}^{(t)}) = d_i^{(t)}$ . This implies that for  $\tau < \gamma_i^{(t)} \cdot p_j^{(t)} \leq s_i'(0)$ ,  $BR_i(\mathbf{e}_{-i}^{(t)}) = \min\{d_i^{(t)}, E_{-i}^{(t)} \cdot s_i'^{-1}(\gamma_i^{(t)} \cdot p_j^{(t)})\}$ . For  $\gamma_i^{(t)} \cdot p_j^{(t)} > s_i'(0)$ , we have  $\frac{\partial U_i^{(t)}}{\partial e_{i,j}^{(t)}} < 0$ , thus  $U_i^{(t)}$  is a decreasing function of  $e_{i,j}^{(t)}$ . In this case, the imposed price by the companies is extremely high for customers to afford it, thus  $BR_i(\mathbf{e}_{-i}^{(t)}) = 0$ ,  $\forall i \in \mathbb{I}$ .

Based on Theorem 13 that determines each customer's  $i, i \in \mathbb{I}$  best responses strategy  $BR_i(\mathbf{e}_{-i}^{(t)}) = e_{i,j}^{(t)}$  and considering the quasi-concavity property with respect to  $r_i^{(t)}$  of customer's utility function  $U_i^{(t)}$ , the existence and uniqueness of the NE of the non-cooperative game  $\mathbb{G}$  is derived as follows:

**Theorem 14** *The Nash Equilibrium of the non-cooperative consumption response game  $\mathbb{G}$  exists and is unique.*

*Proof:* The NE is by definition the fixed point in the best response function set that satisfies  $e_{i,j}^{(t)*} = BR_i(\mathbf{e}_{-i}^{(t)})$ . In the two cases, where  $0 \leq \gamma_i^{(t)} \cdot p_j^{(t)} \leq \tau$  and  $\gamma_i^{(t)} \cdot p_j^{(t)} > s_i'(0)$ , the fixed point of the best response function set is unique, i.e., maximum electricity consumption, i.e.,  $e_{i,j}^{(t)*} = d_i^{(t)}$  or no consumption, i.e.,  $e_{i,j}^{(t)*} = 0$ , respectively. In the third case, where  $\tau < \gamma_i^{(t)} \cdot p_j^{(t)} \leq s_i'(0)$ , the uniqueness of the NE point can be proved via adopting the concept of standard function [301], [302]. A function  $f(x)$  is characterized as standard if it satisfies the following properties [223]:

1. Positivity:  $f(x) > 0$
2. Monotonicity: if  $x \geq x'$ , then  $f(x) > f(x')$
3. Scalability:  $\forall a > 1, a \cdot f(x) \geq f(a \cdot x)$

If a fixed point exists in a standard function, then it is unique [301], [302]. As it is shown in [223],  $e_{i,j}^{(t)*} = BR_i(\mathbf{e}_{-i}^{(t)})$  for  $\tau < \gamma_i^{(t)} p_j^{(t)} \leq s_i'(0)$  (i.e., Equation 5.13) can easily

be shown that it is a standard function. Thus, in the case that  $\tau < \gamma_i^{(t)} \cdot p_j^{(t)} \leq s'_i(0)$  the NE exists and is unique.

Finally, as we have already mentioned, the customers' optimal electricity consumption, as this is determined in Equation 5.13 will act as input to the optimal pricing problem, where each power company determines the optimal price.

### Companies' Optimal Pricing Response

In the first stage of the DRM optimization problem, the optimal electricity of each customer was determined, while in the second stage each power company aims to maximize its profit (i.e., Equation 5.5) in a distributed manner, via calculating the optimal price to be announced. Combining Equation 5.10, 5.13, the optimal pricing problem based on customers' optimal consumption response can be written as follows [209].

$$p_j^{(t)*} = \arg \max_{p_j^{(t)}} \left[ \begin{aligned} &P_j^{(t)} = R_j^{(t)}(E_j^{(t)}, p_j^{(t)}) - C_j^{(t)}(E_j^{(t)}) \\ &= (1 - f_j^{(t)}) \cdot p_j^{(t)} \cdot \sum_{i \in \mathbb{I}} \left[ E_{-i}^{(t)} \cdot \left( \frac{k}{\gamma_i^{(t)} p_j^{(t)}} - \frac{1}{\lambda} \right) \right] \\ &- c_j^{(t)} \sum_{i \in \mathbb{I}} \left[ E_{-i}^{(t)} \left( \frac{k}{\gamma_i^{(t)} p_j^{(t)}} - \frac{1}{\lambda} \right) \right] \end{aligned} \right] \quad (5.16)$$

The optimal pricing problem in response to customers' consumption, as it is rewritten in Equation 5.16, is a function only of power company's price.

**Theorem 15** *Each power company's  $j, j \in \mathbb{J}$  optimal price that maximizes its profit, given customers' optimal response consumption, is given as:*

$$p_j^{(t)*} = \left[ \frac{k \cdot \lambda \cdot c_j^{(t)} \cdot \sum_{i \in \mathbb{I}} \frac{E_{-i}^{(t)}}{\gamma_i^{(t)}}}{(1 - f_j^{(t)}) \cdot \sum_{i \in \mathbb{I}} E_{-i}^{(t)}} \right]^{\frac{1}{2}} \quad (5.17)$$

*Proof:* Given customers' optimal consumption response that is determined in the first stage of the DRM optimization problem, the profit function of each power company

$j, j \in \mathbb{J}$  is written as follows:

$$P_j^{(t)}(E_j^{(t)}, p_j^{(t)}) = (1 - f_j^{(t)}) \cdot p_j^{(t)} \cdot \sum_{i \in \mathbb{I}} \left[ E_{-i}^{(t)} \left( \frac{k}{\gamma_i^{(t)} \cdot p_j^{(t)}} - \frac{1}{\lambda} \right) \right] - c_j^{(t)} \cdot \sum_{i \in \mathbb{I}} \left[ E_{-i}^{(t)} \left( \frac{k}{\gamma_i^{(t)} \cdot p_j^{(t)}} - \frac{1}{\lambda} \right) \right] \quad (5.18)$$

Considering the first order derivative of  $P_j^{(t)}$  with respect to  $p_j^{(t)}$ , we have:

$$\frac{\partial P_j^{(t)}(E_{-i}^{(t)}, p_j^{(t)})}{\partial p_j^{(t)}} = -\frac{1}{\lambda} (1 - f_j^{(t)}) \cdot \sum_{i \in \mathbb{I}} E_{-i}^{(t)} + \frac{c_j^{(t)} \cdot k}{p_j^{(t)^2}} \cdot \sum_{i \in \mathbb{I}} \frac{E_{-i}^{(t)}}{\gamma_i^{(t)}} \quad (5.19)$$

As a result, the critical points of the profit function  $P_j^{(t)}(E_{-i}^{(t)}, p_j^{(t)})$  are as follows:

$$p_j^{(t)*} = \left[ \frac{k \cdot \lambda \cdot c_j^{(t)} \cdot \sum_{i \in \mathbb{I}} \frac{E_{-i}^{(t)}}{\gamma_i^{(t)}}}{(1 - f_j^{(t)}) \cdot \sum_{i \in \mathbb{I}} E_{-i}^{(t)}} \right] \quad (5.20)$$

The second order derivative of the profit function  $P_j^{(t)}$  is given as follows:

$$\frac{\partial^2 P_j^{(t)}(E_{-i}^{(t)}, p_j^{(t)})}{\partial p_j^{(t)^2}} = -2 \cdot c_j^{(t)} \cdot \frac{k}{p_j^{(t)^3}} \cdot \sum_{i \in \mathbb{I}} \frac{E_{-i}^{(t)}}{\gamma_i^{(t)}} \quad (5.21)$$

As observed by Equation 5.21, it holds true that  $\frac{\partial^2 P_j^{(t)}(E_{-i}^{(t)}, p_j^{(t)})}{\partial p_j^{(t)^2}} < 0$ , thus the  $p_j^{(t)*}$  as determined in Equation 5.20 maximizes the company's  $j, j \in \mathbb{J}$  profit function  $P_j^{(t)}$ .

### 5.1.6 PC-DRM Algorithm

In this section, each step of the PC-DRM algorithm is presented and analyzed, and its pseudo code is presented as well. The steps of the PC-DRM algorithm can be summarized as follows:

1. **Initialization Phase:** At the beginning of the first timeslot, (i.e.,  $t = 0$ ), each customer  $i, i \in \mathbb{I}$  initializes its probability vector by following a normal distribution, thus  $Pr_{i,j}^{(0)} = \frac{1}{J}$ ,  $\forall i \in \mathbb{I}, \forall j \in \mathbb{J}$ . Consequently, each customer chooses a power company according to its initial probability vector  $\mathbf{Pr}_i^{(0)}$ .

2. **Power Company Selection - PC:** At every other timeslot  $t, t \in \mathbb{T}$ , such that  $t > 0$  each customer  $i, i \in \mathbb{I}$  chooses a power company to be served from, according to its probability vector  $\mathbf{Pr}_i^{(t)}$ . If  $\forall i \in \mathbb{I}, \exists j, j \in \mathbb{J}$  such that  $Pr_{i,j}^{(t)} \rightarrow 1$ , then stop. Otherwise,  $ite = ite + 1$ , where  $ite$  denotes the iteration of the DRM part of the algorithm.
3. **Customers' Optimal Consumption Response:** Given that all the customers have selected their company that they will be served from, the power companies announce their prices and the total electricity consumption (i.e.,  $E^{(t)}$ ) in the Smart Grid Network. Each customer  $i, i \in \mathbb{I}$  determines its optimal consumption response based on Equation 5.13, as  $e_{i,j}^{(t)}|_{ite}$
4. **Companies' Optimal Pricing Response:** Given customers' optimal electricity consumption. each power company determines its optimal prices based on Equation 5.16, as  $p_j^{(t)*}|_{ite}$
5. **Checking for Convergence:** if  $|e_{i,j}^{(t)*}|_{ite+1} - e_{i,j}^{(t)*}|_{ite}| \rightarrow 0$  and  $|p_{i,j}^{(t)*}|_{ite} - p_{i,j}^{(t)*}|_{ite+1}| \rightarrow 0, \forall i \in \mathbb{I}, \forall j \in \mathbb{J}$ , then the two-stage non-cooperative game has converged to its NE point. Otherwise go to Step 3.
6. **Stochastic Learning Automata:** Each power company  $j, j \in \mathbb{J}$  determines its reward probability  $r_j^{(t)}$ , and it is broadcasted to the customers. Each customer  $i, i \in \mathbb{I}$  updates its probability vector  $\mathbf{Pr}_i^t$  based on Equation 5.8 and 5.9. Return to Step 2.

The PC-DRM learning distributed algorithm can be summarized as follows:

The PC-DRM distributed algorithm can be characterized as a low complexity algorithm (as it is also confirmed by the numerical results in Section 5.1.7), due to the constant in terms of complex operations that are made both in the customers' and companies' side. Furthermore, due to its low complexity the PC-DRM algorithm can be installed and realized through the customers' smart meters in a real-time

---

**Algorithm 6** PC-DRM Algorithm

---

```

1: Input/Initialization:  $\mathbb{I}, \mathbb{J}, d_i^{(t)}, \gamma_i^{(t)}, c_j^{(t)}, Pr_{i,j}^{(0)} = \frac{1}{J} \forall i \in \mathbb{I}, \forall j \in \mathbb{J}, \forall t \in \mathbb{T}$ 
2: Output: NE point  $\mathbf{e}^{(t)*} = (e_1^{(t)*}, \dots, e_I^{(t)*}), \mathbf{p}^{(t)*} = (p_1^{(t)*}, \dots, p_J^{(t)*}) \forall t \in \mathbb{T}$ 
3: for each timeslot  $t, t \in \mathbb{T}$  do
4:    $Ite = 0, Convergence = 0$ 
5:   Each customer  $i, i \in \mathbb{I}$  selects a power company based on  $\mathbf{Pr}_i^{(t)}$ 
6:   while not  $Convergence$  do
7:      $Ite = Itc + 1$ 
8:     for  $i = 1$  to  $I$  do
9:       Customer  $i$  determines its  $e_{i,j}^{(t)*}$  based on Equation 5.13
10:    end for
11:    for  $j = 1$  to  $J$  do
12:      Power Company  $j$  determines its optimal price  $p_j^{(t)*}$  based on Equation 5.16
13:    end for
14:    if  $(Ite > 0 \ \&\& \ |e_i^{(t)*}|_{ite} - e^{(t)*}|_{ite-1}| \rightarrow 0 \ \&\& \ |p_j^{(t)*}|_{ite} - p^{(t)*}|_{ite-1}| \rightarrow 0, \forall i \in \mathbb{I}, \forall j \in \mathbb{J})$  then
15:       $Convergence = 1$ 
16:    end if
17:  end while
18: end for

```

---

manner, while from the companies' point of view, the proposed algorithm can run at the companies' management and decision-making center. Finally, in Section 5.1.7 it is shown that the action customers' selection probabilities converge fast, something that indicates and confirms the efficiency of the stochastic learning automata methodology, that we propose in this work.

### 5.1.7 Empirical Evaluation

In this Section, a detailed numerical performance evaluation and comparative study of the proposed framework is conducted through modeling and simulations. The results illustrate the operation, features, and benefits of the proposed demand response management framework. Initially, we focus on the operation performance of our framework, in terms of the obtained optimal customers' consumption responses and companies' prices. Moreover, the distribution of the customers to the available companies in the Smart Grid Network is studied, and the corresponding power com-

panies' profit values are presented. Furthermore, the operation and the convergence of the distributed learning algorithm (i.e., stochastic learning automata) is illustrated, while the Demand and Response Management optimization problem to its stable solution, is presented as well. In addition, a detailed comparative evaluation of our approach against other alternative approaches is provided, and the differences with respect to the achieved customers' and companies' satisfaction, and customers' electricity energy consumption, are discussed.

On our base experimental scenario, we considered a Smart Grid Network consisting of  $J = 5$  power companies, and  $I = 100$  customers. Also, we considered  $k = 1000$ ,  $\lambda = 100$  and as a learning step  $b = 0.6$ , while the  $\gamma_i^{(t)}$  parameter of each customer is randomly generated. Each company has constant characteristics throughout the day (i.e, generation cost  $c_j^{(t)}$ , and discount policy  $f_j^{(t)}$ ), while the corresponding values that were used are:

1.  $\mathbf{f} = \{.0285, .027, .029, 0.3, .028\}$
2.  $\mathbf{c} = \{.255, .245, .265, .285, .265\}$

### Pure Operation of the PC-DRM Algorithm

At first, we focus on the power companies' selection process, via adopting the proposed distributed learning framework (i.e., Section 5.1.4). Each power company  $j, j \in \mathbb{J}$  aims to improve its market profile by achieving a low peak to average ratio (i.e.,  $PAR_j$ ) and high competitiveness (i.e.,  $Comp_j$ ). As we mentioned before, the low  $PAR_j$  factor indicates that the power company  $j$  balances the customers' electricity consumption over time via avoiding great consumption peaks, which may not be able to support. On the other hand, the high competitiveness factor  $Comp_j$  of the power company expresses the company's penetration in the market, in terms of the customers' portion that it serves.

Specifically, Figure 5.4a and Figure 5.4b present each power company's  $j, j \in \mathbb{J}$

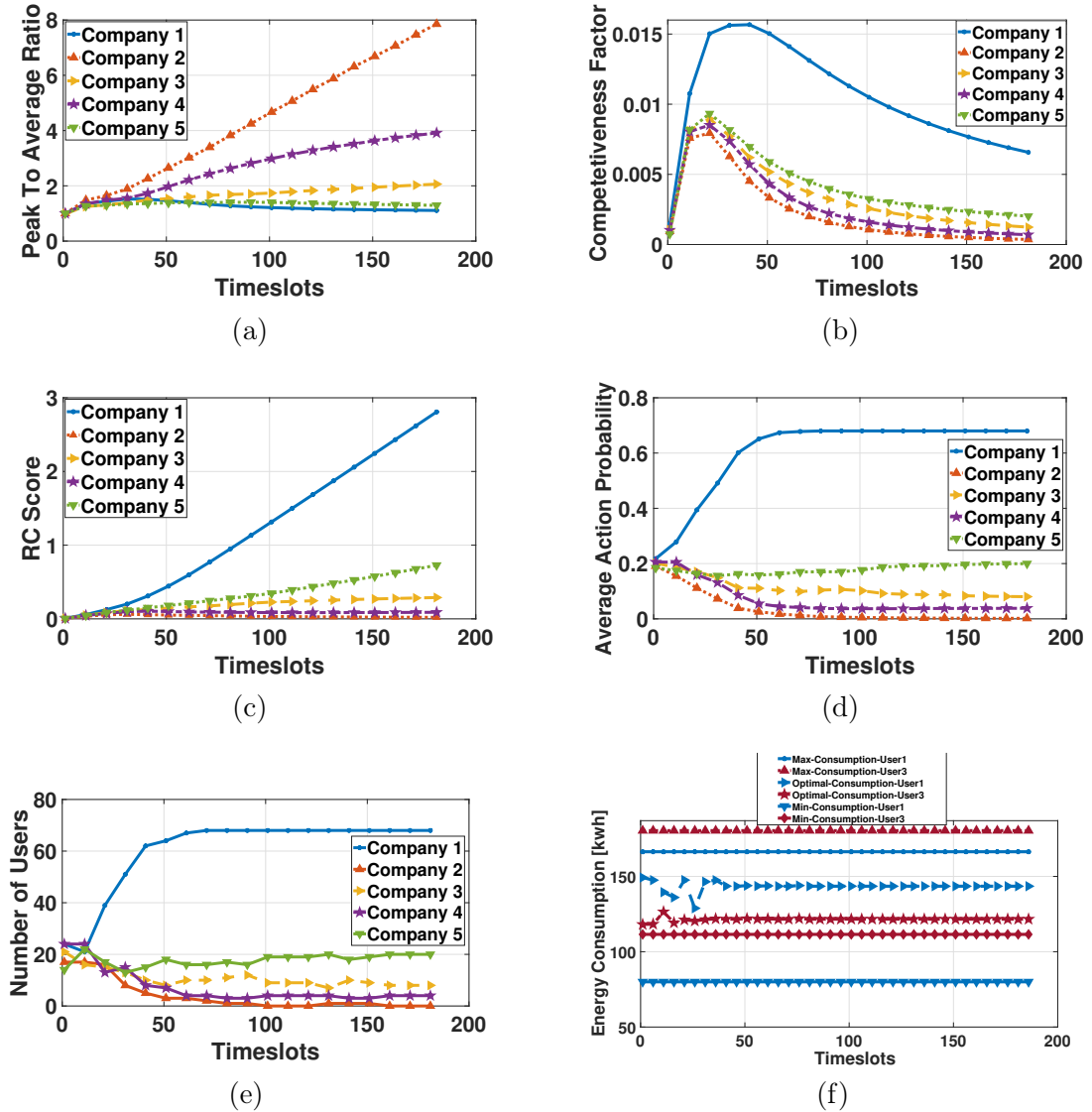


Figure 5.4: Pure operation of PC-DRM

peak to average ratio  $PAR_j$  and competitiveness factor  $Comp_j$  as a function of the PC-DRM algorithm's timeslots until the convergence of the distributed learning mechanism. Based on the considered configuration of this experimental setup, it is observed that the power companies 1 and 5 maintain the lowest  $PAR$  and competitiveness factor  $Comp$ . Moreover, it is noted that both the  $PAR$  and  $Comp$  factors



are determined by the solution of the DRM optimization problem, via the customers' optimal consumption to which the two-stage game-theoretic part converges at each timeslot.

Furthermore, Figure 5.4c depicts each company's  $j$  reputation and competitiveness factor  $RC_j$ , (i.e., Equation 5.4) as a function of the PC-DRM algorithm's timeslots. The results illustrate that the power companies 1 and 5 build a higher reputation and competitiveness factor  $RC$  in the market, compared to the rest, since they both achieve a lower  $PAR$  (i.e., Figure 5.4a) and a higher competitiveness factor (i.e., Figure 5.4b). Consequently, these two power companies create a better profile in the market, and the customers by learning and adapting their selection via the stochastic learning automata methodology, they have a higher average selection probability for these two companies (i.e., Figure 5.4d, and these two companies attract a higher portion of customers (i.e., Figure 5.4e over the timeslots. Specifically, as Figure 5.4e demonstrates, these two companies serve almost 90% of the market's customers, with company 1 absorbing almost 70%, as it achieves the highest reputation and competitiveness score, (i.e., Figure 5.4c), while company 5 with the second-best profile in the market serves approximately 20% of the market's customers.

Considering the DRM optimization problem, which in this work is studied via adopting a two-stage non-cooperative game-theoretic solution, Figure 5.4f presents two indicative customers' optimal energy consumption as a function PC-DRM algorithm's timeslots until its convergence to the stable customers' association to the power companies. As the customers converge to their stable power company selection, while at the same time they determine their optimal electricity consumption (i.e.,  $e_{i,j}^{(t)*}$  - Equation 5.11) for each timeslot by converging to the NE point of the non-cooperative game, their optimal electricity consumption converges to feasible values, while fulfilling their non-shiftable electricity needs (i.e., Min-consumption curves), as a higher priority is given to them, while at the same time the customers do not over-consume electricity, thus  $e_{i,j}^{(t)*} \leq d_i^{(t)}, \forall t \in \mathbb{T}$ . Moreover, as Figure 5.4f

illustrates, both of the presented customers consume a higher level of electricity than their non-shiftable demands, which confirms that the proposed framework achieves satisfies both the non-shiftable and a portion of the shiftable electricity needs of the market's customers.

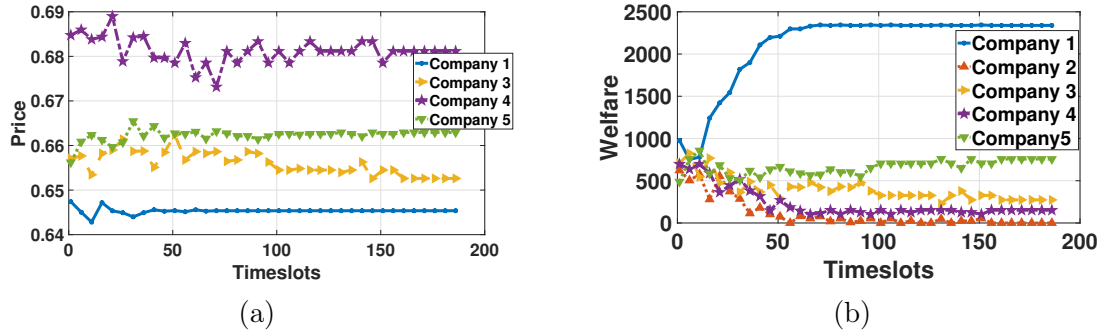


Figure 5.5: Companies optimal prices and perceived profits

Additionally, Figure 5.5a and Figure 5.5b depict the power companies' prices and profit values' convergence, as a function of PC-DRM algorithms' timeslots, respectively. It is noted that Figure 5.5a refers to indicative prices units per unit of electricity consumption (e.g.,  $\frac{\$}{KWh}$ ). Based on Figure 5.4f, 5.5a, and 5.5b, it is concluded that the DRM optimization problem converges to its final NE point, as the association between customers and power companies converges to its stable case, where both the customers and the power companies maximize their utilities and profits, leading them to low feasible low energy consumption and pricing policies, respectively. Moreover, it is worth to be noted, that company 5, which absorbs the second-highest portion of the markets' customers, is not the company with the second-lowest price in the market (i.e., company 3 has a lower price), which indicates that the announcement of a lower price by a company does not guarantee the absorbing of a higher portion of customers, as the customers select their power companies based on their market's profiles (i.e., reputation and competitiveness score), which depict the overall power companies' behavior in the market through the timeslots.

In addition, as Figure 5.5a illustrates, the companies 1 and 5 due to their higher reputation and competitiveness score (i.e., Figure 5.4c) converge to higher profit values compared to the rest of power companies, as this concludes to improved customers' preference to be served by these companies as we mentioned in Figure 5.4c. Finally, the power company 2 has not been selected by any customer (Figure 5.4c) in the scenario under consideration, and therefore its announced price is zero and it is not present in Figure 5.5a.

### Comparative Evaluation

In this section, we provide a detailed comparative evaluation study of the proposed PC-DRM framework against other approaches either from the recent literature or different implementation alternatives, highlighting the benefits of the PC-DRM algorithm in terms of customers' energy consumption and satisfaction. It is noted that for fairness and completeness purposes in the comparison, the power companies' profit values, as well as the convergence time of the different frameworks, are also evaluated and discussed.

Specifically, we evaluate the proposed PC-DRM framework against to five different approaches:

1. The demand response management algorithm (referred to as Evo) as proposed in [292], where the association of the customers to power companies is modeled as an evolutionary game and the customers form a population which is associated with only one company, as an outcome of the evolutionary game.
2. An alternative variation of the proposed PC-DRM algorithm-referred as MLdc, where the customers update their selection probabilities (i.e., Equation 5.8, 5.9) by using the reward probability  $r_j^{(t)} = \frac{f_j^{(t)}}{c_j^{(t)}}$ , in order to capture the profile of each power company  $j, j \in \mathbb{J}$ , in terms of its announced discounts and costs of the electricity generation. As a result, the customers select a power

company based only on monetary-related power companies' characteristics (i.e., discount  $f_j^{(t)}$ , and production cost  $c_j^{(t)}$ ), without considering the electricity-related characteristics of each power company, i.e., peak to average ratio  $PAR_j$  and competitiveness factor  $Comp_j$ . The DRM optimization problem is solved based on the DRM part of the PC-DRM algorithm.

3. A variation of the PC-DRM algorithm - referred as MLlp, by using as a reward probability the  $r_j^{(t)} = \frac{1}{\exp p_j^{(t)*}}$ , which is based on the power company's  $j, j \in \mathbb{J}$  optimal price  $p_j^{(t)*}$ . Specifically, the MLlp approach proposes to the customers as the best power company choice, the one with the lowest price.
4. The Random algorithm, where each customer is associated randomly to a power company, as each customer maintains an equal probability  $Pr_{i,j}^{(t)} = \frac{1}{J}$  of selecting each power company. The companies' prices and customers' consumed electricity are determined based on the DRM optimization part (i.e., the non-cooperative game) of the PC-DRM framework.
5. The best discount and cost - referred to as Bdc algorithm, which associates all the customers with the power company  $j, j \in \mathbb{J}$  that maintains the best  $\frac{f_j^{(t)}}{c_j^{(t)}}$  factor and the DRM optimization problem is also solved based on DRM part of the proposed PC-DRM framework.

Figures 5.6a, and 5.6b depict customers' perceived average utility and optimal energy consumption, respectively, as a function of the number of timeslots that all the comparative frameworks need to converge to stable customers' association to the power companies. As it is shown, the proposed PC-DRM algorithm achieves the highest customers' utilities (i.e., Figure 5.6a), and among the lowest customers' electricity consumption (i.e., Figure 5.6b). This trend stems from the holistic consideration of the power companies' characteristics, i.e., both the monetary and the electricity-related characteristics, as these are captured by the reputation and competitiveness factor Equation 5.4. Moreover, the MLdc variation of the PC-DRM

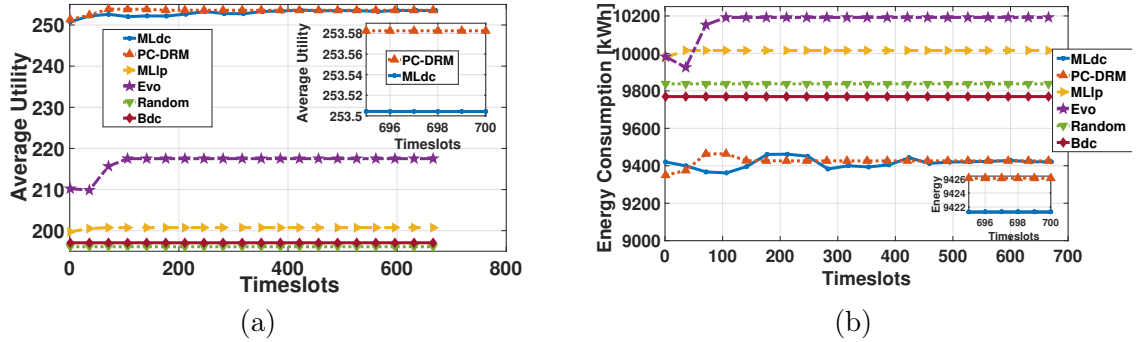


Figure 5.6: Customers' utility and overall consumed energy

algorithm, which considers only the power companies' monetary-related characteristics to perform the customers' association with the power companies, achieves similar customers' utilities and electricity consumption, showing the significance of the monetary factors. This happens mainly because the monetary factors contribute in the optimal power companies' pricing policy (Equation 5.10) and customers' consumption response (Equation 5.11), thus they affect the power companies' electricity-related factors (i.e., PAR, Comp). As a result, our proposed framework consists of a more general and holistic approach compared to MLdc, by avoiding possible high peaks of consumption in the case where power companies aim to attract the customers by high discounts in short-term periods, thus the sufficient satisfaction of the customers' is guaranteed by the PC-DRM approach in a long-term period.

On the other hand, the approaches that do not provide the opportunity to the customers to learn from their past decisions (i.e., Bdc, Random approaches) achieve the lowest customers' utility and high electricity consumption. Specifically, each customer selects its power company at a single time, by not exploring the Smart Grid Network environment for better choices. Furthermore, the Evo [292] algorithm, which is based on the outcome of an evolutionary game-theoretic approach, associates all the customers to only one power company, leading in that way into a monopoly scenario where the customers achieve significantly lower average utility, while at the

same time their electricity consumption increases. Finally, the MLlp algorithm leads the customers to select the power company with the lowest price, thus they tend to consume more electricity, which creates a domino effect, as the customers' cost increases, and their perceived average utility decreases.

Scenarios	Profit					Avg. Profit	T(sec)
	C1	C2	C3	C4	C5		
PC-DRM	2338	0	273	151	756	704	1.01
MLdc	951	837	1004	349	374	703	4.86
MLlp	840	1323	784	457	974	876	0.43
Evo	0	4392	0	0	0	878	0.65
Random	479	1380	759	793	848	852	0.04
Bdc	4208	0	0	0	0	842	0.05

Figure 5.7: Power companies' welfare and algorithms' convergence time

Figure 5.7 includes comparatively, the achieved power companies' profit values, the average profit, and the actual convergence time (in seconds) for all the comparative approaches. As it is shown, the PC-DRM and MLdc algorithms present similar companies' average profit values, while the PC-DRM proposed framework presents significantly lower complexity by achieving almost a five-fold reduction in convergence time. This is observed, since, in an open electricity market, where the power companies have similar monetary-related characteristics (i.e., production cost, discounts), the customers may flip among the companies, thus the MLdc approach has a delayed convergence. The more holistic approach of the PC-DRM algorithm, where electricity-related characteristics are also considered, contributes to customers' faster decision-making in selecting the most appropriate company for receiving service from. The Random and Bdc algorithms, which allow the customers to make a single time power company selection have the lowest convergence time compared to all the other approaches. As we already mentioned, the last two algorithms present quite poor performance in terms of customers' utility and electricity consumption. Finally, both Evo and MLlp approaches to achieve similar results in terms of com-

panies' profit values and convergence time.

### 5.1.8 Summary

The joint problem of power company selection and demand response management in a competitive open electricity market of a Smart Grid Network, consisting of multiple power companies and multiple customers is studied. Initially, a low complexity distributed learning approach is proposed, where the customers acting as learning automata explore the environment (i.e., market) and select a power company to be served from autonomously. Then, the demand response management problem - DRM, is formulated as a two-stage non-cooperative, where at the first stage the customers determine their optimal electricity consumption that maximizes their perceived utility and a stable point (i.e., Nash Equilibrium) is achieved, while at the second stage each power company determines its optimal pricing policy that maximizes its profit. Moreover, a distributed iterative and low complexity algorithm is introduced to jointly implement the power company selection and the demand response management processes.

A detailed performance evaluation of the proposed approach was conducted via modeling and simulation, and the presented results confirmed the superiority of the proposed PC-DRM framework, in terms of the achieved customers' and companies' satisfaction, customers' energy consumption, and implementation complexity.

## 5.2 A Risk-aware Social Cloud Computing Model

A flexible resource sharing paradigm is introduced, to enable the allocation of users' computing tasks in a social cloud computing system offering both *Virtual Machines*(VMs) and *Serverless Computing*(SC) functions. VMs are treated as a safe computing resource, while SC due to the uncertainty introduced by its shared nature, is treated as a common pool resource, being susceptible to potential over-exploitation. These computing options are differentiated based on the potential satisfaction perceived by the user, as well as their corresponding pricing while taking into account the social interactions among the users. Considering the inherent uncertainty of the considered computing environment, Prospect Theory and the theory of the Tragedy of the Commons are adopted to properly reflect the users' behavioral characteristics, i.e., gain-seeking or loss-averse behavior, as well as to formulate appropriate prospect-theoretic utility functions, embodying the social-aware and risk-aware user's perceived satisfaction. A distributed maximization problem of each user's expected prospect-theoretic utility is formulated as a non-cooperative game among the users and the corresponding Pure Nash Equilibrium (PNE), i.e., optimal computing jobs offloading to the VMs and the SC, is determined, while a distributed low-complexity algorithm that converges to the PNE is introduced. The performance and key principles of the proposed framework are demonstrated through modeling and simulation.

### 5.2.1 Related Work

The remarkable growth of social networks over the last decade - evidenced by the more than 2.8 billion Facebook users and the 187 million Twitter users in 2021 - has concluded to new solutions for communication networks and mobile computing. It is predicted that by 2021, more than 67% of the overall enterprise information technology infrastructure and software development will be served by cloud-based offerings [303].



The *Social Cloud Computing* is arising as a resource sharing framework, which exploits the users' social ties to improve the services offered by the cloud providers. In [304], the trust levels among users in social networks are exploited to create a dynamic social cloud computing environment, where the users are sharing their cloud computing resources, creating a volunteer social cloud computing environment. In [305], the authors tackle the problem of placing the users' computing tasks over multiple clouds considering the social-aware services and the users' social ties. While social cloud computing is still in its infancy, a new cloud computing solution is coined by the industry, named *Serverless Computing (SC)*, where the users' computing tasks are defined as a workflow of event-triggered functions [306]. In contrast to the model of *Virtual Machines (VMs)*, where the users are renting the VMs from the cloud provider and the resources could remain idle in the case of sporadic requested computing tasks, concluding to unwanted monetary cost, the SC model allows the users to offload computing tasks to the cloud provider, who remains responsible to manage the infrastructure and respective resources [307]. The users run stateless functions at the cloud providers' servers and are charged concerning the allocated memory and the actual required CPU time of executing them [308], thus promising a more cost-efficient and flexible model compared to the VMs. Example platforms supporting the SC model include: AWS Lambda [309], Google Cloud Functions [310], etc.

**Motivation:** However, all these efforts in social cloud computing and serverless computing have been progressing in isolation from each other. Thus, despite the advances that have been achieved in both these areas independently, the lack of joint consideration and exploitation of the users' social relation and the available VMs and SC by the cloud provider, limits their potential exploitation and adoption in a realistic scenario.

### 5.2.2 Contributions

We aim to fill the aforementioned research gaps by introducing a risk-aware social computing framework, which exploits the computing capabilities of the available VMs and SC offered by the cloud provider while accounting for the users' social ties and their risk-aware behavior. The latter stems from the risk imposed by the shared nature of the SC, which may become non-responsive due to its over-exploitation.

1. Each user dynamically offloads part of its computing jobs to the VMs and/or the SC. Fixed price is assumed for the use of VMs, while a social-aware SC pricing is considered based on the "social importance" of a user within the system. A holistic user's actual utility function is introduced to capture the user's satisfaction by executing its jobs in a specific time frame while considering the corresponding price.
2. Each VM is characterized as a "safe resource", as it is exclusively rent by a user, and accordingly, the user enjoys guaranteed computing service. In contrast, the SC is typically more cost-efficient, having the potential of providing high satisfaction to the user. Given that its computing resources are shared among many users, it is characterized as Common Pool Resource (CRP) - Chapter 2.3, which introduces risk in users' decisions to offload their computing jobs to it, as it can potentially become over-exploited. To this end, we capture this phenomenon via adopting the theory of the Tragedy of the Commons (Chapter 2.3). The problem of users' risk-aware offloading decision to the VMs and the SC, is formulated by using our introduced risk-aware decision-making framework (Chapter 2.3.2). Each user's prospect-theoretic utility function is introduced by considering its actual utility, its behavioral patterns, and the probability of the SC's failure (i.e., non-responsiveness).
3. The problem of each user determining the number of computing jobs that will be offloaded at the SC and the VMs, is formulated as a maximization problem of its expected prospect-theoretic utility, and treated as a non-cooperative game among the users. The existence and uniqueness of a Pure Nash Equilibrium (PNE) are shown,

while a distributed and low-complexity algorithm is introduced, and its convergence to the unique PNE is proven.

4. A series of simulation experiments is performed to evaluate the performance of the proposed risk-aware social computing framework. A comparative study demonstrates its superiority, in terms of user's satisfaction and proper system operation.

### 5.2.3 System Model

A Cloud Provider (CP) consisting of Virtual Machines (VMs) and Serverless Computing (SC) functions is considered. A set of  $\mathbb{N} = \{1, \dots, N\}$  users is assumed, while a set of  $\mathbb{T} = \{1, \dots, T\}$  time slots is defined, where  $D_t$  denotes the duration of each time slot  $t$ . Each user  $i$  has a number  $J_i^{(t)}$  of computing jobs that want to offload to the CP for remote execution per time slot. In the VMs case, a user can reserve a VM with its operating system and predefined on-demand computational and storage capabilities, while in the SC the user executes its serverless instance as an application in a common operating system without any control over the resources on which the job is executed. Given a specific type of VMs, we define as  $\lambda_{vm}^{(t)}(D_t)$  the maximum number of jobs that can be executed by the VM in the duration  $D_t$ , where  $\lambda_{vm}^{(t)}(D_t)$  is an increasing function of the duration  $D_t$ . Each user  $i$  aims at determining the number  $\lambda_i^{(t)}$  ( $\lambda_i^{(t)} \leq J_i^{(t)}$ ) of computing jobs to be executed at the SC, while the rest ( $J_i^{(t)} - \lambda_i^{(t)}$ ) jobs are offloaded to the VMs.

With respect to the social aspects of the cloud computing system, we define an overlay virtual representation of the system as follows:  $\mathbb{S} = \{\mathbb{N}, \mathbb{E}, \mathbb{W}\}$ , where the users  $\mathbb{N} = \{1, \dots, N\}$  may interact with each other. Specifically, the edge set, i.e., interactions, is denoted as  $\mathbb{E} = \{(i, j) : e_{i,j} = 1, \forall i, j \in \mathbb{N}\}$ , where  $e_{i,j} = 1$  indicates the existence of information flow from user  $i$  to user  $j$ . The weight set  $\mathbb{W} = \{w_{i,j}, \forall i, j \in \mathbb{N}\}$  is defined, where  $w_{i,j} \in \mathbb{R}$  depicts the strength of the interaction (e.g., criticality of information) that is exchanged between the source user  $i$  and the destination user  $j$ , while  $w_{i,j} = 0, \forall i, j \in \mathbb{N}$  such that  $e_{i,j} = 0$ . Therefore,

## Chapter 5. Smart Technologies

each user is characterized by its social factor  $f_i = \frac{\omega_1 \sum_{j \in \mathbb{N}, j \neq i} w_{i,j} + \omega_2 \sum_{j \in \mathbb{N}, j \neq i} w_{j,i}}{\sum_{j \in \mathbb{N}} f_j}$ , where  $\omega_1, \omega_2 \in [0, 1]$ ,  $\omega_1 + \omega_2 = 1$  depict the weights of a user's interactions by acting as a sender or receiver of information, respectively.

In the VMs case, each user is charged based on the reserved VMs:  $p_{i,vmp}^{(t)} = \frac{J_i^{(t)} - \lambda_i^{(t)}}{\lambda_{vm}^{(t)}(D_t)} \cdot p_{vm}$ , where  $p_{vm}$  is a fixed VM's price [308]. In the SC case, the user is charged based on its execution time:  $p_{i,sc}^{(t)} = \lambda_i^{(t)} D_t f_i^{-1} p_{sc}^{(t)}(\lambda_T^{(t)})$ , where the average SC's response time is  $D_t$  and  $f_i^{-1}$  shows that the more important is a user for the social cloud computing system the greater is the incentive for the SC to assign a lower price. The  $p_{sc}^{(t)}(\lambda_T^{(t)})$  is the SC's rate of return function, which is a function of the overall number of offloaded jobs at the SC, i.e.,  $\lambda_T^{(t)} = \sum_{i \in \mathbb{N}} \lambda_i^{(t)}$ , and is formulated as:

$$p_{sc}^{(t)}(\lambda_T^{(t)}) = \begin{cases} \frac{\Lambda_{sc}^{(t)} - \lambda_T^{(t)}}{\Lambda_{sc}^{(t)}} \cdot p_{sc} & , \text{ if } \lambda_T^{(t)} < \Lambda_{sc}^{(t)} \\ p_{sc}^f & , \text{ otherwise} \end{cases} \quad (5.22)$$

where  $\Lambda_{sc}^{(t)}$  is the number of jobs threshold that the SC can operationally process during  $D_t$ . If  $\lambda_T^{(t)} \geq \Lambda_{sc}^{(t)}$ , the SC's response time is greater than  $D_t$  and the SC "fails", thus, the SC's price is the minimum one ( $p_{sc}^f < p_{sc}$ ). To this end, the latter phenomenon is essentially described by the introduced concept of the Tragedy of the Commons 2.3. In the case of SC's failure, the user's successfully executed jobs during  $D_t$  are only the ones executed at the VMs.

**Proposition 12** *The SC's rate of return function  $p_{sc}^{(t)}(\lambda_T^{(t)})$  is strictly decreasing with respect to  $\lambda_T^{(t)}$ , since as the  $\lambda_T^{(t)}$  increases the less the SC can guarantee that the average response time is  $D_t$ , and the lower is the SC's price.*

**Actual Utility:** The user's  $i$  actual utility  $z_i^{(t)}$  expressing its satisfaction from executing  $\lambda_i^{(t)}$  jobs at the SC and the rest  $(J_i^{(t)} - \lambda_i^{(t)})$  at the VMs is formulated. This satisfaction is captured by the portion of jobs that are executed successfully during

the timeslot  $t$  and the user's overall cost, as follows.

$$z_i^{(t)}(\lambda_i^{(t)}, \lambda_{-i}^{(t)}) = \frac{E_i^{(t)}}{J_i^{(t)}} - \frac{p_{i,vmp}^{(t)} + p_{i,sc}^{(t)}}{B_i^{(t)}} \quad (5.23)$$

where  $\lambda_{-i}^{(t)}$  is the vector of the offloading decisions of all users except  $i$ ,  $E_i^{(t)}$  are the jobs that are executed successfully during  $D_t$ , and  $B_i^{(t)}$  is the user's  $i$  total budget.

### 5.2.4 The Prospect of Cloud

The SC is a CPR since all the users can arbitrarily offload part of their jobs to it and share its resources. Towards maximizing the actual utility, each user aims at determining in an autonomous and distributed manner the number of jobs  $\lambda_i^{(t)}$  offloaded to the SC, by accounting for the uncertainty of the SC's failure due to over-exploitation. Based on this uncertainty, we introduce the SC's probability of non-responsiveness.

**Assumption 2** *SC's probability of non-responsiveness  $PnR^{(t)}(\lambda_T^{(t)})$  is strictly increasing, convex and twice continuously differentiable with respect to  $\lambda_T^{(t)} \in [0, \Lambda_{sc}^{(t)}]$ , with  $PnR^{(t)}(\lambda_T^{(t)}) = 1, \forall \lambda_T^{(t)} \geq \Lambda_{sc}^{(t)}$ .*

A linear probability of non-responsiveness  $PnR^{(t)}(\lambda_T^{(t)}) = \max\{\frac{\lambda_T^{(t)}}{\Lambda_{sc}^{(t)}}, 1\}$  is considered. Other forms of  $PnR^{(t)}$  that follow Assumption 2 can be considered without damaging the applicability and validity of the following analysis.

### Risk-aware Resource Allocation under Prospect Theory

Our introduced risk-aware decision-making framework (Chapter 2.3.2) is adopted to address the users' subjectivity in decision-making under the SC's uncertainty. Following this behavioral decision-making framework, the users make actions under risk and uncertainty regarding the corresponding payoff of their actions. Each user's satisfaction by offloading a number of jobs to the SC and the VMs is evaluated with

respect to a reference point (reference dependence property). Each user's reference point is the guaranteed utility  $z_{i,0}^{(t)}$  that the user obtains by offloading all the jobs at the VMs (referred to as the safe resource), thus,  $\lambda_i^{(t)} = 0$ . Therefore, each user's  $i$  reference point is  $z_{i,0}^{(t)} = 1 - \left\lfloor \frac{J_i^{(t)}}{\lambda_{vm}^{(t)}(D_t)} \right\rfloor \cdot \frac{p_{vm}}{B_i^{(t)}}$ , where  $\frac{E_i^{(t)}}{J_i^{(t)}} = 1$  (Equation 5.23) since all the jobs are executed successfully during the time slot  $t$ .

To this end, as per Chapter 2.3.2, each user's  $i, i \in \mathbb{N}$  prospect-theoretic utility is defined as follows:

$$u_i^{(t)}(\lambda_i^{(t)}, \lambda_{-i}^{(t)}) = \begin{cases} (z_i^{(t)} - z_{i,0}^{(t)})^{\alpha_i} & , \text{ if } z_i^{(t)} \geq z_{i,0}^{(t)} \\ -k_i \cdot (z_{i,0}^{(t)} - z_i^{(t)})^{\beta_i} & , \text{ if } z_{i,0}^{(t)} > z_i^{(t)} \end{cases} \quad (5.24)$$

If the SC does not fail due to the overall offloaded number of jobs  $\lambda_T^{(t)}$ , then  $z_i^{(t)} \geq z_{i,0}^{(t)}$ , and by appropriate mathematical derivations based on the first branch of Equation 5.24, we conclude that its prospect-theoretic utility is

$$u_i^{(t)}(\lambda_i^{(t)}, \lambda_{-i}^{(t)}) = (\lambda_i^{(t)})^{a_i} \left( \frac{\gamma_i p_{vm}}{\lambda_{vm}^{(t)}(D_t) B_i^{(t)}} - \frac{D_t f_i^{-1} p_{sc}^{(t)}(\lambda_T^{(t)})}{B_i^{(t)}} \right)^{a_i} \quad (5.25)$$

, where  $\gamma_i$  is the user's  $i$  regulator factor, such that  $\frac{-\lambda_i^{(t)}}{\lambda_{vm}^{(t)}(D_t)} \cdot \gamma_i = \left\lfloor \frac{-\lambda_i^{(t)}}{\lambda_{vm}^{(t)}(D_t)} \right\rfloor$ . On the other hand, if the SC "fails", user's  $i$  experienced actual utility  $z_i^{(t)}$  is lower than its reference point  $z_{i,0}^{(t)}$ , and the user's  $i$  prospect-theoretic utility is obtained as:

$$u_i^{(t)}(\lambda_i^{(t)}, \lambda_{-i}^{(t)}) = -k_i (\lambda_i^{(t)})^{a_i} \left( \frac{1}{J_i^{(t)}} - \frac{\gamma_i p_{vm}}{\lambda_{vm}^{(t)}(D_t) B_i^{(t)}} + \frac{f_i^{-1} D_t}{B_i^{(t)}} p_{sc}^f \right)^{a_i} \quad (5.26)$$

where based on the second branch of Equation 5.24, the price of the SC is the minimum one, thus  $p_{sc}^{(t)} = p_{sc}^f$ . For notational convenience we define  $\epsilon_i^{(t)} = \left( \frac{1}{J_i^{(t)}} - \frac{\gamma_i p_{vm}}{\lambda_{vm}^{(t)}(D_t) B_i^{(t)}} + \frac{f_i^{-1} D_t}{B_i^{(t)}} p_{sc}^f \right)^{a_i}$ , and  $h_i^{(t)}(\lambda_T^{(t)}) = \left( \frac{\gamma_i p_{vm}}{\lambda_{vm}^{(t)}(D_t) B_i^{(t)}} - \frac{D_t f_i^{-1} p_{sc}^{(t)}(\lambda_T^{(t)})}{B_i^{(t)}} \right)^{a_i}$ , and if the SC does not fail, then  $h_i^{(t)}(\lambda_T^{(t)}) > 0$ . Thus, considering the probability of non-responsiveness  $PnR^{(t)}$  of the SC, the user's prospect-theoretic utility can be written as:

$$u_i^{(t)}(\lambda_i^{(t)}, \lambda_{-i}^{(t)}) = \begin{cases} (\lambda_i^{(t)})^{a_i} h_i^{(t)}(\lambda_T^{(t)}) & \text{with prob. } 1 - PnR^{(t)} \\ -k_i \epsilon_i^{(t)} (\lambda_i^{(t)})^{a_i} & \text{with prob. } PnR^{(t)} \end{cases} \quad (5.27)$$

, and as the user's  $i$  expected prospect-theoretic utility is given as (Chapter 2.3.2) :

$$\begin{aligned}\mathbb{E}(u_i^{(t)}) &= (\lambda_i^{(t)})^{a_i} h_i^{(t)} (1 - PnR^{(t)}) - (\lambda_i^{(t)})^{a_i} k_i \epsilon_i^{(t)} PnR^{(t)} \\ &= (\lambda_i^{(t)})^{a_i} [h_i^{(t)} (1 - PnR^{(t)}) - k_i \epsilon_i^{(t)} PnR^{(t)}] \\ &= (\lambda_i^{(t)})^{a_i} g_i(\lambda_T^{(t)})\end{aligned}\tag{5.28}$$

where  $g_i(\lambda_T^{(t)}) = [h_i^{(t)} (1 - PnR^{(t)}) - k_i \epsilon_i^{(t)} PnR^{(t)}]$  is the user's effective rate of return from the SC.

### 5.2.5 Optimizing Resource Allocation: Problem Formulation

Each user's  $i$  goal is to maximize its perceived expected prospect-theoretic utility (Equation 5.28) via determining its best recourse allocation strategy, i.e., the number of jobs  $\lambda_i^{(t)}$  that are offloaded at the SC at timeslot  $t$ . This problem is formulated as a maximization problem of each user's  $i$  expected prospect-theoretic utility function (Equation 5.28), as follows:

$$\max_{\lambda_i^{(t)} \in S_i^{(t)}} \mathbb{E}(u_i^{(t)}) = (\lambda_i^{(t)})^{a_i} g_i(\lambda_T^{(t)})\tag{5.29}$$

where  $S_i^{(t)}$  is the user's  $i$  strategy space as it is defined later.

The above maximization problem can be treated as a non-cooperative game  $\mathbb{G} = \{\mathbb{N}, \{S_i^{(t)}\}, \{\mathbb{E}(u_i^{(t)})\}\}$  among the  $N$  users, where  $S_i^{(t)} = [0, \min(J_i^{(t)}, \Lambda_{sc}^{(t)})]$  is the strategy space of each user  $i$ , and  $\mathbb{E}(u_i^{(t)})$  is its expected prospect-theoretic utility. Towards solving the non-cooperative game, the concept of Pure Nash Equilibrium (PNE) - Chapter 2.1 is adopted. Let  $\lambda^{*,(t)} = [\lambda_1^{*,(t)}, \dots, \lambda_N^{*,(t)}]$  denote the users' resource allocation strategies and  $\lambda_{-i}^{*,(t)}$  the vector of all the users' resource allocation strategies except user  $i$  at the PNE point.

### Existence and Uniqueness of PNE

The best response strategy of user  $i$  is

$$B_i(\lambda_{-i}^{(t)}) = \arg \max_{\lambda_i^{(t)} \in S_i^{(t)}} \mathbb{E}(u_i^{(t)}(\lambda_i^{(t)}, \lambda_{-i}^{(t)})) : S_{-i}^{(t)} \rightrightarrows S_i^{(t)}\tag{5.30}$$

where  $S_{-i}^{(t)} = \times_{j \in \mathbb{N} - \{i\}} S_j^{(t)}$

**Theorem 16** *For each user  $i$ , its best response strategy exists and it is single-valued, such that  $\lambda_i^{*,(t)} = B_i(\lambda_{-i}^{(t)})$ .*

We adopt the notation  $\lambda_{-i,T}^{(t)} = \sum_{j \in \mathbb{N}, j \neq i} \lambda_j^{(t)}$  to depict the total number of offloaded jobs at the SC of all users except user  $i$ . The proof of Theorem 6 can be readily concluded based on Berge's Theorem [226] and the following Lemmas 9-11.

**Lemma 9** *For each user  $i$  the following holds true: i) there exists a value  $\bar{\lambda}_i^{(t)}$ , such that  $g_i(\bar{\lambda}_i^{(t)}) = 0$ , ii) if  $\lambda_{-i,T}^{(t)} \geq \bar{\lambda}_i^{(t)}$  then  $\lambda_i^{*,(t)} = 0$ , and iii) if  $\lambda_{-i,T}^{(t)} < \bar{\lambda}_i^{(t)}$  there exists an user-specific interval  $A_i^{(t)} \subset [0, \bar{\lambda}_i^{(t)})$  such that all user's best responses are positive, and  $\lambda_i^{*,(t)} + \lambda_{-i,T}^{(t)} \in A_i^{(t)}$ .*

The proof of Lemma 9 is directly emanated from the proof of Lemma 1 .

**Lemma 10** *The best response  $\lambda_i^*, \forall i \in \mathbb{N}$  is single-valued  $\forall \lambda_{-i,T} \in [0, \Lambda_{sc}]$ .*

*Proof:* Based on Lemma 9 we know that  $\forall \lambda_i > 0$  such that  $\lambda_i + \lambda_{-i,T} \in A_i$ , we have  $g_i(\lambda_T) > 0$  and  $\frac{\partial g_i(\lambda_T)}{\partial \lambda_T} < 0$ , where  $\lambda_T = \lambda_i + \lambda_{-i,T}$ . Also, since  $g_i(\lambda_T)$  is concave in interval  $A_i$  (Lemma 9), the user's  $i$  expected prospect-theoretic utility is concave, i.e.,  $\frac{\partial^2 \mathbb{E}(u_i)}{\partial \lambda_i^2} = \lambda_i^{a_i} \frac{\partial^2 g_i(\lambda_T)}{\partial \lambda_T^2} + 2a_i \lambda_i^{a_i-1} + a_i(a_i-1) \lambda_i^{a_i-2} g_i(\lambda_T) < 0$ . As a result, since any best response  $\lambda_i^*$  satisfies  $\lambda_i^* + \lambda_{-i,T} \in A_i$ ,  $\lambda_i^*$  is an argument of maximum of  $\mathbb{E}(u_i)$ , and therefore is unique.

**Lemma 11** *The user's best response  $\lambda_i^* : S_{-i} \rightrightarrows S_i$  is continuous for  $\lambda_{-i} \in S_{-i}$ .*

The proof of Lemma 11 is derived based on Berge's Theorem [226] and Lemma 10, and is directly emanated from the proof of Lemma 3.

**Theorem 17** *A Pure Nash Equilibrium  $\lambda^* = [\lambda_1^*, \dots, \lambda_N^*]$  of the non-cooperative game  $\mathbb{G} = [\mathbb{N}, \{S_i\}, \{\mathbb{E}(u_i)\}]$  exists.*



*Proof:* The strategy set  $S_i$  is a convex compact subset of the Euclidean space and so is the joint strategy space,  $S = S_1 \times \cdots \times S_N \subset \mathbb{R}^N$ . By defining a mapping  $T : S \rightarrow S$  such that  $T(\lambda_1, \dots, \lambda_N) = (\lambda_1^*, \dots, \lambda_N^*)$ , from Lemma 10,  $T$  is single-valued and from Lemma 11 is continuous. Brouwer's fixed point theorem guarantees the existence of a strategy profile  $s = \{\lambda_i^*\}_{i \in \mathbb{N}} \in S$  that is invariant under the best response mapping and therefore is a PNE of  $\mathbb{G}$  [226].

**Lemma 12** *The function  $d_i(\lambda_T) = \frac{-a_i g_i(\lambda_T)}{\frac{\partial g_i(\lambda_T)}{\partial \lambda_T}}$  is strictly decreasing with respect to  $\lambda_T$ ,  $\forall \lambda_T \in A_i$ .*

*Proof:* The first-order derivative of  $d_i(\lambda_T)$  is  $\frac{\partial d_i(\lambda_T)}{\partial \lambda_T} = -a_i \frac{(\frac{\partial g_i(\lambda_T)}{\partial \lambda_T})^2 - g_i(\lambda_T) \frac{\partial^2 g_i(\lambda_T)}{\partial \lambda_T^2}}{(\frac{\partial g_i(\lambda_T)}{\partial \lambda_T})^2}$ . When  $\lambda_T \in A_i$ , based on Lemma 9 it holds true that  $g_i(\lambda_T) > 0$  and  $\frac{\partial^2 g_i(\lambda_T)}{\partial \lambda_T^2} \leq 0$ , therefore it follows directly that  $\frac{\partial d_i(\lambda_T)}{\partial \lambda_T} < 0$ ,  $\forall \lambda_T \in A_i$ .

**Theorem 18** *The Pure Nash Equilibrium of the non-cooperative game  $\mathbb{G}$  is unique.*

*Proof:* We use the notation  $\lambda_T^*$  to denote the total offloaded number of jobs at the SC at the PNE of game  $\mathbb{G}$ . The proof of Theorem 18 is based on the reduction to absurdity. Let  $\lambda_{T(1)}^*, \lambda_{T(2)}^*$  be two distinct PNE points. Without loss of generality we assume that  $\lambda_{T(2)}^* > \lambda_{T(1)}^*$ . We define the set  $Sup \triangleq \{i \in \mathbb{N} : \lambda_T^* < \bar{\lambda}_i\}$ , thus it includes every user that offloads a non-zero number of jobs at the SC. Thus,  $Sup_2 \subset Sup_1$ . Also, we have  $\sum_{j \in Sup_1} d_j(\lambda_{T(1)}^*) = \lambda_{T(1)}^*$ ,  $\sum_{j \in Sup_2} d_j(\lambda_{T(2)}^*) = \lambda_{T(2)}^*$ . So,  $\sum_{j \in Sup_2} d_j(\lambda_{T(1)}^*) + \sum_{j \in Sup_1 \setminus Sup_2} d_j(\lambda_{T(1)}^*) = \lambda_{T(1)}^* \Rightarrow \sum_{j \in Sup_2} d_j(\lambda_{T(1)}^*) \leq \lambda_{T(1)}^* < \lambda_{T(2)}^* = \sum_{j \in Sup_2} d_j(\lambda_{T(2)}^*)$ . However,  $d_j(\lambda_T)$  is decreasing, so  $d_j(\lambda_{T(1)}^*) > d_j(\lambda_{T(2)}^*)$ ,  $\forall j \in Sup_2$ , which is contradiction. Thus,  $\lambda_{T(1)}^* = \lambda_{T(2)}^*$ .

### 5.2.6 Algorithm-Convergence to PNE

Based on Lemma 12, each user's best response strategy  $\lambda_i^*$  is decreasing with respect to the total number of offloaded jobs  $\lambda_{-i,T}$  of the rest users. Thus,  $\mathbb{G}$  belongs to the

best-response potential games, and therefore the sequential best response dynamics converge to the PNE [228]. Each user  $i$  first receives the total number of offloaded jobs of the rest users, i.e.,  $\lambda_{-i,T}$ , in order to compute its best response  $\lambda_i^*$  and it determines if  $\lambda_i^* = 0$ , thus, whether  $g_i(\lambda_{-i,T}) \leq 0$  and  $\frac{\partial g_i(\lambda_T)}{\partial \lambda_T}|_{\lambda_T=\lambda_{-i,T}} < 0$  holds true (conditions stemming from Lemma 9). If the user  $i$  finds that  $\lambda_{-i,T} < \bar{\lambda}_i$ , then its  $\lambda_i^*$  exists and is single-valued (Theorem 16). Specifically, due to the existence of the unique root of  $\frac{\partial \mathbb{E}(u_i)}{\partial \lambda_i} = 0$ , and considering that  $\frac{\partial \mathbb{E}(u_i)}{\partial \lambda_i}$  is a continuously differentiable and decreasing (i.e.,  $\frac{\partial^2 \mathbb{E}(u_i)}{\partial \lambda_i^2} < 0$ , Lemma 10) with respect to  $\lambda_i$ , then the unique root  $r_i^*$  can be found via binary search into  $[0, \Lambda_{sc}]$  with an approximation  $\epsilon \rightarrow 0$ , and finally user's  $i$  best response to be  $\lambda_i^* = \min(J_i, r_i^*)$ . The complexity of the binary search is  $\mathcal{O}(\log_2 \Lambda_{sc})$ . In each iteration of the sequential best response dynamics, only one user  $i$  determines its best response strategy via executing arithmetical calculations (Algorithm 7). By denoting as *Ite* the number of iterations that are needed for convergence to the PNE, the complexity of the Algorithm 7 is  $\mathcal{O}(N * \text{Ite} * \log_2 \Lambda_{sc})$ . It is noted that the execution time of Algorithm 1 scales very well with respect to the number of users, as it is presented in the Empirical Evaluation.

### 5.2.7 Empirical Evaluation

In this section, we provide detailed numerical results to illustrate the performance of the proposed approach in terms of the following aspects: basic operation of our framework, scalability, and framework's behavior under heterogeneous users in terms of loss aversion parameter  $k_i$ . Finally, a comparative evaluation of our approach against alternative resource allocation techniques is provided.

In our study, the duration of each timeslot is  $D_t = 1\text{sec}$  and the price of reserving a VM for  $D_t$  is  $p_{vm} = 10$ , while the SC's price per unit of time is  $p_{sc} = 0.3$  and  $p_{sc}^f = 0.2$  [308]. The maximum number of jobs that can be executed by an VM instance during  $D_t$  is  $\lambda_{vm}^{(t)} = 10$ . A directed social network is created with random topology and 100 users, where each user has  $J_i^{(t)} \in [400, 1000]$  number of jobs. Each

---

**Algorithm 7** Distributed Algorithm for Convergence to PNE

---

```

1: Input/Initialization:  $\mathbb{S}, D_t, f_i, p_{sc}, p_{sc}^f, \Lambda_{sc}, p_{vm}, \lambda_{vm}$ 
    $Ite = 0, \lambda_i \in [0, \min(J_i, \Lambda_{sc})], \forall i \in \mathbb{N}$ 
2: Output: PNE profile  $\lambda^* = [\lambda_1^*, \dots, \lambda_N^*]$ 
3: while PNE not reached do
4:    $Ite = Itc + 1$ 
5:   for  $i = 1$  to  $N$  do
6:     User  $i$  receives the  $\lambda_{-i,T}$ 
7:     if  $(g_i(\lambda_{-i,T}) \leq 0 \ \&\& \ \frac{\partial g_i(\lambda_T)}{\partial \lambda_T}|_{\lambda_T=\lambda_{-i,T}} < 0)$  then
8:        $\lambda_i^* = 0$ 
9:     else
10:       $r_i^* = BinarySearch([0, \Lambda_{sc}], \epsilon), \epsilon \rightarrow 0$ 
11:       $\lambda_i^* = \min(J_i, r_i^*)$ 
12:    end if
13:  end for
14:  Check convergence to PNE
15: end while

```

---

user is associated with its social factor  $f_i$ . For the SC, we have  $\Lambda_{sc}^{(t)} = 10\% \times \sum_{i \in \mathbb{N}} J_i^{(t)}$ . Unless otherwise stated, we assume homogeneous users with parameters  $a_i = 0.2$ ,  $k_i = 5$ .

### Pure Operation of the Proposed Framework

Figure 5.8a illustrates the average number of offloaded jobs to the SC (left vertical axis) and the average expected prospect theoretic utility (right vertical axis expressed in logarithmic scale), as a function of the iterations (low horizontal axis) and the execution time (upper horizontal axis). Figure 5.8b presents the overall number of jobs at the SC (left vertical axis) and the SC's probability of non-responsiveness

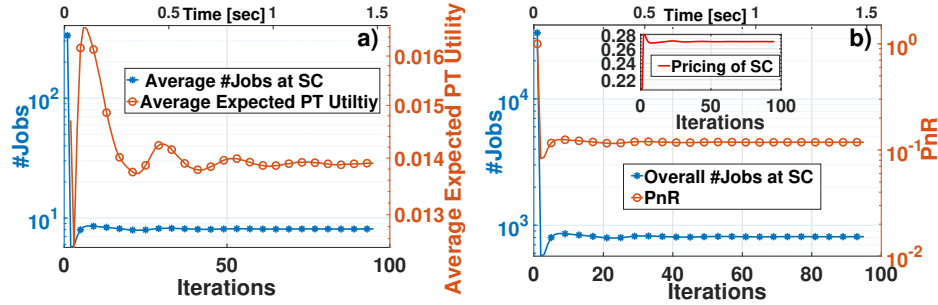


Figure 5.8: Pure operation

(right vertical axis), while in the contained sub-figure the corresponding SC's pricing is depicted. From the results in Figure 1a and Fig 1b, we confirm that starting from a random initial strategy, as the time evolves the algorithm converges to a stable point (i.e. unique PNE point), where each user has determined its best response strategy. Throughout this evolving process and till we reach the PNE point, the users either offload a larger number of jobs at the SC in order to increase their expected prospect theoretic utility, or they follow an opposite resource allocation strategy, i.e., a lower number of jobs at the SC, when the SC's probability of non-responsiveness increases.

### Scalability Evaluation

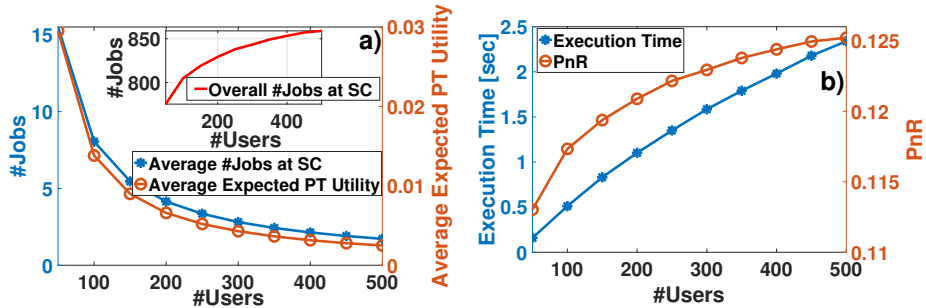


Figure 5.9: Scalability Evaluation - Increased number of users

Figure 5.9 illustrates each user's average number of offloaded jobs at the SC (and

the sub-figure presents the total number of jobs at the SC) and the average expected prospect theoretic utility with respect to the number of users. As the number of users increases, the SC becomes more congested (increased total number of offloaded jobs at the SC - contained sub-figure in Figure 5.9a), while each user offloads a smaller number of jobs, since its incentive is reduced due to the higher SC's probability of non-responsiveness (Figure 5.9b), while experiencing a lower expected prospect theoretic utility (Figure 5.9a). Figure 5.9b shows the actual required time for our algorithm to converge to the PNE. As observed from the results our algorithm's execution time presents sub-linear behavior with respect to the number of users and is well aligned with our scalability analysis.

### Heterogeneous Users - Loss Aversion

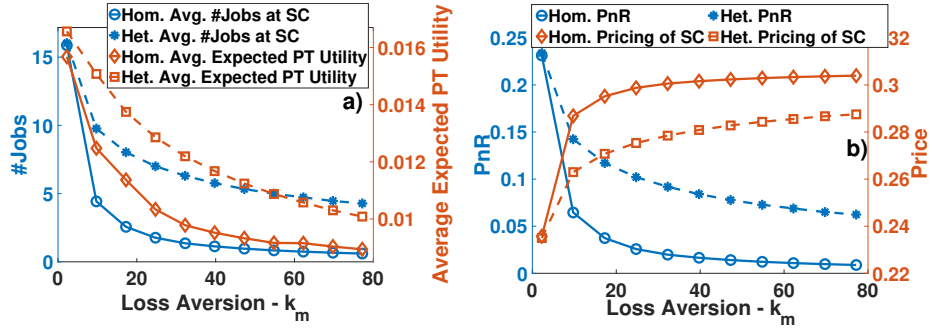


Figure 5.10: Loss Aversion impact study

The impact of users' heterogeneous loss aversion prospect theoretic behavior on the achievable performance is studied. In particular, in Figure 5.10a and Figure 5.10b we compare a scenario of heterogeneous users, where each user is associated with a different personalized loss aversion index  $k_i$ , against a homogeneous scenario where all users assume the same exactly loss aversion parameter  $k_m$ . For fairness in the comparison we consider that  $k_m$  is equal to the average loss aversion parameter value of all the members of the heterogeneous group. It is noted that the more loss

averse is the users' behavior (higher loss aversion parameter), the less number of jobs they offload at the SC (Equation 5.28). The opposite holds true for the risk seeking users, which may lead the SC to "failure", thus the users' expected prospect theoretic utility will decrease.

Indeed, based on Figure 5.10a and Figure 5.10b, the heterogeneous users led the system to higher congestion levels, as there is an increase in the average number of offloaded jobs at the SC and a decrease in the SC's pricing  $p_{sc}(\lambda_T)$  (Eq.5.22). However, in our case, Figure 3a illustrates that the increase of the average number of offloaded jobs at the SC led the heterogeneous users to achieve a higher average expected prospect theoretic utility compared to the homogeneous case.

### Comparative Analysis

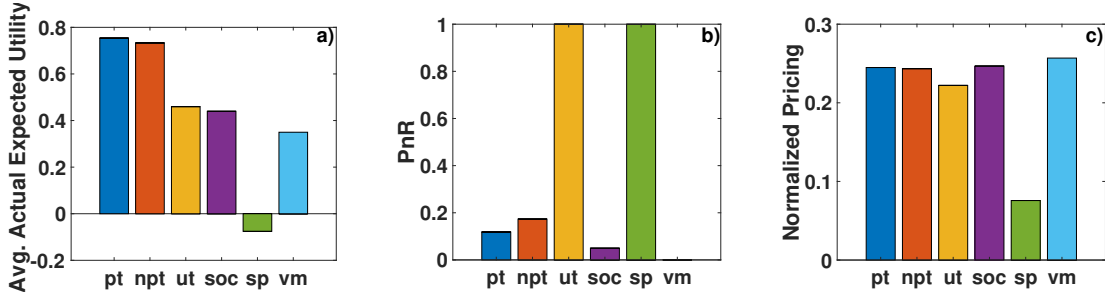


Figure 5.11: Comparative Evaluation - Impact of users' behavior

We present a comparative study of our proposed theoretic framework (that assumes prospect theoretic (pt) users) with five other alternatives, assuming user behaviors as follows: (a) non prospect theoretic (npt) users, but expected actual utility  $\mathbb{E}(z_i(\lambda_i, \lambda_{-i}))$  maximizers instead, taking into account the SC's probability of non-responsiveness, (b) actual utility maximizers (ut) users, where each user maximizes its actual utility (Equation 5.23) without considering the SC's probability of non-responsiveness, (c) social (soc) users, where each user based on its social factor  $f_i$  offloads  $f_i * J_i$  number of jobs at the SC, (d) (sp) users where each user  $i$  offloads all

of its jobs  $J_i$  at the SC, and (e) (vm) users where each user offloads all of its jobs  $J_i$  at the VMs.

The comparative evaluation is performed in terms of: (i) the expected actual utility, (Figure 5.11a), (ii) SC's probability of non-responsiveness (Figure 5.11b), and (iii) users' average normalized pricing, i.e.,  $\frac{P_{i,vm}^{(t)} + P_{i,sc}^{(t)}}{B_i^{(t)}}$  (Figure 5.11c). In particular, Figure 5.11a shows that both (pt) and the (npt) frameworks achieve a higher average actual expected utility (with (pt) slightly outperforming) compared to the rest of the approaches, due to the realistic consideration of the system's uncertainty (through the SC's probability of non-responsiveness). Both the (ut) and the (sp) frameworks, by ignoring the SC's probability of non-responsiveness, lead the SC to "failure", i.e.,  $PnR(\lambda_T) = 1$  (Figure 5.11b), and therefore these two approaches conclude to a lower user average actual expected utility compared to the (pt) and the (npt) (Figure 5.11a). Please note that although the (ut) approach leads the SC to "failure", the users still offload part of their jobs at the VMs, and therefore achieve a positive average actual expected utility, while on the other hand, under the (sp) alternative, users achieve a negative average expected actual utility since none of their jobs is executed successfully.

On the other hand, the users under the (soc) approach, by offloading some jobs simply based on their social factor  $f_i$ , do not lead the SC to "failure", however, conclude to a lower average actual expected utility compared to the (ut) approach, since they do not perform any optimization. Under (vm) alternative the SC option is not exploited and each user offloads all its jobs at the VMs. Thus its actual expected utility is its reference point, which is lower compared to the ones achieved by the (ut) and the (soc). Finally, it is stressed that the (pt) framework operates better than the (npt), achieving lower SC's probability of non-responsiveness (Figure 5.11b) and higher average expected actual utility (Figure 5.11a), because in the (npt) case, each user does not follow a risk-aware behavior and determines its best response strategy  $\lambda_i^*$  by only considering its guaranteed actual utility, and as a result, the

SC's utilization is better by the (pt) users.

Figure 5.11c presents the average users' normalized pricing for all the scenarios. In the (sp) case, the users by offloading all of their jobs at the SC perceive the lowest pricing  $p_{sc}^f$  per unit of time, while for the opposite reason highest price is experienced in the (vm) case. The (soc) users perceive the second-highest average normalized price since they offload a small portion of their jobs. Comparing the (npt) alternative with the (pt), we notice that they present very similar performance, with (npt) concluding to slightly lower average normalized price compared to the (pt), since by offloading a higher portion of their jobs at the SC, they perceive a lower price from the SC.

The same holds for the (ut) users, who offload a larger number of jobs at the SC compared to the (pt) and (npt) users (Figure 5.11b), and as a result, they perceive the second-lowest average normalized price (Figure 5.11c).

### 5.2.8 Summary

A novel risk-based distributed approach, towards determining each user's computing tasks optimal allocation strategy, in a social cloud computing environment offering both options of VM and SC computing, is designed. Based on the properties of Prospect Theory and the theory of The Tragedy of the Commons, we take into account the loss-averse and gain-seeking behavior of the users, as well as the uncertainty introduced due to the shared nature of the SC model. To address the decision-making problem at hand, a non-cooperative game is formulated among the users, where the goal of each user is to maximize its perceived expected prospect theoretic utility. The existence and uniqueness of the non-cooperative game's PNE are proven, and a distributed low-complexity algorithm that converges to the PNE is devised. Detailed numerical results were presented highlighting the performance benefits of our proposed approach.



## Chapter 6

# Simulation and Emulation Environment

In this chapter, the software libraries and platforms that were used in this dissertation towards developing scientific and robust simulation and emulation environments for the described empirical evaluations, are presented and described. The empirical evaluations were developed in Python [311], and in the programming platform MATLAB [312]. In particular, the simulation environments were developed in Python through the utilization of software libraries such as NumPy [313], SciPy [314], Pandas [315], Matplotlib [316], and PyTorch [317].

- NumPy is the fundamental package for scientific computing in Python. It is a Python library that provides a multidimensional array object, various derived objects (such as masked arrays and matrices), and an assortment of routines for fast operations on arrays, including mathematical, logical, shape manipulation, sorting, basic linear algebra, basic statistical operations, and random simulation.
- SciPy is a scientific library for Python for mathematics, science and engineering. The SciPy library depends on NumPy, which provides convenient and fast N-

dimensional array manipulation. It provides many user-friendly and efficient numerical practices such as routines for numerical integration and optimization.

- Pandas is a Python package that was utilized for data processing, as it provides fast, flexible, and expressive data structures designed to make working with structured (tabular, multidimensional, potentially heterogeneous) and time series data both easy and intuitive. Generally, Pandas aims to be the fundamental high-level building block for doing practical, real world data analysis in Python.
- Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python, and was used as our main software option for generating and illustrating our empirical evaluations.
- PyTorch is an open source Machine Learning library based on the Torch library, primarily developed by Facebook's AI Research lab. PyTorch was used in this dissertation for the development of Deep Reinforcement Learning, Supervised, Unsupervised, and Probabilistic models that were used either as our introduced decision-making frameworks and approaches, or for emulating the users' behavioral patterns (e.g., risk-aware attitudes) based on publicly available user data.

Our emulation environment was mainly developed in the programming platform MATLAB. In particular, the MATLAB toolbox for Wireless Communication was used for emulating concepts of over-the-air signals, whereas the Reinforcement Learning toolbox was utilized towards training policies using look-up tables, and for implementing the proposed decision-making frameworks. The considered and studied environments, e.g., Mobile Edge Computing, were designed and emulated through the MATLAB Simulink toolbox. Finally, the MATLAB Optimization Toolbox was utilized for finding parameters that minimize or maximize our introduced objectives with respect to the described constraints of the autonomous computing systems.

# Chapter 7

## Future Work

Our work thus far suggests several avenues for further research. Based on our results, we believe that our introduced autonomous decision-making frameworks can be extended to real-time applications that are characterized by resource-constrained computing systems and shared resources. Furthermore, we believe that our introduced theory and approaches that incorporate humans' behavioral characteristics, risk-aware attitudes, and propose more efficient and realistic equilibrium points for non-cooperative games, could serve as the underlying intelligence of resource-constrained computing systems designed to provide daily personalized assistance to their owners.

In particular, part of our current and future work includes the extension of the introduced autonomous decision-making frameworks for Mobile Edge Computing (MEC). The uncertain mobility of mobile devices (users), and their stochastic computation demands alongside the dynamic network and computation conditions of the MEC environment, arise additional key themes. In particular, mobile devices' decision-making regarding MEC servers' selection to perform their computation offloading, under the setting of a multiple MEC servers environment, and the required computation and communication resource allocation influence their experienced completion latency and energy consumption for their applications. Towards this direction, in the following chapter, we present our current research work, which could be

considered as an extension of the introduced decision-making frameworks in Chapter 3. Specifically, the combined problem of mobile users' association to MEC servers, and mobile users' computation offloading problem, are investigated jointly and addressed through a two-stage Deep Reinforcement Learning and risk-aware framework.

## 7.1 A Deep Reinforcement Learning Approach for Risk-aware Orchestration in MEC

A full-fledged risk-aware orchestration of multiple mobile users is proposed for the setting of a stochastic and dynamic multiple MEC servers environment. The proposed framework consists of two stages, thus the mobile users association to MEC servers (first stage), and mobile users' computation offloading to the corresponding MEC servers (second stage). The mobile users association to MEC servers over the long time horizon is formalized as a Markov Decision Process (MDP) and is performed through an SDN controller that acts as a DRL decision-making agent (DRL-SDN) empowered with a powerful function approximator (neural network). The DRL-SDN agent trains its neural network in an online fashion towards approximating the optimal  $Q$ -values (Chapter 2.2), by exploiting its perceived feedback from the environment. The DRL-SDN agent associates the mobile users to the MEC servers through a stochastic  $\epsilon$ -greedy policy, which is adjusted over time, and the DRL-SDN agent determines an optimal association policy that optimizes the mobile user's perceived cumulative completion latency and energy consumption over the long time horizon. Each cluster of a MEC server and the associated mobile users is treated independently, and the mobile users' computation offloading is performed under the consideration of the MEC server's computing uncertainty (Chapter 3), and through the introduced distributed algorithm for convergence to the unique Pure Nash Equilibrium point (Algorithm 2 in Chapter 3.5).

### 7.1.1 Contributions

Towards extending our introduced decision-making frameworks in Chapter 3, we incorporate mobile users' behavioral factors in both mobile users' association with the MEC servers and mobile users computation offloading problem. The two problems are decoupled and addressed through a two-stage approach over the long time horizon, and through the consideration of a DRL-SDN enabled MEC framework. The main contributions are summarized below:

1. A stochastic and dynamic multiple mobile users and multiple MEC servers environment is considered. The joint problem of mobile users' association to MEC servers, alongside the former's computation offloading after the association, is considered. In contrast with the vast majority of the research literature (Chapter 3.1) the two problems are decoupled. The latter allows us to investigate the considered joint problem under a more realistic setting compared to the already adopted environments in Chapter 3, where the mobile users' characteristics and demands, as long as the MEC servers' key computing attributes change stochastically over time.
2. The joint problem is studied over the long time horizon, thus over multiple timeslots, and the mobile users' association to the MEC servers is formulated as a Markov Decision Process (MDP) [97]. The latter problem is handled by a centralized Deep Reinforcement Learning (DRL) SDN controller, i.e., DRL-SDN agent, that is empowered with a powerful function approximator (neural network). The DRL-SDN agent seeks an optimal association policy through online training, which optimizes the mobile user's long-term cumulative perceived completion latency and energy consumption by offloading parts of her applications to the MEC servers over the considered timeslots.
3. The DRL-SDN agent's neural network is utilized as a Deep Q-Network (DQN) (Chapter 2.2) and approximates the optimal Q-values, from which an optimal association policy is implicitly defined. At each timeslot, during the first stage, the mobile

## Chapter 7. Future Work

user's information, e.g., mobility, application's demands, risk-aware characteristics, and the MEC servers' computing characteristics constitute the mobile user's state representation, and is used as input to the DRL-SDN agent's DQN. The Q-values of the actions is determined as the output of the DQN, where the actions are all the different MEC servers, i.e., association options. The Q-value of each action expresses the mobile user's expected perceived cumulative completion latency and energy consumption after being associated with the particular MEC server. The DRL-SDN agent uses the Q-values and associates the mobile user to a MEC server through its  $\epsilon$ -greedy policy [62], which is adjusted over time and expresses the DRL-SDN agent's trade-off between exploration and exploitation.

4. After each mobile user's association to a MEC server, each cluster of MEC server and associated mobile users is treated independently, and the latter's computation offloading constitutes the second stage of the proposed approach. Each mobile user can offload an arbitrarily part of its application to the MEC server that has been associated with. Given the computing uncertainty of each MEC server that mainly stems from the latter's shared nature and limited computation and storage capabilities, each MEC server is treated as a Common Pool of Resources (CPR). The inherited computing uncertainty is captured through Prospect Theory, where each mobile user's perceived completion latency and the energy consumption is formulated in a probabilistic manner (expected prospect-theoretic utility), and through the mobile user's risk-aware characteristics' consideration. Each mobile user's computation offloading decision-making that optimizes its expected prospect-theoretic utility is formulated as a non-cooperative game among the MEC server's associated mobile users. A distributed decision-making framework (Algorithm 3.5.3 in Chapter 3) is utilized for converging to the unique Pure Nash Equilibrium (PNE) point of the non-cooperative game.

5. Each mobile user after performing its computation offloading, receives her actual overall completion latency and energy consumption. The latter constitutes the DRL-

## Chapter 7. Future Work

SDN agent's corresponding associated reward for its action, thus for associating the mobile user with the particular MEC server. The DRL-SDN agent stores its observed interactions for each mobile user in the form  $(state, action, next\ state, reward)$  into a single replay memory buffer.

6. The DRL-SDN agent's online training consists of randomly sampled batches of interactions from the replay memory buffer, which are used for evaluating the *Temporal Difference Loss* function (Equation 2.28), whose gradient is used for updating the DQN's parameters through gradient descent. The updating of the DQN's parameters with the use of interactions of different mobile users better exploits the power of neural networks and defines an additional generalization dimension over the mobile users. The latter constitutes another differentiation of our work with the vast majority of the research literature, which commonly assumes that each mobile user's device act as an RL agent that is trained on its observed interactions. Despite that, the consideration of a centralized DRL-SDN controller addresses the latter commonly made assumption, which may not be realistic in a variety of networking scenarios [318], due to each mobile user's device's limited energy availability and computing capability.

7. A series of experiments are performed to evaluate the performance and the inherent attributes of the proposed DRL-SDN-enabled MEC framework that addresses the problem of the mobile users' association with the MEC servers and mobile users' computation offloading problem under the setting of a dynamic and stochastic environment. A comparative study demonstrates its superiority and benefits, in terms of the mobile user's perceived overall completion latency and energy consumption throughout the long-time horizon.

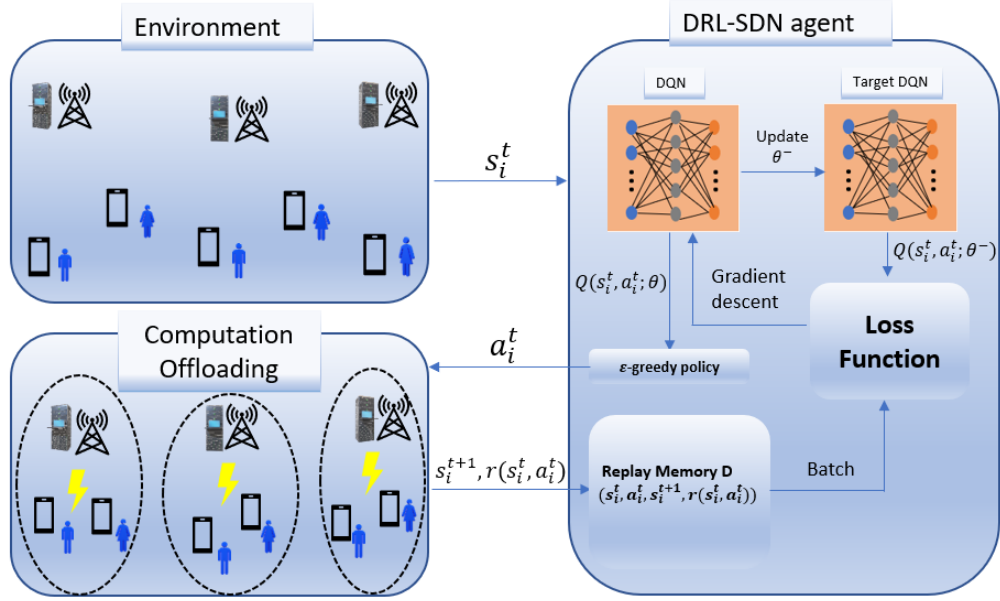


Figure 7.1: DRL-SDN enabled MEC environment

### 7.1.2 System Model

A DRL-SDN enabled multiple MEC servers and multiple mobile users system, as shown in Fig. 7.1, is considered. Following the existing literature in MEC (Chapter 3.1), the MEC servers could be located at the Macro Base Stations (MBSs) of the macrocells or the Access Points (APs) of the small cells, e.g., femtocells [176, 179]. The whole operation of the examined system is divided into time slots (e.g., different time points throughout a day), where  $\mathcal{T} = \{1, \dots, t, \dots, T\}$  denotes their corresponding set. At each time slot, each mobile user's association with a MEC server is performed by the SDN controller, which exploits the mobile user's and MEC environment's information, and acts as a Deep Reinforcement Learning (DRL) agent with a global view of the considered system. As a result, after each mobile user's association with a MEC server through the DRL-SDN agent, each MEC server has been associated with a cluster of mobile users. Moreover, regarding mobile users' computation offloading, we adopt the described setting in Chapter 3.5, thus each mobile user is assumed able to offload part of her application to the MEC server that



## Chapter 7. Future Work

has been associated with, whereas the remaining part of her application is processed locally at her device.

In our model, we denote by  $\mathcal{U} = \{1, \dots, i, \dots, U\}$  the set of mobile users, and with  $\mathcal{S} = \{1, \dots, s, \dots, S\}$  the set of MEC servers in the system. We further denote by  $\mathcal{A}_i^t = (B_i^t, \phi_i^t)$  the mobile user's  $i$  application at timeslot  $t$ , which is characterized by: a)  $B_i^t$  the amount of the inputs bits (data to be processed), and b)  $\phi_i^t \cdot B_i^t$  the required Cpu-Cycles, where  $\phi_i^t \in [\frac{\text{Cpu-Cycles}}{\text{bit}}]$ . In particular,  $\phi_i^t > 0$  describes the level of the mobile user's  $\mathcal{A}_i^t$  computing intensity, and a higher value of  $\phi_i^t$  expresses a more computing demanding application. Furthermore, at each time slot  $t$ , each mobile user's local device is characterized by a limited energy availability  $\bar{e}_i^t$  (associated with the actual device's remaining battery), which decreases over time slots, thus  $\bar{e}_i^t > \bar{e}_i^{t+1}$ , as we do not consider any energy harvesting capability for the mobile user's device.

At each time slot  $t \in \mathcal{T}$ , after the mobile users' association with the MEC servers through the DRL-SDN agent, a cluster of mobile users denoted as  $\mathcal{U}_s^t$ , where  $\mathcal{U}_s^t \subseteq \mathcal{U}$ , is associated with the MEC server  $s \in \mathcal{S}$ , while with  $\alpha_i^t$  we denote the mobile user's  $i$  MEC sever  $s$  that has been associated with, thus  $\alpha_i^t \in \mathcal{S}$ , and  $\alpha_i^t = s, \forall i \in \mathcal{U}_s$ . In this work, following the recent existing literature in MEC, we study mobile users' computation offloading under the setting of partial offloading. In other words, as in Chapter 3, mobile users' applications can be arbitrarily partitioned into subsets of any size, so mobile user  $i$  is able to offload a part  $b_{i,s}^t \in [0, B_i^t]$  of hers application  $\mathcal{A}_i^t$  to the MEC server  $s$ , where  $s = \alpha_i^t$ , for remote execution, and keep the remaining part  $(B_i^t - b_{i,s}^t)$  for local processing. As a result, we have  $b_{i,s}^t = 0$  if mobile user  $i$  decides to compute its whole application  $\mathcal{A}_i^t$  locally. We adopt the computation and communication models that were introduced in Chapter 3.5, and for clarity purposes, a summarized description of them is given in the following. Moreover, for the rest of the analysis regarding mobile users computation offloading, we drop the time slot notation  $t$  for notational convenience since the same hold true  $\forall t \in \mathcal{T}$ .

**Communication Model** We consider a typical interference limited communication environment, where each MEC server  $s \in \mathcal{S}$  operates and receives data over a dedicated communication link [319], thus each mobile user  $i$  that is associated with the MEC server  $s$ , while transmitting hers data  $b_{i,s}$  senses the interference  $\sum_{j \in \mathcal{U}_s, j \neq i} p_{j,s} \cdot g_{j,s}$  from the mobile users  $\mathcal{U}_s$ , who are the ones associated and transmitting to the same MEC server  $s$ , thus  $a_j = s, \forall j \in \mathcal{U}_s$ . The communication channel gain between the mobile user  $i$  and the MEC server  $s$  is denoted by  $g_{i,s}$ , and  $p_{i,s}$  is the mobile user's  $i$  transmission power to offload hers data  $b_{i,s}$  to MEC server  $s$ . Moreover, given the bandwidth  $W_s$  allocated to to MEC server's  $s$  communication link, the mobile user's  $i$  achievable data rate, while offloading hers  $b_{i,s}$  is given as:

$$R_{i,s} = W_s \cdot \log \left( 1 + \frac{p_{i,s} \cdot g_{i,s}}{\sigma_0^2 + \sum_{j \in \mathcal{U}_s, j \neq i} p_{j,s} \cdot g_{j,s}} \right) \quad (7.1)$$

, where  $\sigma_0^2$  indicates the variance of the Additive White Gaussian Noise (AWGN) of the MEC server  $s$ . To this end, mobile user  $i$  by offloading  $b_{i,s}$  amount of data to the MEC server  $s$  experiences an overhead consisting of the transmission latency of the data

$$O_{i,s}^{l,tr} = \frac{b_{i,s}}{R_{i,s}} [\text{sec}] \quad (7.2)$$

, and the corresponding transmission energy consumption

$$O_{i,s}^{e,tr} = \frac{b_{i,s} \cdot p_{i,s}}{R_{i,s}} [\text{J}] \quad (7.3)$$

**Computing Model** We assume that each MEC server  $s \in \mathcal{S}$  is empowered with a strong computing resource, e.g., high-speed CPU, and is able to process the received offloaded data in parallel. Moreover, each MEC server's  $s$  computing capability is limited by the total amount of data  $B_s$  that can process at the same time, e.g., due to either limited memory storage or finite multi-core architecture, as this was described

## Chapter 7. Future Work

in Chapter 3.5. The computing capability of each MEC server  $s$ , which is denoted by  $\bar{f}_s[\frac{\text{Cpu-Cycles}}{\text{sec}}]$ , is shared among the mobile users  $\mathcal{U}_s$ , thus the ones that have been associated and offload parts of their applications to this MEC server  $s$ . As a result, each mobile user  $i \in \mathcal{U}_s$  perceives a computing capability by the MEC server  $s$  in order to remotely execute hers offloaded data  $b_{i,s}$ , and it is defined through mobile user's  $i$  return function  $f_{i,s}(\bar{b}_s)$  that is given as follows (Equation 3.31):

$$f_{i,s}(\bar{b}_s) = \frac{\phi_i}{\sum_{j \in \mathcal{U}_s} \phi_j} \cdot f_s(\bar{b}_s) \left[ \frac{\text{Cpu-Cycles}}{\text{sec}} \right] \quad (7.4)$$

, where  $\bar{b}_s$  denotes the total amount of offloaded data to the MEC server  $s$ , thus  $\bar{b}_s = \sum_{i \in \mathcal{U}_s} b_{i,s}$ , and  $f_s(\bar{b}_s)$  defines MEC server's  $s$  production function, which is given as follows (Equation 3.32):

$$f_s(\bar{b}_s) = \begin{cases} (1 - \frac{\bar{b}_s}{\mathcal{B}_s}) \cdot \bar{f}_s & , \text{ if } \bar{b}_s < \mathcal{B}_s \\ 0 & , \text{ otherwise} \end{cases} \quad (7.5)$$

, where  $\mathcal{B}_s$  is the MEC server's  $s$  total amount of data that can process without failing its operation. As a result, each mobile user  $i$  receives a computation capability  $f_{i,s}(\bar{b}_s)$  by the MEC server  $s$ , which is personalized based on hers application's  $\mathcal{A}_i$  computing demands  $\phi_i$  (e.g., mobile users with applications that have higher computing demands will receive higher computing capabilities). Moreover, mobile user's  $i$  assigned computing capability  $f_{i,s}$  decreases with respect to the total amount of offloaded data  $\bar{b}_s$ , as in a realistic scenario an MEC server  $s$  is characterized by a threshold amount of data, i.e.,  $\mathcal{B}_s$ , that can process at the same time, and as  $\bar{b}_s$  increases, the MEC server  $s$  becomes more over-exploited, thus the computing capability that each mobile user perceives decreases.

**Actual Overhead:** As a result, mobile user  $i$  receives an  $f_{i,s}$  computing capability by MEC server  $s$  in order to remotely execute hers offloaded data  $b_{i,s}$ , and the corresponding execution latency is  $O_{i,s}^{l,ex} = \frac{\phi_i \cdot b_{i,s}}{f_{i,s}} [\text{sec}]$ . Additionally, by considering

## Chapter 7. Future Work

mobile user's  $i$  transmission latency (Eq. 7.2), then mobile user's  $i$  overall latency overhead by offloading  $b_{i,s}$  to the MEC server  $s$  is formulated as follows:

$$O_{i,s}^l = O_{i,s}^{l,tr} + O_{i,s}^{l,ex} = b_{i,s} \left( \frac{1}{R_{i,s}} + \frac{\phi_i}{f_{i,s}} \right) \quad (7.6)$$

Furthermore, mobile user  $i$  by offloading  $b_{i,s}$  to MEC server  $s$ , executes the remaining part  $(B_i - b_{i,s})$  of its application locally on hers device. In particular, by denoting as  $dc_i[\frac{\text{Cpu-Cycles}}{\text{sec}}]$ , and  $de_i[\frac{\text{J}}{\text{Cycle}}]$  mobile user's local device's computing capability and energy consumption respectively, the local computing execution latency is given as follows:

$$O_{i,d}^l = \frac{\phi_i \cdot (B_i - b_{i,s})}{dc_i} \quad (7.7)$$

, and device's local energy consumption is determined as:

$$O_{i,d}^e = \phi_i \cdot (B_i - b_{i,s}) \cdot de_i \quad (7.8)$$

To this end, based on Eq. 7.6 and Eq. 7.7, mobile user's  $i$  overall latency overhead is formalized as:

$$O_i^l = O_{i,s}^l + O_{i,d}^l = b_{i,s} \left( \frac{1}{R_{i,s}} + \frac{\phi_i}{f_{i,s}} \right) + \frac{\phi_i(B_i - b_{i,s})}{dc_i} \quad (7.9)$$

, and based on Eq. 7.3 and Eq. 7.8, mobile user's  $i$  overall energy overhead is given as:

$$O_i^e = \frac{b_{i,s} p_{i,s}}{R_{i,s}} + \phi_i(B_i - b_{i,s}) de_i \quad (7.10)$$

Furthermore, both the latency and energy overheads are of high importance in mobile user's computation offloading decision, and in this work we are interested on accounting both of them, as well as the emerging tradeoffs. In particular, based on Eq. 7.9 and Eq. 7.10, we express the mobile user's latency and energy overheads through the actual overhead function, which is defined as follows:

$$O_i(b_{i,s}, \bar{b}_s) = w_i^l \cdot \frac{O_i^l}{\bar{t}} + w_i^e \cdot \frac{O_i^e}{\bar{e}_i} \quad (7.11)$$

, where  $w_i^l, w_i^e \in [0, 1], w_i^l + w_i^e = 1$ , denote the weights of mobile user's  $i$  application's  $\mathcal{A}_i$  completion latency, and device's energy consumption, respectively, that can be tuned by the mobile user according to different priorities and considerations (e.g., low battery consideration  $w_i^e > w_i^l$ ). Moreover,  $\bar{e}_i$  is mobile user's device's current available energy (e.g., actual remaining battery), and  $\bar{t}$  is the duration of time slot  $t \in \mathcal{T}$ . The above-normalized formalization of mobile user's overall overhead guarantees the same order of magnitude of the two types of overhead (latency, energy), which in principle are of different nature. Moreover, in contrast with Chapter 3.5, where a static scenario was considered, here each mobile user at each time slot can dynamically adjust the corresponding weights  $w_i^e, w_i^l$ , based on her application's completion latency importance, and device's energy consumption significantly.

### 7.1.3 Mobile Users Computation Offloading

As described in Chapter 3.5.2, each MEC server  $s \in \mathcal{S}$  constitutes a Common Pool of Resources (CPR) (Chapter 2.3), and it is characterized by a threshold amount of data  $B_s^t$  (e.g., storage limitations) that is able to process at each timeslot  $t \in \mathcal{T}$  without its operation to fail. Moreover, each MEC server's  $s$  threshold amount of data  $\mathcal{B}_s^t$  cannot be deterministically known by the mobile users, and the latter constitutes a computing uncertainty. As a result, each mobile user's perceived completion latency and energy consumption by offloading a part of its application to the MEC server that has been associated with is formulated in a probabilistic manner through Prospect Theory, and by considering the mobile user's risk-aware characteristics. As before, since here we focus on mobile user's computation offloading, a specific time slot  $t \in \mathcal{T}$  is considered, and a static association between mobile users and MEC servers is assumed. Thus, for each MEC server  $s \in \mathcal{S}$ , a corresponding associated group of mobile users  $\mathcal{U}_s^t$  is considered, where each mobile user  $i \in \mathcal{U}_s^t$  is able to offload an arbitrarily part  $b_{i,s}^t$  of its application  $\mathcal{A}_i^t$  to the MEC server  $s$  for remote execution. As a result, the time slot notation  $t$  is dropped for notational convenience, since the

same hold true  $\forall t \in \mathcal{T}$ .

Each MEC server's computing uncertainty to process the mobile users' offloaded data, is captured through its probability of failure, which is defined with respect to the associated mobile users' offloaded data, i.e.,  $\bar{b}_s$ , as it represents the MEC server's  $s$  probability to fail serving the mobile users' offloaded data due to its over-exploitation of its computing capabilities. As in Chapter 3.5, each MEC server's  $s$  (CPR) probability of failure  $p_s(\bar{b}_s)$  is strictly increasing, convex and twice continuously differentiable with respect to  $\bar{b}_s \in [0, \mathcal{B}_s)$ , with  $p_s(\bar{b}_s) = 1, \forall \bar{b}_s \geq \mathcal{B}_s$ . Here, without loss of generality, we consider a linear probability of failure function, thus  $p_s(\bar{b}_s) = \frac{\bar{b}_s}{\mathcal{B}_s}, \forall \bar{b}_s < \mathcal{B}_s$ , while  $p_s(\bar{b}_s) = 1, \forall \bar{b}_s \geq \mathcal{B}_s$ . To this end, the MEC server  $s$  will deterministically fail to process the associated mobile users' offloaded data  $\bar{b}_s$ , if the latter exceeds the MEC server's  $s$  computation capacity, i.e.,  $\mathcal{B}_s$ . On the other hand, in the case where the associated mobile users' offloaded data does not exceed the MEC server's  $s$  computation capacity, i.e.,  $\bar{b}_s < \mathcal{B}_s$ , then the MEC server's  $s$  probability of failure is not zero, but it probabilistically depends on the amount of offloaded data that it needs to process, i.e.,  $p_s(\bar{b}_s) = \frac{\bar{b}_s}{\mathcal{B}_s}$ , as the mobile users do not deterministically know the actual threshold amount of data  $\mathcal{B}_s$  of the MEC server  $s$ , while they are making their computation offloading decisions. The latter mainly stems from the fact that a MEC server's  $s$  threshold amount of data  $\mathcal{B}_s$  is time variant, as in a realistic scenario a MEC server performs a wide range of computations [320], and not only mobile users' offloaded data processing, thus the MEC server's computing capacity dynamically changes over time.

Following the principles of Prospect Theory, each mobile user's reference point constitutes a "safe" outcome for the mobile user, and here this reference point expresses the corresponding actual overhead that the mobile user would have perceived if processed locally her whole application, i.e.,  $B_i$ . In other words, by setting  $b_{i,s} = 0$

## Chapter 7. Future Work

in Eq. 7.11, each mobile user's  $i$  reference point is defined as follows:

$$\begin{aligned} q_{i,0} &= w_i^l \cdot \frac{\phi_i \cdot B_i}{dc_i \cdot \bar{t}} + w_i^e \cdot \frac{\phi_i \cdot B_i \cdot de_i}{\bar{e}_i} \\ &= \phi_i \cdot B_i \cdot \left[ w_i^l \cdot \left( \frac{1}{\bar{t} \cdot dc_i} - \frac{de_i}{\bar{e}_i} \right) + \frac{de_i}{\bar{e}_i} \right] \end{aligned} \quad (7.12)$$

, where  $w_i^e$  has been replaced by  $(1 - w_i^l)$ .

Furthermore, mobile user's prospect-theoretic utility function by offloading  $b_{i,s}$  to the MEC server  $s$  is formulated as follows:

$$u_{i,s}(q_{i,s}) = \begin{cases} (q_{i,0} - q_{i,s})^{\alpha_i} & , \text{ if } q_{i,s} \leq q_{i,0} \\ -k_i \cdot (q_{i,s} - q_{i,0})^{\gamma_i} & , \text{ if } q_{i,s} > q_{i,0} \end{cases} \quad (7.13)$$

, where  $q_{i,s} = O_i(b_{i,s}, \bar{b}_s)$  is the mobile user's  $i$  actual overhead perceived by the MEC server  $s$  by offloading  $b_{i,s}$  amount of data, as defined in Eq. 7.11, and  $q_{i,0}$  is the mobile user's  $i$  reference point, as defined in Eq. 7.12. Each mobile user  $i$  aims to autonomously determine hers computation offloading decision  $b_{i,s}$ , such that hers perceived prospect-theoretic utility  $u_{i,s}$  is maximized. The real parameters  $\alpha_i, \gamma_i \in (0, 1]$  describe mobile user's  $i$  sensitivity in gains and losses (Chapter 3.5.2), respectively, and these ones are expressed through mobile user's experienced actual overhead  $q_{i,s}$  (Eq. 7.11), and hers reference point  $q_{i,0}$  (Eq. 7.12).

Considering the case that the MEC server  $s$  does not fail to process the associated mobile users'  $\mathcal{U}_s$  offloaded data  $\bar{b}_s$ , then the mobile user's  $i$  perceived actual overhead  $q_{i,s}$  is assumed lower than hers reference point  $q_{i,0}$ , which corresponds to the overall actual overhead for processing the whole application, i.e,  $b_{i,s} = 0$ , locally, as the MEC server's  $s$  computation capability  $\bar{f}_s$  is significantly higher compared to mobile user's device's computation capability  $dc_i$ , thus  $q_{i,s} \leq q_{i,0}$ . Therefore, based on the first branch of mobile user's prospect-theoretic utility, and by substituting Eq. 7.11 from Eq. 7.12, we have that  $u_{i,s} = b_{i,s}^{\alpha_i} \cdot \left[ \frac{w_i^l}{\bar{t}} \left( \frac{\phi_i}{dc_i} - \frac{\phi_i}{f_{i,s}} - \frac{1}{R_{i,s}} \right) + \frac{1-w_i^l}{\bar{e}_i} \cdot (\phi_i de_i - \frac{p_{i,s}}{R_{i,s}}) \right]^{\alpha_i}$ . On the other hand, if the MEC server  $s$  fails to process mobile users' offloaded data, then each mobile user's  $i$  perceived actual overhead is  $q_{i,s} = q_{i,0} + \frac{w_i^l}{\bar{t}} \frac{b_{i,s}}{R_{i,s}} + \frac{1-w_i^l}{\bar{e}_i} \frac{b_{i,s} p_{i,s}}{R_{i,s}}$ ,

## Chapter 7. Future Work

thus it is consisted of the overall local computing  $q_{i,0}$ , as the mobile user  $i$  process hers application locally due to MEC server's  $s$  failure, and the extra communication overhead, i.e., latency and energy transmission overheads (Eq. 7.2 & 7.3), as initially the mobile user offloads hers  $b_{i,s}$  data to the MEC server  $s$  before the latter's failure. As a result, in the MEC server's  $s$  failure case, it holds true that  $q_{i,s} > q_{i,0}$ , and based on the second branch of Eq. 7.13, we have that  $u_{i,s} = -k_i \cdot b_{i,s}^{\alpha_i} \left[ \frac{1}{R_{i,s}} \left( \frac{w_i^l}{t} + \frac{1-w_i^l}{\bar{e}_i} p_{i,s} \right) \right]^{\alpha_i}$ .

For notational convenience,  $h_{i,s} \triangleq \left[ \frac{w_i^l}{t} \left( \frac{\phi_i}{dc_i} - \frac{\phi_i}{f_{i,s}} - \frac{1}{R_{i,s}} \right) + \frac{1-w_i^l}{\bar{e}_i} \cdot \left( \phi_i de_i - \frac{p_{i,s}}{R_{i,s}} \right) \right]^{\alpha_i}$ , considering that  $h_{i,s} > 0$  if the MEC server  $s$  does not fail, and we define as  $\lambda_{i,s} = \left[ \frac{1}{R_{i,s}} \left( \frac{w_i^l}{t} + \frac{1-w_i^l}{\bar{e}_i} p_{i,s} \right) \right]^{\alpha_i}$ . As a result, by considering the MEC server's  $s$  probability of failure  $p_s(\bar{b}_s)$ , the mobile user's  $i$  prospect-theoretic utility function can be rewritten as:

$$u_{i,s} = \begin{cases} b_{i,s}^{\alpha_i} \cdot h_{i,s}(\bar{b}_s) & , \text{ with prob. } 1 - p_s(\bar{b}_s) \\ -k_i \cdot \lambda_{i,s} \cdot b_{i,s}^{\alpha_i} & , \text{ with prob. } p_s(\bar{b}_s) \end{cases} \quad (7.14)$$

To this end, the mobile user's expected prospect-theoretic utility is given as follows:

$$\begin{aligned} \mathbb{E}[u_{i,s}] &= b_{i,s}^{\alpha_i} \cdot h_{i,s}(\bar{b}_s)(1 - p_s(\bar{b}_s)) - k_i \cdot \lambda_{i,s} \cdot b_{i,s}^{\alpha_i} \cdot p_s(\bar{b}_s) \\ &= b_{i,s}^{\alpha_i} \cdot er_{i,s}(\bar{b}_s) \end{aligned} \quad (7.15)$$

, where  $er_{i,s}(\bar{b}_s) = h_{i,s}(\bar{b}_s)(1 - p_s(\bar{b}_s)) - k_i \lambda_{i,s} p_s(\bar{b}_s)$  is the effective rate of return of the MEC server  $s$  for the mobile user  $i$ .

The mobile user  $i \in \mathcal{U}$  associated with a MEC server  $s \in \mathcal{S}$ , thus  $i \in \mathcal{U}_s \subseteq \mathcal{U}$ , aims at autonomously determining hers computation offloading decision  $b_{i,s}$  such that hers perceived overall overhead (Eq. 7.11) is minimized. Considering the computing uncertainty that stems from each MEC server  $s$ , and mobile users' risk-aware behavioral attitudes under uncertainty, this problem can be formulated as a maximization problem of each mobile user's expected prospect-theoretic utility function (Eq. 7.15) as follows:

$$\max_{b_i \in [0, B_i]} \mathbb{E}[u_{i,s}] = \max_{b_i \in [0, B_i]} b_{i,s}^{\alpha_i} \cdot er_{i,s}(\bar{b}_s) \quad (7.16)$$



As in Chapter 3.5, the above maximization problem can be treated as a non-cooperative game. In particular, considering a MEC server  $s$ , and its corresponding group of associated mobile users  $\mathcal{U}_s$ , where  $a_i = s, \forall i \in \mathcal{U}_s$ , let us denote as  $\mathcal{G}_s = \{\mathcal{U}_s, \{\Gamma_i\}_{i \in \mathcal{U}_s}, \mathbb{E}[u_{i,s}]\}$  the non-cooperative game among the mobile users  $\mathcal{U}_s$ , where each mobile user's strategy space is  $\Gamma_i = [0, \min(B_i, \mathcal{B}_s)]$ , and her perceived payoff is the expected prospect-theoretic utility (Eq. 7.15). Towards solving the aforementioned non-cooperative game  $\mathcal{G}_s$ , the concept of Nash Equilibrium (NE) is adopted. Specifically, the computation offloading vector  $\mathbf{b}_s^* = (b_{1,s}^*, \dots, b_{i,s}^*, \dots, b_{U_s,s}^*)$  constitutes the NE point of the non-cooperative game  $\mathcal{G}_s$  if and only if no mobile user  $i \in \mathcal{U}_s$  has the incentive to change her own computation offloading strategy given the computation offloading strategies of the rest of the mobile users. Let  $\mathbf{b}_{-i,s}^* = (b_{1,s}^*, \dots, b_{i-1,s}^*, b_{i+1,s}^*, \dots, b_{U_s,s}^*)$  denote mobile users computation offloading strategies except for mobile user  $i$  at the NE point. The NE point of each non-cooperative game  $\mathcal{G}_s$  exists and is unique (Chapter 3.5.3).

Each non-cooperative game  $\mathcal{G}_s$  belongs to the class of *best-response potential games*, thus the sequential best response dynamics converge to the PNE of the game  $\mathcal{G}_s$  [228]. To this end, towards solving the aforementioned non-cooperative game  $\mathcal{G}_s$  with the consideration of each mobile user's risk-aware characteristics, the distributed decision-making framework introduced in Chapter 3.5.3 is adopted. In particular, at each iteration of the Algorithm 2 the selected mobile user  $i$  receives the total amount of offloaded data of the mobile users  $\mathcal{U}_s$ , i.e.,  $\bar{b}_s$ , and the interference factor  $\sum_{j \in \mathcal{U}_s} p_{j,s} \cdot g_{j,s}$  from the mobile user  $j$  that was selected at the previous iteration  $\tau - 1$ . The mobile user  $i$  then determines the amount of offloaded data of the rest mobile users, i.e.,  $\bar{b}_{-i,s} = \bar{b}_s - b_{i,s}$ , and then checks if her computation offloading best response is zero, i.e., whether  $\bar{b}_{-i,s} \geq b_{i,s}^{th}$ . The latter holds true if and only if  $er_{i,s}(\bar{b}_{-i,s}) \leq 0$ . If the mobile user  $i$  finds that  $\bar{b}_{-i,s} < b_{i,s}^{th}$ , then her computation offloading best response  $b_{i,s}^*$  exists and is single-valued (Theorem 6). In particular, due to the existence of the unique root of  $\frac{\partial \mathbb{E}[u_i]}{\partial b_{i,s}} = 0$ , and based on the fact that

## Chapter 7. Future Work

$\frac{\partial^2 \mathbb{E}[u_i]}{\partial b_{i,s}^2} < 0$ , thus  $\frac{\partial \mathbb{E}[u_i]}{\partial b_{i,s}}$  is strictly decreasing with respect to  $b_{i,s}$ , then the unique root  $r_{i,s}^*$  can be found through binary search into  $[0, \mathcal{B}_s]$  with an approximation  $\epsilon$ , such that  $\epsilon \rightarrow 0$ , and the mobile user's  $i$  computation offloading best response strategy to be  $b_{i,s}^* = \min(B_i, r_{i,s}^*)$ . Finally, after the mobile user has determined hers computation offloading decision strategy  $b_{i,s}^*$ , then the total amount of offloaded data is updated, i.e.,  $\bar{b}_s = \bar{b}_{-i,s} + b_{i,s}^*$ , and the interference factor in the case where  $b_{i,s}^* = 0$ . The latter ones are shared with the rest of the mobile users through intra-channel broadcasted signals [319].

The complexity of the binary search is  $\mathcal{O}(\log_2 \mathcal{B}_s)$  [229]. Moreover, given that  $U_s$  mobile users execute the following computation offloading algorithm, and given that the rest of operations involve arithmetical calculations, the overall complexity for converging to the PNE of the non-cooperative game  $\mathcal{G}_s$  is  $\mathcal{O}(U_s * \mathbb{T} * \log_2 B_s)$ , where  $\mathbb{T}$  are the number of iterations needed for convergence.

### 7.1.4 Mobile Users Association to MEC Servers: a Markov Decision Process

At each timeslot  $t \in \mathcal{T}$ , each mobile user  $i \in \mathcal{U}$  must be associated with a MEC server  $s \in \mathcal{S}$  to perform hers computation offloading decision regarding hers application  $\mathcal{A}_i^t$ . The mobile user's association with a MEC server  $s$  impacts significantly the former's experienced completion latency (Eq. 7.9) for hers application, and hers device's energy consumption (Eq. 7.10). Furthermore, in a stochastic, time-variant, and heterogeneous multiple MEC servers environment, several mobile user's, and environment's characteristics, e.g., user mobility, computing demands, computation, and communication conditions, change over time, and they constitute the mobile user's assignment problem not straightforward. Specifically, mobile users' association with MEC servers can be viewed as a matching problem, whose combinatorial complexity makes it infeasible to be addressed by regular matching theory [321].

## Chapter 7. Future Work

For that reason, many research works focus on solving the assignment problem by considering either preference functions [322, 323], e.g., transmission power, which drives their matching-based approaches, or through decentralized learning schemes that utilize heuristic-crafted preference functions [324]. On the other hand, many research works [325, 326, 327] incorporate mobile users association with MEC servers into mobile users computation offloading problem, where the two problems are solved and optimized jointly, but by not considering a full stochastic MEC environment due to the complexity of the problem. The latter mainly stems from the fact that in a realistic stochastic scenario, several time-variant factors should be taken under consideration in the optimization, which makes the dimensionality of the assignment problem a key limitation that is not properly being addressed in the described research works. Despite that, the majority of the research works solve the assignment problem for a single time instance, whereas the mobile users association with MEC servers should be properly addressed over the long time horizon by considering the underlying stochasticity of the environment, i.e., environment's dynamics.

To this end, here we utilize Deep Reinforcement Learning (Chapter 2.2.2) towards solving the mobile users association with the MEC servers problem over the long time horizon. Considering the unknown dynamics of the considered multiple mobile users and multiple MEC servers environment, the mobile users' assignment to MEC servers could be treated as a Markov Decision Process (MDP) over the long time horizon  $\mathcal{T}$ . In particular, at each time slot  $t \in \mathcal{T}$ , the DRL-SDN agent must associate each mobile user  $i \in \mathcal{U}$  with a MEC server  $s \in \mathcal{S}$ , in order the former one to perform hers computation offloading decision (Chapter 7.1.3). To this end, each mobile user  $i$  and the considered multiple MEC servers constitute the DRL-SDN agent's environment that interacts with, and whose dynamics are unknown. In the following, we formulate the mobile users association to MEC servers by adopting the MDP model, and we describe the corresponding meaning to each element in the MDP formulation, i.e.,  $(\mathcal{S}, \mathcal{A}, r, \rho, \gamma)$ , as these were introduced in Chapter 2.2, and

based on our considered environment. Additionally, in the following, the indication of the time slot, i.e.,  $t$ , is included in each notion that was introduced in Chapter 7.1.3.

**State space:** Towards providing an effective and personalized association of each mobile user  $i$  with a MEC server  $s$ , at each time slot  $t \in \mathcal{T}$ , each mobile user  $i$  is characterized by her state representation  $s_i^t$ . The latter is composed of mobile user's and MEC servers' related information. Specifically, at the time slot  $t \in \mathcal{T}$ , the mobile user's  $i$  state  $s_i^t \in \mathbb{S}$  includes the following information:

- mobile user's position vector:  $\mathbf{v}_i^t = (x_i^t, y_i^t)$ .
- mobile user's application's  $\mathcal{A}_i^t$  characteristics, completion latency and energy consumption weight preferences:

$$\mathbf{c}_i^t = (B_i^t, \phi_i^t, w_i^{l,t}, w_i^{e,t}), w_i^{l,t} + w_i^{e,t} = 1 \text{ (Chapter 7.1.3)}$$

- mobile user's risk-aware characteristics:  $\mathbf{z}_i^t = (\alpha_i^t, k_i^t)$ .
- mobile user's local device's characteristics:

$$\mathbf{d}_i^t = (dc_i^t, de_i^t, \bar{e}_i^t)$$

- mobile user's transmission power vector:

$$\mathbf{p}_i^t = (p_{i,1}^t, \dots, p_{i,s}^t, \dots, p_{i,S}^t), s \in \mathcal{S}$$

- MEC servers threshold amount of data vector:

$$\mathcal{B}^t = (\mathcal{B}_1^t, \dots, \mathcal{B}_s^t, \dots, \mathcal{B}_S^t), s \in \mathcal{S}$$

- MEC servers computing capability vector:

$$\bar{\mathbf{f}}^t = (\bar{f}_1^t, \dots, \bar{f}_s^t, \dots, \bar{f}_S^t), s \in \mathcal{S}$$

## Chapter 7. Future Work

Concretely, we denote the mobile user's  $i$  state representation at time slot  $t \in \mathcal{T}$ , as a vector  $s_i^t = (\mathbf{v}_i^t, \mathbf{c}_i^t, \mathbf{d}_i^t, \mathbf{p}_i^t, \mathcal{B}^t, \bar{\mathbf{f}}^t) \in \mathbb{S}$ .

**Action space:** In each time slot  $t \in \mathcal{T}$ , the mobile user  $i$  needs to be associated with a MEC server  $s \in \mathcal{S}$ , i.e.,  $a_i^t = s$ , based on hers current state  $s_i^t \in \mathbb{S}$ . As a result, the feasible action space of the DRL-SDN agent in each time slot is the set of MEC servers, thus  $\mathbb{A} \triangleq \mathcal{S} = \{1, \dots, s \dots, S\}$ .

**Reward:** In each time slot  $t \in \mathcal{T}$ , the DRL-SDN agent observes mobile user's  $i$  current state  $s_i^t$ , and associates the mobile user  $i$  with the MEC server  $s \in \mathcal{S}$ , thus it performs action  $a_i^t = s \in \mathbb{A}$ . After that, the DRL-SDN agent receives a reward  $r(s_i^t, a_i^t)$  for associating mobile user  $i$ , whose state is  $s_i^t$ , to the MEC server  $a_i^t \in \mathbb{A}$ . In our model, with the objective of minimizing both the mobile user's overall experienced completion latency for hers applications, i.e.,  $\mathcal{A}_i^t, \forall t \in \mathcal{T}$ , and device's overall energy consumption over the long time horizon  $\mathcal{T}$ , we define reward  $r(s_i^t, a_i^t)$  as follows:

$$r(s_i^t, a_i^t) = -\mathbb{O}_i^t \quad (7.17)$$

, where  $\mathbb{O}_i^t$  is the mobile user's  $i$  actual overall overhead that perceives after performing hers computation offloading decision, as this is defined by Algorithm 2 in Chapter 3.5.3, to the MEC server  $a_i^t$ . As a result, based on Eq. 7.11, 7.12 and our discussion in Chapter 7.1.3, the mobile user's  $i$  experienced actual overhead at time slot  $t \in \mathcal{T}$  is defined based on if MEC server  $a_i^t$  fails to process mobile user's  $i$  offloaded data  $b_{i,s}^t$  (determined by Algorithm 2 - Chapter 3.5.3), and is given as follows:

$$\mathbb{O}_i^t = \begin{cases} O_i^t(b_{i,s}^t, \bar{b}_s^t) & , \text{ if not failure} \\ q_{i,0}^t + \frac{b_{i,s}^t}{R_{i,s}^t} \left( \frac{w_i^{l,t}}{t} + \frac{w_i^{e,t}}{e_i} p_{i,s}^t \right) & , \text{ if failure} \end{cases} \quad (7.18)$$

**Transition probability:** Given DRL-SDN agent's action  $a_i^t$  at time slot  $t \in \mathcal{T}$ , the transition probability  $\rho(s_i^{t+1}|s_i^t, a_i^t)$  represents the probability that the mobile user's  $i$  state is transformed to  $s_i^{t+1}$  from  $s_i^t$  in the next time slot.

According to the MDP model given above, the mobile user's association with a MEC server can be formulated as a problem of finding the optimal association policy that maximizes the long-term discounted cumulative reward (Eq. 2.15). Specifically, the definition of the association policy obtained by the DRL-SDN agent is given as follows:

**Definition 9** *An association policy  $\pi_i : \mathbb{S} \rightarrow \mathbb{A}$  is defined as a mapping from state  $s_i^t$  to action  $a_i^t$ , i.e.,  $\pi(s_i^t) = a_i^t$ . Specifically, the DRL-SDN agent determines an association action  $a_i^t \in \mathbb{A}$  for the mobile user  $i$  according to policy  $\pi_i$ , and given mobile user's  $i$  observed environment state  $s_i^t$ .*

To this end, DRL-SDN agent aims to determine an optimal association policy  $\pi_i^*$  that maximizes the long term discounted cumulative reward (cumulative perceived completion latency and energy consumption) obtained over the long time horizon  $\mathcal{T}$ , i.e.,

$$\pi_i^* = \arg \max_{\pi_i} \mathbb{E} \left[ \sum_{t=0}^{\mathcal{T}} \gamma^t r(s_i^t, a_i^t) \right] = \arg \max_{\pi_i} \mathbb{E} \left[ \sum_{t=0}^{\mathcal{T}} -\gamma^t \mathbb{O}_i^t \right] \quad (7.19)$$

### 7.1.5 Full-Fledged Risk-aware Orchestration

The DRL-SDN agent learns an optimal association policy  $\pi_i^*$  that maximizes mobile user's  $i$  perceived cumulative completion latency and energy consumption over the considered set of timeslots  $\mathcal{T}$ , by learning and approximating the optimal Q-values ( $Q^*$ ) through its DQN, as it is presented in Fig. 7.1. At each timeslot  $t \in \mathcal{T}$ , each mobile user's  $i \in \mathcal{U}$  association to a MEC server  $s \in \mathcal{S}$  is performed through the DRL-SDN agent (first stage). In particular, the DRL-SDN agent receives the mobile user's  $i$  characteristics, and by combining them with the MEC servers' characteristics,

it constructs the mobile user's  $i$  state representation  $s_i^t$  (Chapter 7.1.4). As a result, the DRL-SDN agent, evaluates each action's Q-value through its DQN (forward propagation), and a MEC server, i.e.,  $a_i^t$  is selected through its  $\epsilon$ -greedy policy.

Therefore, after each mobile user's  $i$  association to a MEC server  $s$ , the computation offloading (second stage) takes place independently (in parallel) for each cluster of MEC server  $s$  and associated mobile users  $\mathcal{U}_s$ , and through the proposed decision-making framework, i.e., Algorithm 2 in Chapter 3.5.3). Each mobile user  $i$  performs its computation offloading  $b_{i,s}^t$ , and perceives an actual overhead  $\mathbb{O}_i^t$ , as per Eq. 7.18. The latter is used as the DRL-SDN agent's associated reward  $r(s_i^t, a_i^t) = -\mathbb{O}_i^t$  for performing action  $a_i^t$ , thus it is reported back, and it stored to the DRL-SDN agent's replay memory buffer  $\mathcal{D}$  (Chapter 2.2.2). At each learning iteration  $i$  the DRL-SDN agent randomly samples a batch of the form:

$$B = \{(s_1^t, a_1^t, s_1^{t+1}, r(s_1^t, a_1^t)), \dots, (s_M^t, a_M^t, s_M^{t+1}, r(s_M^t, a_M^t))\}$$

where  $M$  is the batch size, and the *Temporal Difference Loss* function is evaluated as per Eq. 2.28. The latter's gradient, which is defined as follows:

$$\nabla_{\theta_i} \mathcal{L}_i(\theta_i) = \mathbb{E}_{\mathcal{D}} [(y_i - Q(s_i^t, a_i^t; \theta)) \nabla_{\theta_i} Q(s_i^t, a_i^t; \theta_i)] \quad (7.20)$$

is used for updating of DQN's parameters  $\theta_i$  through backward propagation (gradient descent). Finally, the target DQN's parameters  $\theta_i^-$  are updated periodically with the DQN's parameters  $\theta_i$ . Our full-fledged proposed framework is presented bellow.

### 7.1.6 Empirical Evaluation

In this Section, a detailed numerical evaluation is presented to study the performance and the inherent attributes of the proposed DRL-SDN enabled and risk-aware based framework regarding mobile users' computation offloading and association with MEC servers. We consider multiple mobile users and multiple MEC servers environment, and we study the performance of the proposed framework by focusing on mobile users'

---

**Algorithm 8 :** Association & Computation Offloading

---

**Input:** Mobile users  $\mathcal{U}$ , MEC servers  $\mathcal{S}$

---

**for**  $t = 1$  to  $T$  **do**

Association (DRL-SDN):

**for**  $i = 1$  to  $U$  **do**

            Evaluate  $Q(s_i^t, a_i^t; \theta)$ ,  $\forall a_i^t \in \mathbb{A}$ .

            Choose action  $a_i^t$  ( $\epsilon$ -greedy policy)

**end for**

Computation Offloading (Mobile users):

**for**  $s = 1$  to  $S$  (In parallel) **do**

            Algorithm 2 in Chapter 3.5.3

**end for**

Learning (DRL-SDN):

**for** each learning iteration  $i$  **do**

            Sample batch  $B$

            Estimate  $\mathcal{L}_i(\theta_i)$  (Eq. 2.28)

            Update  $\theta_i$  through gradient descent steps of  $\nabla_{\theta_i} \mathcal{L}_i(\theta_i)$

**if**  $i \% \tau == 0$  ( $\tau$  updating period) **then**

$\theta_i^- = \theta_i$

**end if**

**end for**

**end for**

---

association to MEC servers. Regarding mobile users' computation offloading through the distributed decision-making framework (Algorithm 2 - Chapter 3.5.3), indicative results are presented in Chapter 3.5.4. In particular, our proposed mobile users' association with the MEC servers through the DRL-SDN agent, which trains in an online fashion its DQN towards determining an optimal association policy (Eq. 7.19)



## Chapter 7. Future Work

is evaluated concerning mobile users' perceived cumulative completion latency and energy consumption over the long time horizon  $\mathcal{T}$ . Specifically, we demonstrate the DQN's performance for different neural network architectures, learning parameters, while we present a scalability analysis for an increasing number of users and MEC servers. Furthermore, we present the DRL-SDN agent's capability to learn and determine an optimal association policy in the realistic case of mobile users' dynamic entrance and exit. Finally, a comparative evaluation of the DQN is presented with respect to different neural network designs, which have demonstrated state-of-the-art performance in the recent RL research literature.

We consider an  $1000 \times 1000$  grid of multiple mobile users and multiple MEC servers network, where our base simulation scenario consists of servicing  $U = 200$  mobile users via a set of  $S = 3$  MEC servers, where the latter ones are randomly distributed in the considered grid. We investigate the problem of mobile users computation offloading and association to MEC servers over 20 time slots, i.e.,  $T = 20$ , that could be considered as different time instances over a particular time window, e.g., day, week. Towards constructing a realistic and stochastic dynamic environment all mobile user's and MEC servers' characteristics, which are included to mobile user's state representation  $s_i^t$  as well, are drawn from distinct multivariate Gaussian distributions with random means and covariance matrices, which are defined per mobile user  $i \in \mathcal{U}$ , and per time slot  $t \in \mathcal{T}$ . In other words, mobile user's  $i$  position vector  $\mathbf{v}_i^t$  at time slot  $t$  is drawn from a multivariate Gaussian distribution of two dimensions, i.e.,  $\mathbf{v}_i^t = (x_i^t, y_i^t)$ , thus  $\mathbf{v}_i^t \sim N(\mu_{i,v}^t, \Sigma_{i,v}^t)$ , with random mean  $\mu_{i,v}^t$  and covariance matrix  $\Sigma_{i,v}^t$ . As the indexing indicates, the latter multivariate Gaussian distribution is defined per mobile user's  $i$  position vector, i.e.,  $(\mu_{i,v}^t, \Sigma_{i,v}^t) \neq (\mu_{j,v}^t, \Sigma_{j,v}^t)$  for  $i \neq j$  and per time slot  $t$ , i.e.,  $(\mu_{i,v}^t, \Sigma_{i,v}^t) \neq (\mu_{i,v}^{t'}, \Sigma_{i,v}^{t'})$  for  $t \neq t'$ .

Equivalently, the rest of each mobile user's and MEC servers' characteristics are drawn as follows:

$$(B_i^t, \phi_i^t) \sim N(\mu_{i,\mathcal{A}}^t, \Sigma_{i,\mathcal{A}}^t)$$

## Chapter 7. Future Work

$$(a_i^t, k_i^t) \sim N(\mu_{i,r}^t, \Sigma_{i,r}^t)$$

$$(\bar{e}_i^0, de_i^0, dc_i^0) \sim N(\mu_{i,d}, \Sigma_{i,d})$$

$$(\mathcal{B}_s^t, \bar{f}_s^t) \sim N(\mu_s^t, \Sigma_s^t)$$

$\forall t \in \mathcal{T}, \forall i \in \mathcal{U}, \forall s \in \mathcal{S}$ . Moreover, regarding mobile user's  $i$  local device's energy consumption ( $de_i^t$ ), and computing capability ( $dc_i^t$ ), are kept constant over the time slots, thus  $de_i^t = de_i^0, dc_i^t = dc_i^0, \forall t \in \mathcal{T}, \forall i \in \mathcal{U}$ . On the other hand, each mobile user's local device's energy availability  $\bar{e}_i^t$  at time slot  $t$  is updated dynamically by subtracting from the previous remaining energy availability  $\bar{e}_i^{(t-1)}$ , the mobile user's  $i$  overall consumed energy for performing hers computation offloading  $b_{i,s}^t$  (if any), and for executing locally the remaining part of hers application ( $B_i^t - b_{i,s}^t$ ) (if any). Furthermore, regarding each mobile user's  $i$  completion latency and energy consumption weight preferences, i.e.,  $(w_i^{l,t}, w_i^{e,t})$ , it holds true that  $w_i^{l,0} = w_i^{e,0} = 0.5, \forall i \in \mathcal{U}$ . However,  $\forall t \in \mathcal{T}, t > 0$ , the mobile user's  $i$  completion latency weight preference is updated as  $w_i^{l,t} = \frac{\bar{e}_i^t}{\bar{e}_i^0} \cdot w_i^{l,0}$ , where the latter physical meaning indicates that the mobile user's  $i$  completion latency weight preference is a decreasing function with respect to the time slots, as the remaining energy availability  $\bar{e}_i^t$  is being decreased over time as well, thus the mobile user  $i$  shifts hers attention to hers device's energy consumption as the time pass, since  $w_i^{e,t} = 1 - w_i^{l,t}$ .

For all the aforementioned drawn values, as per [231, 230], we make sure that  $B_i^t \in [1000, 5000]\text{KB}$ ,  $B_i^t \cdot \phi_i^t \in [1000, 5000]$  Mega-Cycles,  $dc_i^0 \in [0.1, 1] \cdot 10^9 \frac{\text{Cpu Cycles}}{\text{sec}}$ ,  $de_i^0 \in [0.1, 1] \cdot 10^{-9} \frac{\text{J}}{\text{Cpu Cycle}}, \forall i \in \mathcal{U}, \forall t \in \mathcal{T}$ . Furthermore,  $\mathcal{B}_s^t \in [30, 70]\% \cdot \sum_{i \in \mathcal{U}} B_i^t, \forall s \in \mathcal{S}, \forall t \in \mathcal{T}$ . Each MEC server's  $s \in \mathcal{S}$  channel bandwidth  $W_s = 15\text{MHz}$  (Eq. 7.1), and each mobile user's transmission power  $p_{i,s}^t$  at timeslot  $t$  is defined as  $p_{i,s}^t = \frac{d_{i,s}^t}{R_s^2}$ , where  $d_{i,s}^t$  is the mobile user's  $i$  distance from MEC server  $s$  at timeslot  $t$ , and  $R_s$  is the MEC server's  $s$  coverage area, which here is set to  $R_s = 250\text{m}, \forall s \in \mathcal{S}$ . Finally, each mobile user's channel gain  $g_{i,s}^t$  with the MEC server  $s$  is modeled as  $g_{i,s}^t = \frac{1}{(d_{i,s}^t)^\theta}$ , where  $\theta = 3$  is the distance loss exponent.

## Chapter 7. Future Work

Furthermore, the online training of the DRL-SDN agent occurs over a number of episodes, where each episode consists of the considered 20 timeslots, i.e.,  $\mathcal{T} = \{1, \dots, 20\}$ . The DRL-SDN agent learns the optimal Q-values, by randomly sampling batches of the form  $(s_i^t, a_i^t, r(s_i^t, a_i^t), s_i^{t+1})$ , and through gradient steps towards minimizing the *Temporal Difference Loss* function (Eq. 2.28). We adopt the Adam optimizer [328] as the optimization method for updating the parameters of the DQN. In the following figures, unless otherwise explicitly stated, we demonstrate the average cumulative reward, i.e.,  $\frac{\sum_{t=0}^{T=20} -\gamma^t \mathbb{Q}_i^t}{U}$  for the base scenario of simulation with  $U = 200$  mobile users, and  $S = 3$  MEC servers. The DRL-SDN agent associates each mobile user  $i \in \mathcal{U}$  to a MEC server  $s \in \mathcal{S}$  through an  $\epsilon$ -greedy policy [62], as follows:

$$\alpha_i^t = \begin{cases} \arg \max_{a_i^t \in \mathcal{A}} Q(s_i^t, a_i^t) & , \text{ with prob. } 1 - \epsilon^t \\ \text{random} & , \text{ with prob. } \epsilon^t \end{cases} \quad (7.21)$$

where  $a_i^t \in \mathcal{S}$  is the MEC server with which the mobile user  $i$  is associated for the timeslot  $t \in \mathcal{T}$ , given the mobile user's  $i$  corresponding state representation  $s_i^t$  (Chapter 7.1.4).

To this end, the DRL-SDN agent through its  $\epsilon$ -greedy policy (Eq. 7.21), it maintains a trade-off between exploration and exploitation. In particular, at the beginning of the training, a high value, i.e.,  $\epsilon^0 = 1$ , indicates a high exploration for the DRL-SDN agent, as the latter is needed in order the DRL-SDN agent to receive sufficient feedback for all the different options of associations of a mobile user to a MEC server, and learn the unknown environment. On the other hand, as the training evolves, the  $\epsilon$  value is decreasing with respect to the episodes, i.e.,  $\epsilon^t = \epsilon_{decay} \cdot \epsilon^{t-1}$ , where  $\epsilon_{decay} < 1$ , as the DRL-SDN agent is tending to trust more its learned Q-values (exploitation).

**Pure Operation** Fig. 7.2a & 7.2b demonstrate the DRL-SDN agent's learning as a function of the episodes, and concerning different sizes of sampling batches, and a different number of hidden units on each layer of the DQN. First of all, as Fig. 7.2a

## Chapter 7. Future Work

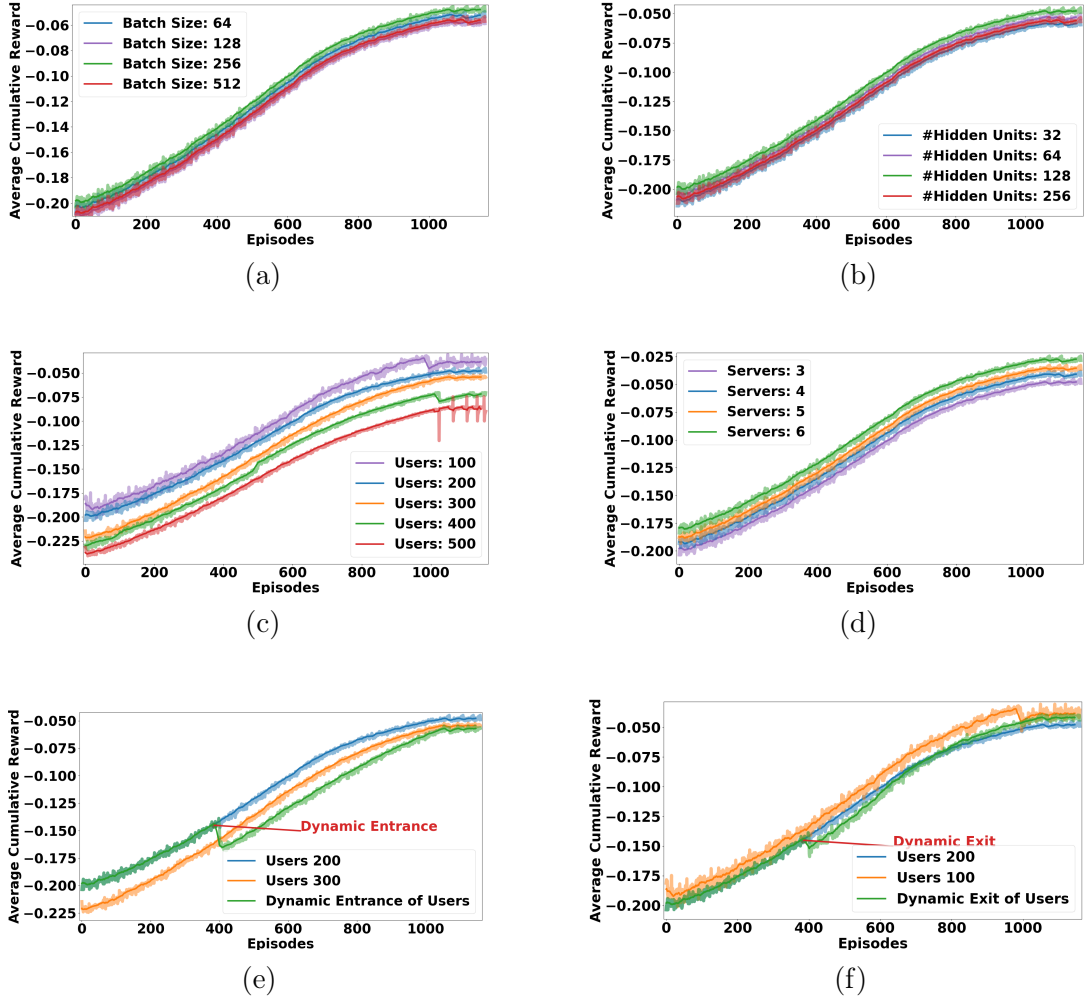


Figure 7.2: Mobile users' association to MEC servers: DRL-SDN agent's online learning

& 7.2b indicate, the average cumulative reward is increasing throughout DRL-SDN agent's learning, and almost after 1000 episodes, the DRL-SDN agent establishes an association policy, through which the mobile users can experience almost a fivefold decrease in their perceived cumulative completion latency and energy consumption over the considered 20 timeslots. Furthermore, the DRL-SDN agent's learning stabilizes after a certain point of episodes, as the DRL-SDN agent's  $\epsilon$ -greedy policy is

essentially becoming a deterministic policy concerning the learned Q-values, thus the DRL-SDN agent at each timeslot  $t \in \mathcal{T}$  associates each mobile user  $i \in \mathcal{U}_s$  with the MEC server  $s \in \mathcal{S}$  that provides the highest Q-value (Eq. 7.21). Moreover, Fig. 7.2a, indicates that DRL-SDN agent's learning is more effective with sampling batches of the size of 256, as a higher cumulative reward is achieved. Similarly, Fig. 7.2b, shows that a greater number of hidden units on each layer of the DQN does not always lead to more efficient learning, as the DRL-SDN agent is overfitting concerning its learned Q-values, a well-common phenomenon for high capacity neural networks. It is worth to be mentioned, that all combinations of batch sizes and a number of hidden units were investigated towards tuning the DRL-SDN agent, while here for practical purposes we demonstrate individually the cases of the batch size and the number of hidden units. The tuning of the DRL-SDN agent, indicated the best combination of batch size and a number of hidden units to be 256 and 128, respectively. Finally, we did not need to perform tuning concerning the learning parameter of the DRL-SDN agent per gradient step, as the Adam optimizer [328] adjusts its learning parameter dynamically on each step. Moreover, it is worth to be noted that the rest presented figures correspond to the best combination of batch size and a number of hidden units.

**Scalability Analysis:** Fig 7.2c & 7.2d demonstrate the average cumulative reward of DRL-SDN agent's association of mobile users to MEC servers, as a function of the episodes, and concerning a different number of mobile users and MEC servers. It is worth to be mentioned that due to the DRL-SDN agent's  $\epsilon$ -greedy policy (Eq. 7.21), regardless of the complexity of the considered environment, e.g., number of mobile users or MEC servers, we expect the DRL-SDN agent to exploit its learned Q-values after a certain period, as its association policy becomes deterministic (first branch in Eq. 7.21). Indeed, the latter is confirmed from Fig 7.2c and 7.2d, as regardless of the number of mobile users and MEC servers, the average cumulative reward stabilizes as the DRL-SDN agent's training evolves. Moreover, as the number of mobile

users increases the latter's association to MEC servers becomes more complex, as the competition for each MEC server's computing resources is increasing as well, and as a result is more difficult for the DRL-SDN agent to establish an optimal policy in that case. However, as Fig 7.2c shows the DRL-SDN agent can learn an association policy after almost 1000 episodes, and even in the extreme case of 500 mobile users, the latter experience almost a two-fold decrease in their perceived cumulative completion latency and energy consumption over the considered number of timeslots. On the other hand, Fig 7.2d shows that the average cumulative reward increases, as the number of MEC servers increases. In other words, the DRL-SDN agent can efficiently identify and utilize the increased number of MEC servers, and determine an optimal association policy such that the mobile users' perceived completion latency and energy consumption decreases as the number of MEC servers increases.

**Dynamic Entrance and Exit:** Multiple mobile users and MEC servers environment are challenging mainly due to their stochastic nature. In the above evaluation, we have shown how the DRL-SDN agent can train and utilize its DQN to establish an optimal policy for associating the mobile users to the appropriate MEC servers, such that the former's perceived cumulative completion latency and the energy consumption is minimized over timeslots. The latter demonstrates the ability of the DRL-SDN agent to learn the stochastic unknown environment, where all mobile users' and MEC servers' characteristics per timeslot are changing dynamically. However, in a realistic environment, considering each MEC server's coverage area, new mobile users enter the considered environment, or/and mobile users exit the environment, as they are located inside/beyond the coverage area of the MEC servers. We call the latter dynamic scenarios, as dynamic entrance and exit of mobile users, respectively. Here, towards investigating the DRL-SDN agent's ability to adapt its learning for the described dynamic scenarios, the latter is performed at a particular point of time during the DRL-SDN agent's online training. In particular, regarding the new mo-

mobile users' dynamic entrance, a point of time is selected, and new mobile users that have not been seen before by the DRL-SDN agent, are considered in the association and computation offloading. Similarly, regarding the mobile users' dynamic exit, a point of time is selected, and several mobile users are assumed to be located beyond the coverage of each MEC server, thus are not yet considered in the association and computation offloading.

Fig. 7.2e demonstrates the DRL-SDN agent's perceived average cumulative reward as a function of the episodes, and by considering the case of mobile users' dynamic entrance. Initially, the base simulation scenario is considered, thus a set of  $U = 200$  mobile users who are being served by a set of  $S = 3$  MEC servers. However, as Fig. 7.2 shows, a particular episode is selected, in which a set of 100 new mobile users enter the considered environment. The notion of new mobile users refers to the fact that the latter's characteristics over the considered number of timeslots e.g., mobility, applications' demands, risk-aware characteristics, etc, have not been seen before by the DRL-SDN agent. As a result, as we in Fig. 7.2e, at the time of new mobile users' entrance, the DRL-SDN agent's achieved average cumulative reward is decreasing, as the association policy that was established until this point, is no longer effective for the new considered environment, where 100 new mobile users are included in the association and computation offloading as well. However, as the DRL-SDN agent's training evolves, the latter can adapt its learned association policy, and the average mobile users' experienced cumulative completion latency and the energy consumption is decreasing over the rest episodes. Furthermore, Fig. 7.2 indicates the average cumulative reward as a function of the episodes for the case of  $U = 200$  (blue line), and the case of  $U = 300$  mobile users (orange line), where the latter one corresponds to the case where the DRL-SDN agent starts to learn an optimal association policy, and all  $U = 300$  mobile users are considered from the beginning, in contrast with the dynamic entrance case, where the DRL-SDN agent starts to learn an optimal association policy with the consideration of only  $U = 200$

mobile users, and the rest 100 mobile users enter the considered environment later, as new mobile users. As we see, in the case of mobile users' dynamic entrance (green line) the DRL-SDN agent at the end, can establish an association policy, such that its perceived average cumulative reward is lower than the case of  $U = 200$  mobile users (blue line), due to the higher completion of each MEC server's computing resources, and close to the corresponding value of the case where the new entered 100 mobile users, were considered from the beginning of the DRL-SDN agent's training, thus the case of  $U = 300$  mobile users totally (orange line). As a result, the DRL-SDN agent can adapt its learning for the case of mobile users' dynamic entrance, and determine an effective final association policy.

Fig. 7.2f presents the DRL-SDN agent's perceived average cumulative reward as a function of the episodes, and by considering the case of mobile users' dynamic exit. In particular, the DRL-SDN agent starts to learn an association policy with the consideration of the base simulation scenario, thus  $U = 200$  mobile users, and  $S = 3$  MEC servers. However, as the DRL-SDN agent's training evolves, a particular episode is selected, and 100 mobile users who are randomly selected, are assumed to be located beyond the coverage area of each MEC server. As a result, the latter mobile users are not yet considered in the association and computation offloading for the rest of the episodes. As Fig. 7.2f indicates, at the point of mobile users' dynamic exit, the DRL-SDN agent's average cumulative reward is decreasing, as its currently learned policy, which corresponds for the case of  $U = 200$  mobile users, is not yet effective for the new case of  $U = 100$  mobile users. However, the DRL-SDN agent in the case of mobile users' dynamic exit (green line) can adapt its association policy and effectively address the considered dynamic scenario, as its final learned association policy achieves a higher average cumulative reward compared to the case where  $U = 200$  mobile users are considered until the end (blue line), and slightly lower than DRL-SDN agent's perceived value in the case where only the corresponding remaining  $U = 100$  mobile users in the dynamic scenario of exit, are considered



from the beginning and until the end of DRL-SDN agent's training (orange line).

**Comparative Evaluation:** Here, we compare the performance of the considered DQN against alternative DRL approaches, which are commonly considered as improvements of DQN in the RL research literature. In particular, we consider the cases of Double DQN (DDQN), Dueling DQN (DuQN), and Dueling Double DQN (D3QN). The idea of DDQN [329] is to reduce over-estimations by decomposing the max operation in the target  $y_i$  of the *Temporal Difference Loss* function (Eq. 2.28), into action selection and action evaluation. Specifically, at each learning iteration  $i$ , the  $y_i$  in Eq. 2.28 is replaced with:

$$y_i^{\text{double}} = \mathbb{E}_{s_{t+1} \sim D} \left[ r(s_t, a_t) + \gamma Q(s_{t+1}, \arg \max_{a_{t+1}} Q(s_{t+1}, a_{t+1}; \theta_i); \theta_i^-) \right]$$

, thus the selection of the best action  $a_{t+1}$  is performed through the DQN, and its evaluation (Q-value) is performed through the target network with parameters  $\theta_i^-$ . On the other hand in Dueling DQN (DuQN) [330] an alternative neural network architecture is introduced, mainly motivated from the fact that is not always necessary for the RL agent to evaluate each action for a given state. In the considered problem of the mobile users' association to the MEC servers, the latter idea is equivalent to the observation that given a mobile user's  $i$  state representation  $s_i^t$  at timeslot  $t$ , it may be the case that the mobile user's perceived cumulative completion latency and energy consumption after that state, is not highly affected by the DRL-SDN agent's chosen action, i.e., MEC server.

For example, consider the case where at timeslot  $t$  the mobile user  $i$  desires to execute an application  $\mathcal{A}_i^t$  with relative small size of input data  $B_i^t$ . In that case, is probable more beneficial for the mobile user  $i$  to execute its application locally at hers device, and avoid the additional transmission overhead. As a result, in that case, it is not necessary for the DRL-SDN agent to examine and explore all different association options, i.e., actions, as regardless the MEC server  $s$  that the mobile user  $i$  will be associated with, the latter will not perform any computation

offloading. Generally, in DuDQN the DQN's output layer is decomposed in two different streams. The first one approximates the optimal value function of a state  $s_t$ , i.e.,  $V^*(s_t; \theta)$ , and the second one approximates the optimal advantage function for performing action  $a_t$  at the given state  $s_t$ , i.e.,  $A^*(s_t, a_t; \theta)$ . In particular, given a policy  $\pi$  the value function of the state  $s_t$  is given as  $V^\pi(s_t) = \mathbb{E}_{a_t \sim \pi(s_t)}[Q^\pi(s_t, a_t)]$ , and the advantage function is defined as  $A^\pi(s_t, a_t) = Q^\pi(s_t, a_t) - V^\pi(s_t)$ . As a result, the DuDQN's parameters are updated via gradient steps of the *Temporal Difference Loss* function, where the only difference with the regular DQN, is that each Q-value is estimated through the aggregation of the corresponding value and advantage function, i.e.,  $Q(s_t, a_t; \theta) = V(s_t; \theta) + A(s_t, a_t; \theta)$ . Finally, D3QN [331] is considered as the combination of the ideas of Double DQN (DDQN) and Dueling DQN (DuQN).

The comparative evaluation is performed by considering the base simulation scenario of  $U = 200$  mobile users and  $S = 3$  MEC servers, and concerning the perceived average cumulative reward after 1000 episodes of training, alongside with the perceived average cumulative reward on  $U = 200$  new mobile users that have not been seen by the DRL-SDN agent during its training. In other words, after the DRL-SDN agent is trained in an online fashion for the considered  $U = 200$  mobile users and 1000 episodes, then its perceived average cumulative reward is evaluated on  $U = 200$  new mobile users. The latter is estimated as the average value over 1000 executions of 20 timeslots, where we make sure that the mobile users' characteristics are drawn from different multivariate Gaussian distributions compared to the ones that were used for the DRL-SDN agent's training. As a result, the perceived average cumulative reward for the  $U = 200$  new (unseen) mobile users, constitutes a generalization measurement for each DRL approach.

Table 7.1 summarizes the corresponding results. It is worth to be noted that for each of the above DRL approaches, the best values of the free parameters, i.e., batch size, number of hidden units, were used, and were identified through a common tun-

Table 7.1: Comparative evaluation of different DRL approaches

	Avg. Cum. Reward after 1000 Episodes	Avg. Cum. Reward for 200 Unseen Users	Difference (%)
DQN	-0.047	-0.051	8.5%
DDQN	-0.045	-0.048	6.6%
DuQN	-0.041	-0.044	7.3%
D3QN	-0.037	-0.039	5.4%

ing process concerning the perceived average cumulative reward on the 200 unseen mobile users. For example, for the DQN approach, the best values of batch size and a number of hidden units were identified as 256, and 128, respectively, thus similarly as in the Pure Operation analysis above. As Table 7.1 indicates, the presented DRL approaches can effectively generalize. Specifically, the difference between the DRL-SDN agent’s perceived average cumulative reward after 1000 episodes, and the corresponding value for the case where the DRL-SDN agent’s learning association policy is used for  $U = 200$  new mobile users is less than 9%. Moreover, Double DQN (DDQN) and Dueling DQN (DuQN) can improve the regular setting of DQN in both the perceived average cumulative reward during the online training and the DRL-SDN agent’s ability to generalize on new mobile users. DuQN demonstrates the highest improvement, mainly because can efficiently explore the most promising areas of the state-action space. Finally, the combination of DDQN and DuQN constitutes the best DRL approach, where the DRL-SDN agent learns the best association policy for the considered  $U = 200$  mobile users, which generalizes effectively (5.4% difference) in the case of  $U = 200$  unseen mobile users as well.

### 7.1.7 Summary

The joint problem of mobile users’ association to the MEC servers, and mobile users’ computation offloading is decoupled and studied through a two-stage Deep Reinforcement Learning (DRL) risk-aware approach. A centralized DRL-SDN controller

that acts as a DRL agent is considered and performs the mobile users' association to the MEC servers. The latter is formulated as an MDP over the long-time horizon, and the DRL-SDN agent seeks an optimal association policy that minimizes the mobile user's perceived cumulative completion latency and energy consumption. Towards addressing the high-dimensional, unknown and stochastic environment, the DRL-SDN agent maintains and trains its DQN in an online fashion, while it associates each mobile user to a MEC server through its  $\epsilon$ -greedy policy. The mobile users' computation offloading to the corresponding MEC server is formulated as a non-cooperative game under the consideration of each MEC server's computing uncertainty, and a distributed decision-making framework is adopted (Chapter 3.5.3) for converging to the unique Nash Equilibrium point. The proposed model and approach consider the mobile user's risk-aware characteristics in both the association and the computation offloading. The latter comes in contrast to the majority of the existing methods in the literature, where the mobile users are assumed risk-neutral. Furthermore, the mobile users' association through a centralized DRL-SDN agent enables better utilization of the power of neural networks, and an additional generalization dimension over the mobile users. Despite that, the centralized DRL-SDN agent overcomes the commonly made assumption by the research literature, where each mobile user's device is assumed capable of performing complex learning approaches, which does not always hold due to the latter's limited energy availability, and computing capability. Detailed numerical results were presented highlighting the performance benefits of our proposed approach, and the DRL-SDN agent's ability to learn a complex stochastic environment, and effectively adapt its learned association policy for the mobile users' dynamic entrance and exit scenarios.

## 7.2 Future Directions

Our introduced decision-making frameworks utilize Reinforcement Learning (RL) towards determining optimal policies in unknown and stochastic environments, which are characterized by partial or incomplete information. In this thesis, we primarily leveraged RL in its online fashion, thus the learning of the RL agents (computing systems) is performed online during their actual decision-making process. The process of RL involves iteratively collecting experiences by interacting with the environment, typically with the latest learned policy, and then using that experience to improve the policy (Chapter 2.2).

In many settings, the latter is impractical either because data collection is expensive, e.g., education or healthcare decision-making computing systems, and/or dangerous, e.g., autonomous driving, healthcare. In our studied application domains, the collection of experiences with intermediate non-optimal policies could harm the end-user experience in a variety of different ways. For that reason, we strongly believe that Reinforcement Learning and our introduced decision-making frameworks could be studied from an offline perspective as well, thus without iterative online interaction with the real-world environment. The latter is commonly called *Offline Reinforcement Learning* [332], and has recently started attracting tremendous research attention [333, 334].

In particular, by leveraging the information of past experiences and interactions that could have been generated by previously developed and deployed decision-making frameworks, *Offline Reinforcement Learning* could utilize only these logged experiences to learn an optimal policy through *off-policy learning*. Q-learning and its deep learning variant DQN (Chapter 2.2) constitute indicative examples of *off-policy learning* methods, since for the learning of the optimal Q-values, there is no need of knowing the actual policy that was used to select these experiences, thus in Equations 2.27 & 2.28 the Q-value of the best action is used as a target value.

## *Chapter 7. Future Work*

To this end, our decision-making frameworks could be extended in a way such that the learning of the optimal policies, e.g., data offloading or/and association with MEC servers, to happen offline by using logged interactions that were generated by previously applied decision-making frameworks by different and/or the same mobile users. The latter would hopefully lead the decision-making agents to establish an optimal policy offline, and then apply this policy to the real-world environment, e.g., multiple MEC servers environment, and better improve it through the new observed interactions. The offline learning of an initial optimal policy in contrast with the regular online RL could avoid low-quality actions to be performed in particular states and time instances that could harm the end-users perceived experience.

Furthermore, in this thesis, due to the actual nature of the studied applications domains, the inherited uncertainties of the shared resources were investigated under probabilistic models that follow certain assumptions (Chapters 3.5, 3.6, and 5.2). We are strongly suggesting the theoretical investigation of our introduced theory and decision-making frameworks under alternative probabilistic models that could be more powerful and/or realistic on different application domains. The theoretical analysis of the notions of convergence, existence, and uniqueness of Equilibrium points, as these were provided in this thesis, would be highly important under these alternative probabilistic models.

Finally, in this thesis we were interested in non-cooperative games, thus games where the decision-makers (computing systems) compete with each other for the shared resources. It would be really valuable the investigation and extension of our proposed theory and decision-making frameworks under the principles of cooperative games, which correspond to a different sub-field of Game Theory, as the decision-makers are assumed collaborative instead of rational (non-cooperative games).

# Chapter 8

## Conclusions

Autonomous intelligent decision-making in resource-constrained computing systems is complicated by the limited computing capabilities and competition for shared resources. The latter's' inherited uncertainties constitute additional challenges that need to be addressed and properly considered in the design of decision-making frameworks for resource-constrained computing systems. In this thesis, we address these problems.

First, we model the competition of the computing systems for the shared resources through Game Theory. The consideration of non-cooperative games allows us to develop distributed decision-making frameworks that converge to Equilibrium points, and do not ultimately depend on centralized and single-point of failure entities. The introduction of more efficient Equilibrium points that are defined on the idea of satisfying the computing systems' requirements towards providing personalized services, enables a realistic formalization of the decision-making process and reduces the complexity of Equilibrium optimal solutions.

Second, the incorporation of Reinforcement Learning addresses the computing systems' challenge of performing decisions autonomously in stochastic environments, and learn from their own past experiences. Furthermore, the benefits of neural networks and Deep Reinforcement Learning contribute significantly to the development

## *Chapter 8. Conclusions*

of decision-making frameworks for complex high dimensional environments, where regular techniques lack generalization and effectiveness.

Third, the modeling and incorporation of human awareness in the decision-making process of the computing systems deal with the introduced uncertainties by the shared resources, which could be over-exploited. Prospect Theory constitutes a behavioral model that describes human’s decision-making under uncertainty and risk, through the consideration of risk-aware characteristics. To this end, Prospect Theory allows us to design decision-making frameworks, based on which the computing systems mimic their owner’s behavior concerning risk, and thus optimize their owners’ personalized goals. We detail how Prospect Theory can be incorporated in the regular Reinforcement Learning setting through reward reshaping and therefore provide practical risk-aware decision-making frameworks.

The work from this thesis established methodological practice and introduced theoretical foundations for the design and application of autonomous and risk-aware decision-making frameworks for computing systems with constrained resources. Our introduced approaches were theoretically analyzed and empirically evaluated concerning their convergence properties, operation, scaling capabilities, and overall performance compared to alternative approaches. Although the aforementioned analysis was performed under the prism of particular challenging applications, the proposed theory, and approaches apply to a much broader domain.

Ultimately, our work and suggested future directions will allow for more research intersecting Game Theory, Reinforcement Learning, and Prospect Theory towards designing efficient decision-making frameworks for resource-constrained computing systems that can sense their owners’ behavioral patterns and make decisions autonomously and intelligently. We believe that personalized services, and human awareness, are notions strongly connected, and this thesis proposed theory and practical approaches towards this direction and based on these principles.



# References

- [1] Sarah Bankins and Paul Formosa. When ai meets pc: exploring the implications of workplace social robots and a human-robot psychological contract. *European Journal of Work and Organizational Psychology*, 29(2):215–229, 2020.
- [2] Rania Gihleb, Osea Giuntella, Luca Stella, and Tianyi Wang. Industrial robots, workers’ safety, and health. 2020.
- [3] Akira Taniguchi, Shota Isobe, Lotfi El Hafi, Yoshinobu Hagiwara, and Tadahiyo Taniguchi. Autonomous planning based on spatial concepts to tidy up home environments with service robots. *Advanced Robotics*, pages 1–19, 2021.
- [4] Kazuko Obayashi, Naonori Kodate, and Shigeru Masuyama. Can connected technologies improve sleep quality and safety of older adults and care-givers? an evaluation study of sleep monitors and communicative robots at a residential care home in japan. *Technology in Society*, 62:101318, 2020.
- [5] Steve Whittaker, Yvonne Rogers, Elena Petrovskaya, and Hongbin Zhuang. Designing personas for expressive robots: Personality in the new breed of moving, speaking, and colorful social home robots. *ACM Transactions on Human-Robot Interaction (THRI)*, 10(1):1–25, 2021.
- [6] Dylan Jennings and Miguel Figliozzi. Study of road autonomous delivery robots and their potential effects on freight efficiency and travel. *Transportation Research Record*, 2674(9):1019–1029, 2020.
- [7] Hengli Wang, Rui Fan, Yuxiang Sun, and Ming Liu. Applying surface normal information in drivable area and road anomaly detection for ground mobile robots. *arXiv preprint arXiv:2008.11383*, 2020.
- [8] Mordechai Ben-Ari and Francesco Mondada. Robots and their applications. In *Elements of robotics*, pages 1–20. Springer, 2018.
- [9] Khalid Colchester, Hani Hagrass, Daniyal Alghazzawi, and Ghadah Aldabbagh. A survey of artificial intelligence techniques employed for adaptive educational systems within e-learning platforms. *Journal of Artificial Intelligence and Soft Computing Research*, 7(1):47–64, 2017.

## References

- [10] V Kavitha and Resham Lohani. A critical study on the use of artificial intelligence, e-learning technology and tools to enhance the learners experience. *Cluster Computing*, 22(3):6985–6989, 2019.
- [11] Jorge Ribeiro, Almeida Dias, José Marques, Liliana Ávidos, Isabel Araújo, Nuno Araújo, and Margarida Figueiredo. An artificial intelligence case based approach to motivational students assessment in (e)-learning environments. In *Proceedings of the 10th International Conference on E-Education, E-Business, E-Management and E-Learning*, pages 1–6, 2019.
- [12] Ibrar Yaqoob, Latif U Khan, SM Ahsan Kazmi, Muhammad Imran, Nadra Guizani, and Choong Seon Hong. Autonomous driving cars in smart cities: Recent advances, requirements, and challenges. *IEEE Network*, 34(1):174–181, 2019.
- [13] Sebastian Ramos, Stefan Gehrig, Peter Pinggera, Uwe Franke, and Carsten Rother. Detecting unexpected obstacles for self-driving cars: Fusing deep learning and geometric modeling. In *2017 IEEE Intelligent Vehicles Symposium (IV)*, pages 1025–1032. IEEE, 2017.
- [14] Mario Hirz and Bernhard Walzel. Sensor and object recognition technologies for self-driving cars. *Computer-aided design and applications*, 15(4):501–508, 2018.
- [15] Arsénio Reis, Isabel Barroso, Maria João Monteiro, Salik Khanal, Vitor Rodrigues, Vitor Filipe, Hugo Paredes, and João Barroso. Designing autonomous systems in interactions with elderly people. In *International Conference on Universal Access in Human-Computer Interaction*, pages 603–611. Springer, 2017.
- [16] Arsénio Reis, Rui Xavier, Isabel Barroso, Maria João Monteiro, Hugo Paredes, and João Barroso. The usage of telepresence robots to support the elderly. In *2018 2nd International Conference on Technology and Innovation in Sports, Health and Wellbeing (TISHW)*, pages 1–6. IEEE, 2018.
- [17] Mai Ali, Asma Asim Ali, Abd-Elhamid Taha, Imed Ben Dhaou, and Tuan Nguyen Gia. Intelligent autonomous elderly patient home monitoring system. In *ICC 2019-2019 IEEE International Conference on Communications (ICC)*, pages 1–6. IEEE, 2019.
- [18] Alessandro Vercelli, Innocenzo Rainero, Ludovico Ciferri, Marina Boido, and Fabrizio Pirri. Robots in elderly care. *DigitCult-Scientific Journal on Digital Cultures*, 2(2):37–50, 2018.
- [19] Fernando V Paulovich, Maria Cristina F De Oliveira, and Osvaldo N Oliveira Jr. A future with ubiquitous sensing and intelligent systems. *ACS sensors*, 3(8):1433–1438, 2018.
- [20] Lech Józwiak. Advanced mobile and wearable systems. *Microprocessors and microsystems*, 50:202–221, 2017.

## References

- [21] João J Ferreira, Cristina I Fernandes, Hussain G Rammal, and Pedro M Veiga. Wearable technology and consumer interaction: A systematic review and research agenda. *Computers in Human Behavior*, page 106710, 2021.
- [22] Mahdi Tavakoli, Jay Carriere, and Ali Torabi. Robotics, smart wearable technologies, and autonomous intelligent systems for healthcare during the covid-19 pandemic: An analysis of the state of the art and future vision. *Advanced Intelligent Systems*, 2(7):2000071, 2020.
- [23] Elena Hernández-Nieves, Álvaro Bartolomé del Canto, Pablo Chamoso-Santos, Fernando de la Prieta-Pintado, and Juan M Corchado-Rodríguez. A machine learning platform for stock investment recommendation systems. In *International Symposium on Distributed Computing and Artificial Intelligence*, pages 303–313. Springer, 2020.
- [24] Chong Feng, Muzammil Khan, Arif Ur Rahman, and Arshad Ahmad. News recommendation systems-accomplishments, challenges & future directions. *IEEE Access*, 8:16702–16725, 2020.
- [25] Eleni Stai, Stella Kafetzoglou, Eirini Eleni Tsiropoulou, and Symeon Papavassiliou. A holistic approach for personalization, relevance feedback & recommendation in enriched multimedia content. *Multimedia Tools and Applications*, 77(1):283–326, 2018.
- [26] Andreas Argyriou, Miguel González-Fierro, and Le Zhang. Microsoft recommenders: Best practices for production-ready recommendation systems. In *Companion Proceedings of the Web Conference 2020*, pages 50–51, 2020.
- [27] Sahraoui Dhelim, Nyothiri Aung, Mohammed Amine Bouras, Huansheng Ning, and Erik Cambria. A survey on personality-aware recommendation systems. *arXiv preprint arXiv:2101.12153*, 2021.
- [28] Vasiliki Pouli, Stella Kafetzoglou, Eirini Eleni Tsiropoulou, Aggeliki Dimitriou, and Symeon Papavassiliou. Personalized multimedia content retrieval through relevance feedback techniques for enhanced user experience. In *2015 13th International Conference on Telecommunications (ConTEL)*, pages 1–8, 2015.
- [29] Sanorita Dey, Brittany RL Duff, Niyati Chhaya, Wai Fu, Vishy Swaminathan, and Karrie Karahalios. Recommendation for video advertisements based on personality traits and companion content. In *Proceedings of the 25th International Conference on Intelligent User Interfaces*, pages 144–154, 2020.
- [30] Cisco Visual Networking. Cisco global cloud index: forecast and methodology, 2015–2020. *White paper by Cisco*, 2017.
- [31] Niklas Heuvelod et al. Ericsson mobility report. *Ericsson, Stockholm*, 2017.
- [32] Chetna Singhal and Swades De. *Resource allocation in next-generation broadband wireless access networks*. IGI Global, 2017.

## References

- [33] Oliver Blume Kostas Pentikousis, Ramón Agüero Calvo, and Symeon Papavassiliou. *Mobile networks and management*. Springer, 2010.
- [34] G. Dileep. A survey on smart grid technologies and applications. *Renewable Energy*, 146:2589–2625, 2020.
- [35] Xin-Lin Huang, Xiaomin Ma, and Fei Hu. Machine learning and intelligent communications. *Mobile Networks and Applications*, 23(1):68–70, 2018.
- [36] Nafis Irtija, Fisayo Sangoleye, and Eirini Eleni Tsiropoulou. Contract-theoretic demand response management in smart grid systems. *IEEE Access*, 8:184976–184987, 2020.
- [37] Shahzad Khan, Devashish Paul, Parham Momtahan, and Moayad Aloqaily. Artificial intelligence framework for smart city microgrids: State of the art, challenges, and opportunities. In *2018 Third International Conference on Fog and Mobile Edge Computing (FMEC)*, pages 283–288. IEEE, 2018.
- [38] Nathan Patrizi, Georgios Fragkos, Eirini Eleni Tsiropoulou, and Symeon Papavassiliou. Contract-theoretic resource control in wireless powered communication public safety systems. In *GLOBECOM 2020 - 2020 IEEE Global Communications Conference*, pages 1–6, 2020.
- [39] Ayca Kirimtat, Ondrej Krejcar, Attila Kertesz, and M Fatih Tasgetiren. Future trends and current state of smart city concepts: A survey. *IEEE Access*, 8:86448–86467, 2020.
- [40] Yazdan Ahmad Qadri, Ali Nauman, Yousaf Bin Zikria, Athanasios V Vasilakos, and Sung Won Kim. The future of healthcare internet of things: a survey of emerging technologies. *IEEE Communications Surveys & Tutorials*, 22(2):1121–1167, 2020.
- [41] Wazir Zada Khan, MH Rehman, Hussein Mohammed Zangoti, Muhammad Khalil Afzal, Nasrullah Armi, and Khaled Salah. Industrial internet of things: Recent advances, enabling technologies and open challenges. *Computers & Electrical Engineering*, 81:106522, 2020.
- [42] Fisayo Sangoleye, Nafis Irtija, and Eirini Eleni Tsiropoulou. Smart energy harvesting for internet of things networks. *Sensors*, 21(8):2755, 2021.
- [43] A Abdellah and A Koucheryavy. Survey on artificial intelligence techniques in 5g networks. *journal of Information Technology and Telecommunications, SPbSUT, Russia*, 8(1):1–10, 2020.
- [44] Delia Rico and Pedro Merino. A survey of end-to-end solutions for reliable low-latency communications in 5g networks. *IEEE Access*, 8:192808–192834, 2020.
- [45] Jiayang Liu, Lin Zhong, Jehan Wickramasuriya, and Venu Vasudevan. uwave: Accelerometer-based personalized gesture recognition and its applications. *Pervasive and Mobile Computing*, 5(6):657–675, 2009.

## References

- [46] Ali Al-Shuwaili and Osvaldo Simeone. Energy-efficient resource allocation for mobile edge computing-based augmented reality applications. *IEEE Wireless Communications Letters*, 6(3):398–401, 2017.
- [47] Shaoxuan Wang and Sujit Dey. Adaptive mobile cloud computing to enable rich mobile multimedia applications. *IEEE Transactions on Multimedia*, 15(4):870–883, 2013.
- [48] Adriano M Gil, Afonso R Costa, Atacilio C Cunha, Thiago S Figueira, and Antonio A Silva. Video player architecture for virtual reality on mobile devices. In *International Conference on Human-Computer Interaction*, pages 91–100. Springer, 2020.
- [49] Narges Ashtari, Andrea Bunt, Joanna McGrenere, Michael Nebeling, and Parmit K Chilana. Creating augmented and virtual reality applications: Current practices, challenges, and opportunities. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pages 1–13, 2020.
- [50] Ke Zhang, Yuming Mao, Supeng Leng, Yejun He, and Yan Zhang. Mobile-edge computing for vehicular networks: A promising network paradigm with predictive off-loading. *IEEE Vehicular Technology Magazine*, 12(2):36–44, 2017.
- [51] Xiang Sun and Nirwan Ansari. Edgeiot: Mobile edge computing for the internet of things. *IEEE Communications Magazine*, 54(12):22–29, 2016.
- [52] Pavlos Athanasios Apostolopoulos, Georgios Fragkos, Eirini Eleni Tsiropoulou, and Symeon Papavassiliou. Data offloading in uav-assisted multi-access edge computing systems under resource uncertainty. *IEEE Transactions on Mobile Computing*, pages 1–1, 2021.
- [53] Mohammad Tawalbeh, Alan Eardley, et al. Studying the energy consumption in mobile devices. *Procedia Computer Science*, 94:183–189, 2016.
- [54] John Nash. Non-cooperative games. *Annals of mathematics*, pages 286–295, 1951.
- [55] Abbas Yazdinejad, Gautam Srivastava, Reza M Parizi, Ali Dehghantanha, Kim-Kwang Raymond Choo, and Mohammed Aledhari. Decentralized authentication of distributed patients in hospital networks using blockchain. *IEEE journal of biomedical and health informatics*, 24(8):2146–2156, 2020.
- [56] Viktoriia Shubina, Sylvia Holcer, Michael Gould, and Elena Simona Lohan. Survey of decentralized solutions with mobile devices for user location tracking, proximity detection, and contact tracing in the covid-19 era. *Data*, 5(4):87, 2020.
- [57] Nina Slamnik-Kriještorec, Haris Kremono, Marco Ruffini, and Johann M Marquez-Barja. Sharing distributed and heterogeneous resources toward end-to-end 5g networks: A comprehensive survey and a taxonomy. *IEEE Communications Surveys & Tutorials*, 22(3):1592–1628, 2020.

## References

- [58] Claudio Cicconetti, Marco Conti, Andrea Passarella, and Dario Sabella. Toward distributed computing environments with serverless solutions in edge systems. *IEEE Communications Magazine*, 58(3):40–46, 2020.
- [59] Roger B Myerson. *Game theory*. Harvard university press, 2013.
- [60] Yuhua Xu, Jinlong Wang, and Qihui Wu. Distributed learning of equilibria with incomplete, dynamic, and uncertain information in wireless communication networks. In *Game Theory Framework Applied to Wireless Communication Networks*, pages 63–86. IGI global, 2016.
- [61] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of machine learning*. MIT press, 2018.
- [62] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.
- [63] Sanyam Kapoor. Multi-agent reinforcement learning: A report on challenges and approaches. *arXiv preprint arXiv:1807.09427*, 2018.
- [64] Michał Lewandowski. Prospect theory versus expected utility theory: Assumptions, predictions, intuition and modelling of risk attitudes. *Central European Journal of Economic Modelling and Econometrics*, pages 275–321, 2017.
- [65] Daniel Kahneman and Amos Tversky. Prospect theory: An analysis of decision under risk. In *Handbook of the fundamentals of financial decision making: Part I*, pages 99–127. World Scientific, 2013.
- [66] Ling Tang and Shibo He. Multi-user computation offloading in mobile edge computing: A behavioral perspective. *IEEE Network*, 32(1):48–53, 2018.
- [67] Giorgos Mitsis, Pavlos Athanasios Apostolopoulos, Eirini Eleni Tsiropoulou, and Symeon Papavassiliou. Intelligent dynamic data offloading in a competitive mobile edge computing market. *Future Internet*, 11(5):118, 2019.
- [68] Pavlos Athanasios Apostolopoulos, Marcos Torres, and Eirini Eleni Tsiropoulou. Satisfaction-aware data offloading in surveillance systems. In *Proceedings of the 14th Workshop on Challenged Networks*, pages 21–26, 2019.
- [69] Pavlos Athanasios Apostolopoulos, Eirini Eleni Tsiropoulou, and Symeon Papavassiliou. Cognitive data offloading in mobile edge computing for internet of things. *IEEE Access*, 8:55736–55749, 2020.
- [70] Pavlos Athanasios Apostolopoulos, Eirini Eleni Tsiropoulou, and Symeon Papavassiliou. Risk-aware data offloading in multi-server multi-access edge computing environment. *IEEE/ACM Transactions on Networking*, 28(3):1405–1418, 2020.

## References

- [71] Panagiotis Promponas, Pavlos Athanasios Apostolopoulos, Eirini Eleni Tsiropoulou, and Symeon Papavassiliou. Redesigning resource management in wireless networks based on games in satisfaction form. In *2019 12th IFIP Wireless and Mobile Networking Conference (WMNC)*, pages 24–31. IEEE, 2019.
- [72] Pavlos Athanasios Apostolopoulos, Eirini Eleni Tsiropoulou, and Symeon Papavassiliou. Demand response management in smart grid networks: A two-stage game-theoretic learning-based approach. *Mobile Networks and Applications*, pages 1–14, 2018.
- [73] Pavlos Athanasios Apostolopoulos, Eirini Eleni Tsiropoulou, and Symeon Papavassiliou. Risk-aware social cloud computing based on serverless computing model. In *2019 IEEE Global Communications Conference (GLOBECOM)*, pages 1–6, 2019.
- [74] Georgios Fragkos, Pavlos Athanasios Apostolopoulos, and Eirini Eleni Tsiropoulou. Escape: Evacuation strategy through clustering and autonomous operation in public safety systems. *Future Internet*, 11(1):20, 2019.
- [75] Nathan Patrizi, Pavlos Athanasios Apostolopoulos, Kelly Rael, and Eirini Eleni Tsiropoulou. Socio-physical human orchestration in smart cities. In *2019 IEEE International Conference on Smart Computing (SMARTCOMP)*, pages 115–120, 2019.
- [76] Pavlos Athanasios Apostolopoulos, Zehui Wang, Hanson Wang, Chad Zhou, Kittipat Virochsiri, Norm Zhou, and Igor L. Markov. Personalization for web-based services using offline reinforcement learning, 2021.
- [77] Oskar Morgenstern and John Von Neumann. *Theory of games and economic behavior*. Princeton university press, 1953.
- [78] Jon Bendor, Daniel Diermeier, and Michael M Ting. The empirical content of adaptive models. 2004.
- [79] Yan Zhang and Mohsen Guizani. *Game theory for wireless communications and networking*. CRC press, 2011.
- [80] Quanyan Zhu and T Başar. Decision and game theory for security, 2013.
- [81] John F Nash et al. Equilibrium points in n-person games. *Proceedings of the national academy of sciences*, 36(1):48–49, 1950.
- [82] Symeon Papavassiliou, Eirini Eleni Tsiropoulou, Panagiotis Promponas, and Panagiotis Vamvakas. A paradigm shift toward satisfaction, realism and efficiency in wireless networks resource sharing. *IEEE Network*, 35(1):348–355, 2021.
- [83] Stéphane Ross and Brahim Chaib-draa. Satisfaction equilibrium: Achieving cooperation in incomplete information games. In *Conference of the Canadian Society for Computational Studies of Intelligence*, pages 61–72. Springer, 2006.

## References

- [84] Stéphane Ross and Brahim Chaib-draa. Learning to play a satisfaction equilibrium. In *Workshop on Evolutionary Models of Collaboration*, pages 2332–7790, 2007.
- [85] Samir M Perlaza, Hamidou Tembine, Samson Lasaulce, and Mérouane Debbah. Satisfaction equilibrium: A general framework for qos provisioning in self-configuring networks. In *2010 IEEE Global Telecommunications Conference GLOBECOM 2010*, pages 1–5. IEEE, 2010.
- [86] David Roger Smart. *Fixed point theorems*, volume 66. Cup Archive, 1980.
- [87] Samir M Perlaza, Hamidou Tembine, Samson Lasaulce, and Mérouane Debbah. Quality-of-service provisioning in decentralized networks: A satisfaction equilibrium approach. *IEEE Journal of Selected Topics in Signal Processing*, 6(2):104–116, 2011.
- [88] Bronisław Knaster. Un theoreme sur les fonctions d’ensembles. *Ann. Soc. Polon. Math.*, 6:133–134, 1928.
- [89] Eitan Altman and Zwi Altman. S-modular games and power control in wireless networks. *IEEE Transactions on Automatic Control*, 48(5):839–842, 2003.
- [90] Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.
- [91] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484–489, 2016.
- [92] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.
- [93] O Vinyals, T Ewalds, S Bartunov, P Georgiev, AS Vezhnevets, M Yeo, A Makhzani, H Küttler, J Agapiou, J Schrittwieser, et al. A new challenge for reinforcement learning. *arXiv preprint ArXiv:1708.04782*, 2017.
- [94] Victor do Nascimento Silva and Luiz Chaimowicz. Moba: a new arena for game ai. *arXiv preprint arXiv:1705.10443*, 2017.
- [95] Richard S Sutton and Andrew G Barto. Reinforcement learning. *Journal of Cognitive Neuroscience*, 11(1):126–134, 1999.
- [96] Michael Lederman Littman. *Algorithms for sequential decision making*. Brown University Providence, RI, 1996.
- [97] Martin L Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, Virtual, 2014.



## References

- [98] Richard Bellman. Dynamic programming and lagrange multipliers. *Proceedings of the National Academy of Sciences of the United States of America*, 42(10):767, 1956.
- [99] Nils Nilsson. Teleo-reactive programs for agent control. *Journal of artificial intelligence research*, 1:139–158, 1993.
- [100] W Thomas Miller, Paul J Werbos, and Richard S Sutton. *Neural networks for control*. MIT press, 1995.
- [101] Charles Eric Leiserson, Ronald L Rivest, Thomas H Cormen, and Clifford Stein. Introduction to algorithms. vol. 6, 2001.
- [102] PK Suetin, Alexandra I Kostrikin, and Yu I Manin. *Linear algebra and geometry*. CRC Press, 1989.
- [103] DP Bertsekas. Dynamic programming and optimal control, volume 1. athena scientific belmont, ma, –2005. isbn: 1-886529-26-4 (vol. i, ). *Vol. II,*, pages 1–886529.
- [104] Vincent François-Lavet, Peter Henderson, Riashat Islam, Marc G Bellemare, and Joelle Pineau. An introduction to deep reinforcement learning. *arXiv preprint arXiv:1811.12560*, 2018.
- [105] Ji Li, Hui Gao, Tiejun Lv, and Yueming Lu. Deep reinforcement learning based computation offloading and resource allocation for mec. In *2018 IEEE Wireless Communications and Networking Conference (WCNC)*, pages 1–6. IEEE, 2018.
- [106] Sergey Levine, Chelsea Finn, Trevor Darrell, and Pieter Abbeel. End-to-end training of deep visuomotor policies. *The Journal of Machine Learning Research*, 17(1):1334–1373, 2016.
- [107] Xinlei Pan, Yurong You, Ziyang Wang, and Cewu Lu. Virtual to real reinforcement learning for autonomous driving. *arXiv preprint arXiv:1704.03952*, 2017.
- [108] Yue Deng, Feng Bao, Youyong Kong, Zhiqian Ren, and Qionghai Dai. Deep direct reinforcement learning for financial signal representation and trading. *IEEE transactions on neural networks and learning systems*, 28(3):653–664, 2016.
- [109] Vincent François-Lavet. *Contributions to deep reinforcement learning and its applications in smartgrids*. PhD thesis, Université de Liège, Liège, Belgique, 2017.
- [110] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.
- [111] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. PMLR, 2015.

## References

- [112] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.
- [113] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31, 2017.
- [114] Günter Klambauer, Thomas Unterthiner, Andreas Mayr, and Sepp Hochreiter. Self-normalizing neural networks. *arXiv preprint arXiv:1706.02515*, 2017.
- [115] Richard S. Sutton, David McAllester, Satinder Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Proc. 12th International Conference on Neural Information Processing Systems*, NIPS’99, page 1057–1063, Cambridge, MA, USA, 1999. MIT Press.
- [116] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.
- [117] Samson Lasaulce and Hamidou Tembine. *Game theory and learning for wireless networks: fundamentals and applications*. Academic Press, 2011.
- [118] Ann Nowé, Peter Vrancx, and Yann-Michaël De Hauwere. Game theory and multi-agent reinforcement learning. In *Reinforcement Learning*, pages 441–470. Springer, 2012.
- [119] Mohammad S Obaidat, Georgios I Papadimitriou, and Andreas S Pomportsis. Guest editorial learning automata: theory, paradigms, and applications. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 32(6):706–709, 2002.
- [120] Eirini Eleni Tsiropoulou, Georgios K Katsinis, Alexandros Filios, and Symeon Papavassiliou. On the problem of optimal cell selection and uplink power control in open access multi-service two-tier femtocell networks. In *International Conference on Ad-Hoc Networks and Wireless*, pages 114–127. Springer, 2014.
- [121] Katja Verbeeck and Ann Nowe. Colonies of learning automata. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 32(6):772–780, 2002.
- [122] Yuhua Xu, Jinlong Wang, Qihui Wu, Alagan Anpalagan, and Yu-Dong Yao. Opportunistic spectrum access in unknown dynamic environment: A game-theoretic stochastic learning solution. *IEEE transactions on wireless communications*, 11(4):1380–1391, 2012.
- [123] Jason R Marden, Gürdal Arslan, and Jeff S Shamma. Cooperative control and potential games. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 39(6):1393–1407, 2009.

## References

- [124] H Peyton Young. *Individual strategy and social structure: An evolutionary theory of institutions*. Princeton University Press, 2020.
- [125] Yuhua Xu, Qihui Wu, Liang Shen, Jinlong Wang, and Alagan Anpalagan. Opportunistic spectrum access with spatial reuse: Graphical game and uncoupled learning solutions. *IEEE Transactions on Wireless Communications*, 12(10):4814–4826, 2013.
- [126] Kelly Rael, Georgios Fragkos, Jim Plusquellic, and Eirini Eleni Tsiropoulou. Uav-enabled human internet of things. In *2020 16th International Conference on Distributed Computing in Sensor Systems (DCOSS)*, pages 312–319, 2020.
- [127] Georgios Fragkos, Eirini Eleni Tsiropoulou, and Symeon Papavassiliou. Artificial intelligence enabled distributed edge computing for internet of things applications. In *2020 16th International Conference on Distributed Computing in Sensor Systems (DCOSS)*, pages 450–457, 2020.
- [128] Georgios Fragkos, Cyrus Minwalla, Jim Plusquellic, and Eirini Eleni 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)*, pages 1–6, 2020.
- [129] Georgios Fragkos, Cyrus Minwalla, Jim Plusquellic, and Eirini Eleni Tsiropoulou. Artificially intelligent electronic money. *IEEE Consumer Electronics Magazine*, pages 1–1, 2020.
- [130] Ashish R Hota, Siddharth Garg, and Shreyas Sundaram. Fragility of the commons under prospect-theoretic risk attitudes. *Games and Economic Behavior*, 98:135–164, 2016.
- [131] Garrett Hardin. The tragedy of the commons. *Journal of Natural Resources Policy Research*, 1(3):243–253, 2009.
- [132] Panagiotis Vamvakas, Eirini Eleni Tsiropoulou, and Symeon Papavassiliou. Dynamic spectrum management in 5g wireless networks: A real-life modeling approach. In *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications*, pages 2134–2142, 2019.
- [133] Mark J Machina. Choice under uncertainty: Problems solved and unsolved. *Journal of Economic Perspectives*, 1(1):121–154, 1987.
- [134] Nicholas C Barberis. Thirty years of prospect theory in economics: A review and assessment. *Journal of Economic Perspectives*, 27(1):173–96, 2013.
- [135] Minae Kwon, Erdem Biyik, Aditi Talati, Karan Bhasin, Dylan P Losey, and Dorsa Sadigh. When humans aren’t optimal: Robots that collaborate with risk-aware humans. In *Proceedings of the 2020 ACM/IEEE International Conference on Human-Robot Interaction*, pages 43–52, 2020.

## References

- [136] Chuan Zhang, Yu-xin Tian, Ling-wei Fan, and Ying-hui Li. Customized ranking for products through online reviews: a method incorporating prospect theory with an improved vikor. *Applied Intelligence*, pages 1–20, 2020.
- [137] Peter P Wakker and Jingni Yang. A powerful tool for analyzing concave/convex utility and weighting functions. *Journal of Economic Theory*, 181:143–159, 2019.
- [138] Panagiotis Vamvakas, Eirini Eleni Tsiropoulou, and Symeon Papavassiliou. On controlling spectrum fragility via resource pricing in 5g wireless networks. *IEEE Network Letters*, 1(3):111–115, 2019.
- [139] Giorgos Mitsis, Eirini Eleni Tsiropoulou, and Symeon Papavassiliou. Data offloading in uav-assisted multi-access edge computing systems: A resource-based pricing and user risk-awareness approach. *Sensors*, 20(8):2434, 2020.
- [140] Panagiotis Vamvakas, Eirini Eleni Tsiropoulou, and Symeon Papavassiliou. Risk-aware resource control with flexible 5g access technology interfaces. In *2019 IEEE 20th International Symposium on "A World of Wireless, Mobile and Multimedia Networks" (WoWMoM)*, pages 1–9, 2019.
- [141] Panagiotis Vamvakas, Eirini Eleni Tsiropoulou, and Symeon Papavassiliou. Risk-aware resource management in public safety networks. *Sensors*, 19(18):3853, 2019.
- [142] Panagiotis Vamvakas, Eirini Eleni Tsiropoulou, and Symeon Papavassiliou. On the prospect of uav-assisted communications paradigm in public safety networks. In *IEEE INFOCOM WKSHPS: WCNEE 2019: Wireless Comm. and Net. in Extreme Env.*, pages 1–6. IEEE, 2019.
- [143] Liang Xiao, Jinliang Liu, Yan Li, Narayan B Mandayam, and H Vincent Poor. Prospect theoretic analysis of anti-jamming communications in cognitive radio networks. In *2014 IEEE Global Comm. Conf.*, pages 746–751, 2014.
- [144] Panagiotis Vamvakas, Eirini Eleni Tsiropoulou, and Symeon Papavassiliou. Exploiting prospect theory and risk-awareness to protect uav-assisted network operation. *EURASIP Journal on Wireless Communications and Networking*, 2019(1):286, 2019.
- [145] Athina Thanou, Eirini Eleni Tsiropoulou, and Symeon Papavassiliou. Quality of experience under a prospect theoretic perspective: A cultural heritage space use case. *IEEE Transactions on Computational Social Systems*, 6(1):135–148, 2019.
- [146] Athina Thanou, Eirini Eleni Tsiropoulou, and Symeon Papavassiliou. A sociotechnical approach to the museum congestion management problem. *IEEE Transactions on Computational Social Systems*, 7(2):563–568, 2020.
- [147] Cisco Visual Networking. Internet of things at a glance. *White paper by Cisco*, 2017.
- [148] Ericsson. Ericsson mobility report - q4 2018. *Ericsson Mobility Report*, 2019.

## References

- [149] Georgios Fragkos, Sean Lebien, and Eirini Eleni Tsiropoulou. Artificial intelligent multi-access edge computing servers management. *IEEE Access*, 8:171292–171304, 2020.
- [150] Bowen Zhou and Rajkumar Buyya. Augmentation techniques for mobile cloud computing: A taxonomy, survey, and future directions. *ACM Computing Surveys (CSUR)*, 51(1):1–38, 2018.
- [151] Khadija Akherfi, Micheal Gerndt, and Hamid Harroud. Mobile cloud computing for computation offloading: Issues and challenges. *Applied computing and informatics*, 14(1):1–16, 2018.
- [152] Yuyi Mao, Changsheng You, Jun Zhang, Kaibin Huang, and Khaled B Letaief. A survey on mobile edge computing: The communication perspective. *IEEE Communications Surveys & Tutorials*, 19(4):2322–2358, 2017.
- [153] Weisong Shi, Jie Cao, Quan Zhang, Youhuizi Li, and Lanyu Xu. Edge computing: Vision and challenges. *IEEE internet of things journal*, 3(5):637–646, 2016.
- [154] Georgios Fragkos, Nicholas Kemp, Eirini Eleni Tsiropoulou, and Symeon Papavassiliou. Artificial intelligence empowered uavs data offloading in mobile edge computing. In *ICC 2020 - 2020 IEEE International Conference on Communications (ICC)*, pages 1–7, 2020.
- [155] Xiaolong Xu, Xing Zhang, Xihua Liu, Jieliang Jiang, Lianyong Qi, and Md Zakirul Alam Bhuiyan. Adaptive computation offloading with edge for 5g-envisioned internet of connected vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 2020.
- [156] Mark Billingham. Rapid prototyping for ar/vr experiences. In *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems*, pages 1–4, 2020.
- [157] Mohd Abdul Ahad, Sara Paiva, Gautami Tripathi, and Noushaba Feroz. Enabling technologies and sustainable smart cities. *Sustainable cities and society*, 61:102301, 2020.
- [158] Chun Sing Lai, Youwei Jia, Zhekang Dong, Dongxiao Wang, Yingshan Tao, Qi Hong Lai, Richard TK Wong, Ahmed F Zobaa, Ruiheng Wu, and Loi Lei Lai. A review of technical standards for smart cities. *Clean Technologies*, 2(3):290–310, 2020.
- [159] Quoc-Viet Pham, Fang Fang, Vu Nguyen Ha, Md Jalil Piran, Mai Le, Long Bao Le, Won-Joo Hwang, and Zhiguo Ding. A survey of multi-access edge computing in 5g and beyond: Fundamentals, technology integration, and state-of-the-art. *IEEE Access*, 8:116974–117017, 2020.
- [160] Keith Kirkpatrick. Software-defined networking. *Communications of the ACM*, 56(9):16–19, 2013.

## References

- [161] Manar Jammal, Taranpreet Singh, Abdallah Shami, Rasool Asal, and Yiming Li. Software defined networking: State of the art and research challenges. *Computer Networks*, 72:74–98, 2014.
- [162] A. C. Baktir, A. Ozgovde, and C. Ersoy. How can edge computing benefit from software-defined networking: A survey, use cases, and future directions. *IEEE Communications Surveys Tutorials*, 19(4):2359–2391, 2017.
- [163] Cheng-Xiang Wang, Marco Di Renzo, Slawomir Stanczak, Sen Wang, and Erik G Larsson. Artificial intelligence enabled wireless networking for 5g and beyond: Recent advances and future challenges. *IEEE Wireless Communications*, 27(1):16–23, 2020.
- [164] M. E. Morocho Cayamcela and W. Lim. Artificial intelligence in 5g technology: A survey. In *2018 International Conference on Information and Communication Technology Convergence (ICTC)*, pages 860–865, 2018.
- [165] Peter Henderson, Riashat Islam, Philip Bachman, Joelle Pineau, Doina Precup, and David Meger. Deep reinforcement learning that matters. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.
- [166] Nguyen Cong Luong, Dinh Thai Hoang, Shimin Gong, Dusit Niyato, Ping Wang, Ying-Chang Liang, and Dong In Kim. Applications of deep reinforcement learning in communications and networking: A survey. *IEEE Communications Surveys & Tutorials*, 21(4):3133–3174, 2019.
- [167] Ahmed A Al-habob and Octavia A Dobre. Mobile edge computing and artificial intelligence: A mutually-beneficial relationship. *arXiv preprint arXiv:2005.03100*, 2020.
- [168] Zehui Xiong, Yang Zhang, Dusit Niyato, Ruilong Deng, Ping Wang, and Li-Chun Wang. Deep reinforcement learning for mobile 5g and beyond: Fundamentals, applications, and challenges. *IEEE Vehicular Technology Magazine*, 14(2):44–52, 2019.
- [169] Faris B Mismar, Brian L Evans, and Ahmed Alkhateeb. Deep reinforcement learning for 5g networks: Joint beamforming, power control, and interference coordination. *IEEE Transactions on Communications*, 68(3):1581–1592, 2019.
- [170] An Wang, Zili Zha, Yang Guo, and Songqing Chen. Software-defined networking enhanced edge computing: A network-centric survey. *Proceedings of the IEEE*, 107(8):1500–1519, 2019.
- [171] Ali Shakarami, Mostafa Ghobaei-Arani, and Ali Shahidinejad. A survey on the computation offloading approaches in mobile edge computing: A machine learning-based perspective. *Computer Networks*, page 107496, 2020.

## References

- [172] Zhuang Chen, Qian He, Lei Liu, Dapeng Lan, Hwei-Ming Chung, and Zhifei Mao. An artificial intelligence perspective on mobile edge computing. In *2019 IEEE International Conference on Smart Internet of Things (SmartIoT)*, pages 100–106. IEEE, 2019.
- [173] Shuiguang Deng, Hailiang Zhao, Weijia Fang, Jianwei Yin, Schahram Dustdar, and Albert Y Zomaya. Edge intelligence: the confluence of edge computing and artificial intelligence. *IEEE Internet of Things Journal*, 7(8):7457–7469, 2020.
- [174] Yuyi Mao, Jun Zhang, SH Song, and Khaled Ben Letaief. Power-delay tradeoff in multi-user mobile-edge computing systems. In *IEEE Global Communications Conference (GLOBECOM)*, pages 1–6, 2016.
- [175] Xu Chen, Lei Jiao, Wenzhong Li, and Xiaoming Fu. Efficient multi-user computation offloading for mobile-edge cloud computing. *IEEE/ACM Transactions on Networking*, (5):2795–2808, 2016.
- [176] Ke Zhang, Yuming Mao, Supeng Leng, Quanxin Zhao, Longjiang Li, Xin Peng, Li Pan, Sabita Maharjan, and Yan Zhang. Energy-efficient offloading for mobile edge computing in 5g heterogeneous networks. *IEEE access*, 4:5896–5907, 2016.
- [177] Hongyan Yu, Qu Yuan Wang, and Songtao Guo. Energy-efficient task offloading and resource scheduling for mobile edge computing. In *2018 IEEE Internat. Conf. on Networking, Architecture and Storage*, pages 1–4. IEEE, 2018.
- [178] Te-Yi Kan, Yao Chiang, and Hung-Yu Wei. Task offloading and resource allocation in mobile-edge computing system. In *27th IEEE Wireless and Optical Communication Conference (WOCC)*, pages 1–4, 2018.
- [179] Stefania Sardellitti, Gesualdo Scutari, and Sergio Barbarossa. Joint optimization of radio and computational resources for multicell mobile-edge computing. *IEEE Trans. on Signal & Inf. Proc. over Net.*, 1(2):89–103, 2015.
- [180] Changsheng You, Kaibin Huang, Hyukjin Chae, and Byoung-Hoon Kim. Energy-efficient resource allocation for mobile-edge computation offloading. *IEEE Transactions on Wireless Communications*, 16(3):1397–1411, 2016.
- [181] Yinghao Yu, Jun Zhang, and Khaled B Letaief. Joint subcarrier and cpu time allocation for mobile edge computing. In *2016 IEEE Global Communications Conference (GLOBECOM)*, pages 1–6. IEEE, 2016.
- [182] Qiang Fan and Nirwan Ansari. Towards workload balancing in fog computing empowered iot. *IEEE Trans. on Network Science and Engin.*, 2018.
- [183] Abbas Kiani, Nirwan Ansari, and Abdallah Khreishah. Hierarchical capacity provisioning for fog computing. *IEEE/ACM Trans. on Networking*, 2019.

## References

- [184] Jingjing Yao and Nirwan Ansari. Qos-aware fog resource provisioning and mobile device power control in iot networks. *IEEE Transactions on Network and Service Management*, 16(1):167–175, 2018.
- [185] Tian Zhang. Data offloading in mobile edge computing: A coalition and pricing based approach. *IEEE Access*, 6:2760–2767, 2017.
- [186] Yongqiang Zhang, Jianbo He, and Songtao Guo. Energy-efficient dynamic task offloading for energy harvesting mobile cloud computing. In *IEEE International Conference on Networking, Architecture and Storage*, pages 1–4, 2018.
- [187] Yuyi Mao, Jun Zhang, and Khaled B Letaief. Dynamic computation offloading for mobile-edge computing with energy harvesting devices. *IEEE Journal on Sel. Areas in Com.*, 34(12):3590–3605, 2016.
- [188] Kai Guo, Mingcong Yang, Yongbing Zhang, and Yusheng Ji. An efficient dynamic offloading approach based on optimization technique for mobile edge computing. In *IEEE Int. Conf. on Mob. Cl. Comp., Ser., and Eng.*, pages 29–36, 2018.
- [189] Georgios Fragkos, Nathan Patrizi, Eirini Eleni Tsiropoulou, and Symeon Papavassiliou. Socio-aware public safety framework design: A contract theory based approach. In *ICC 2020 - 2020 IEEE International Conference on Communications (ICC)*, pages 1–7, 2020.
- [190] Georgios Fragkos, Eirini Eleni Tsiropoulou, and Symeon Papavassiliou. Disaster management and information transmission decision-making in public safety systems. In *2019 IEEE Global Communications Conference (GLOBECOM)*, pages 1–6, 2019.
- [191] Minh Le, Zheng Song, Young-Woo Kwon, and Eli Tilevich. Reliable and efficient mobile edge computing in highly dynamic and volatile environments. In *Fog and Mobile Edge Computing (FMEC), 2017 Second International Conference on*, pages 113–120. IEEE, 2017.
- [192] Xiaojuan Wei, Shangguang Wang, Ao Zhou, Jinliang Xu, Sen Su, Sathish Kumar, and Fangchun Yang. Mvr: an architecture for computation offloading in mobile edge computing. In *Edge Computing (EDGE), 2017 IEEE International Conference on*, pages 232–235. IEEE, 2017.
- [193] Andreas Reiter, Bernd Prünster, and Thomas Zefferer. Hybrid mobile edge computing: Unleashing the full potential of edge computing in mobile device use cases. In *Proceedings of the 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*, pages 935–944. IEEE Press, 2017.
- [194] István Ketykó, László Kecskés, Csaba Nemes, and Lóránt Farkas. Multi-user computation offloading as multiple knapsack problem for 5g mobile edge computing. In *Eur. Conf. on Networks and Com.*, pages 225–229, 2016.



## References

- [195] Yanting Wang, Min Sheng, Xijun Wang, Liang Wang, and Jiandong Li. Mobile-edge computing: Partial computation offloading using dynamic voltage scaling. *IEEE Trans.on Communications*, 64(10):4268–4282, 2016.
- [196] Weiwei Chen, Dong Wang, and Keqin Li. Multi-user multi-task computation offloading in green mobile edge cloud computing. *IEEE Transactions on Services Computing*, 2018.
- [197] Yaser Jararweh, Ahmad Doulat, Ala Darabseh, Mohammad Alsmirat, Mahmoud Al-Ayyoub, and Elhadj Benkhelifa. Sdmec: Software defined system for mobile edge computing. In *2016 IEEE International Conference on Cloud Engineering Workshop (IC2EW)*, pages 88–93. IEEE, 2016.
- [198] Juan Wang and Di Li. Adaptive computing optimization in software-defined network-based industrial internet of things with fog computing. *Sensors*, 18(8), 2018.
- [199] Ammar Muthanna, Regina Shamilova, Abdelhamied A Ateya, Alexander Paramonov, and Mohammad Hammoudeh. A mobile edge computing/software-defined networking-enabled architecture for vehicular networks. *Internet Technology Letters*, 3(6):e109, 2020.
- [200] P. Porambage, J. Okwuibe, M. Liyanage, M. Ylianttila, and T. Taleb. Survey on multi-access edge computing for internet of things realization. *IEEE Communications Surveys Tutorials*, 20(4):2961–2991, 2018.
- [201] Anta Huang, Navid Nikaein, Tore Stenbock, Adlen Ksentini, and Christian Bonnet. Low latency mec framework for sdn-based lte/lte-a networks. In *2017 IEEE International Conference on Communications (ICC)*, pages 1–6. IEEE, 2017.
- [202] Bander Alzahrani and Nikos Fotiou. Enhancing internet of things security using software-defined networking. *Journal of Systems Architecture*, 110:101779, 2020.
- [203] M. S. Hossain, C. Xu, Y. Li, A. K. Pathan, J. Bilbao, W. Zeng, and A. El Sadik. Impact of next-generation mobile technologies on iot-cloud convergence. *IEEE Communications Magazine*, 55(1):18–19, 2017.
- [204] Zhao Chen and Xiaodong Wang. Decentralized computation offloading for multi-user mobile edge computing: A deep reinforcement learning approach. *EURASIP Journal on Wireless Communications and Networking*, 2020(1):1–21, 2020.
- [205] Timothy P. Lillicrap, Jonathan J. Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning, 2019.
- [206] Yameng Zhang, Tong Liu, Yanmin Zhu, and Yuanyuan Yang. A deep reinforcement learning approach for online computation offloading in mobile edge computing. In *2020 IEEE/ACM 28th International Symposium on Quality of Service (IWQoS)*, pages 1–10. IEEE, 2020.

## References

- [207] Lei Lei, Huijuan Xu, Xiong Xiong, Kan Zheng, Wei Xiang, and Xianbin Wang. Multiuser resource control with deep reinforcement learning in iot edge computing. *IEEE Internet of Things Journal*, 6(6):10119–10133, 2019.
- [208] Rui Zhao, Xinjie Wang, Junjuan Xia, and Liseng Fan. Deep reinforcement learning based mobile edge computing for intelligent internet of things. *Physical Communication*, 43:101184, 2020.
- [209] Eirini Eleni Tsiropoulou, Panagiotis Vamvakas, and Symeon 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*, 1(1):17–28, 2017.
- [210] Christopher Z Mooney. *Monte carlo simulation*. Number 116. Sage, 1997.
- [211] Prabhu Natarajan, Pradeep K Atrey, and Mohan Kankanhalli. Multi-camera coordination and control in surveillance systems: A survey. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 11(4):57, 2015.
- [212] NF Kahar, RB Ahmad, Z Hussin, and ANC Rosli. Embedded smart camera performance analysis. In *2009 International Conference on Computer Engineering and Technology*, volume 2, pages 79–83. IEEE, 2009.
- [213] Setiya Purbaya, Endro Ariyanto, Dodi Wisaksono Sudiharto, and Catur Wirawan Wijiutomo. Improved image quality on surveillance embedded ip camera by reducing noises. In *3rd Intern. Conf. on Science in Inform. Techn.*, pages 156–159. IEEE, 2017.
- [214] Ning Chen, Yu Chen, Yang You, Haibin Ling, Pengpeng Liang, and Roger Zimmermann. Dynamic urban surveillance video stream processing using fog computing. In *IEEE 2nd Intern. Conf. on Multimedia Big Data*, pages 105–112. IEEE, 2016.
- [215] Naser Hossein Motlagh, Miloud Bagaa, and Tarik Taleb. Uav-based iot platform: A crowd surveillance use case. *IEEE Communications Magazine*, 55(2):128–134, 2017.
- [216] Bongjae Kim, Hong Min, Junyoung Heo, and Jinman Jung. Dynamic computation offloading scheme for drone-based surveillance systems. *Sensors*, 18(9):2982, 2018.
- [217] Teodor Tomic, Korbinian Schmid, Philipp Lutz, Andreas Domel, Michael Kassecker, Elmar Mair, Iris Lynne Grix, Felix Ruess, Michael Suppa, and Darius Burschka. Toward a fully autonomous uav: Research platform for indoor and outdoor urban search and rescue. *IEEE robotics & automation magazine*, 19(3):46–56, 2012.
- [218] Michail Fasoulakis, Eirini Eleni Tsiropoulou, and Symeon Papavassiliou. Satisfy instead of maximize: Improving operation efficiency in wireless communication networks. *Computer Networks*, 2019.

## References

- [219] Christopher JCH Watkins and Peter Dayan. Q-learning. *Machine learning*, 8(3-4):279–292, 1992.
- [220] Tommi Jaakkola, Michael I Jordan, and Satinder P Singh. Convergence of stochastic iterative dynamic programming algorithms. In *Advances in neural information processing systems*, pages 703–710, 1994.
- [221] Satinder Singh, Tommi Jaakkola, Michael L Littman, and Csaba Szepesvári. Convergence results for single-step on-policy reinforcement-learning algorithms. *Machine learning*, 38(3):287–308, 2000.
- [222] Eduard Bertran and Alex Sánchez-Cerdà. On the tradeoff between electrical power consumption and flight performance in fixed-wing uav autopilots. *IEEE Transactions on Vehicular Technology*, 65(11):8832–8840, 2016.
- [223] Cem U Saraydar, Narayan B Mandayam, and David J Goodman. Efficient power control via pricing in wireless data networks. *IEEE Transactions on Communications*, 50(2):291–303, 2002.
- [224] Nicholas Barberis, Ming Huang, and Tano Santos. Prospect theory and asset prices. *The quarterly journal of economics*, 116(1):1–53, 2001.
- [225] Gerhard P Fettweis. The tactile internet: Applications and challenges. *IEEE Vehicular Technology Magazine*, 9(1):64–70, 2014.
- [226] Efe A Ok. *Real analysis with economic applications*, volume 10. Princeton University Press, 2007.
- [227] SB Russ. A translation of bolzano’s paper on the intermediate value theorem. *Historia Mathematica*, 7(2):156–185, 1980.
- [228] Pradeep Dubey, Ori Haimanko, and Andriy Zapechelnyuk. Strategic complements and substitutes, and potential games. *Games and Economic Behavior*, 54(1):77–94, 2006.
- [229] Thomas H Cormen, Charles E Leiserson, Ronald L Rivest, and Clifford Stein. *Introduction to algorithms*. MIT press, 2009.
- [230] Antti P Miettinen and Jukka K Nurminen. Energy efficiency of mobile clients in cloud computing. *HotCloud*, 10:4–4, 2010.
- [231] Meng-Hsi Chen, Ben Liang, and Min Dong. Joint offloading and resource allocation for computation and communication in mobile cloud with computing access point. In *IEEE INFOCOM*, pages 1–9, 2017.
- [232] Xu Chen. Decentralized computation offloading game for mobile cloud computing. *IEEE Transactions on Parallel and Distributed Systems*, 26(4):974–983, 2015.

## References

- [233] Garret1t Hardin. Extensions of the tragedy of the commons. *Science*, 280(5364):682–683, 1998.
- [234] Knut Sydsaeter. *Mathematics for economic analysis*. Pearson Education India, 2013.
- [235] Feliks Ruvimovich Gantmakher. *The theory of matrices*, volume 131. American Mathematical Soc., 2000.
- [236] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.
- [237] J Ben Rosen. Existence and uniqueness of equilibrium points for concave n-person games. *Econom.: Jour. of the Econ. Soc.*, pages 520–534, 1965.
- [238] Chiang Kao. Performance of several nonlinear programming software packages on microcomputers. *Computers & operations research*, 25(10):807–816, 1998.
- [239] Paul T Boggs and Jon W Tolle. Sequential quadratic programming. *Acta numerica*, 4:1–51, 1995.
- [240] Eirini Eleni Tsiropoulou, Panagiotis Vamvakas, and Symeon Papavassiliou. Super-modular game-based distributed joint uplink power and rate allocation in two-tier femtocell networks. *IEEE Transactions on Mobile Computing*, 16(9):2656–2667, 2017.
- [241] Eirini Eleni Tsiropoulou, Georgios K Katsinis, and Symeon Papavassiliou. Distributed uplink power control in multiservice wireless networks via a game theoretic approach with convex pricing. *IEEE Transactions on Parallel and Distributed Systems*, 23(1):61–68, 2012.
- [242] Jun Zhao, Wei Zheng, Xiangming Wen, Xiaoli Chu, Haijun Zhang, and Zhaoming Lu. Game theory based energy-aware uplink resource allocation in ofdma femtocell networks. *International Journal of Distributed Sensor Networks*, 10(3):658158, 2014.
- [243] Peng Xu, Xuming Fang, Meirong Chen, and Yang Xu. A stackelberg game-based spectrum allocation scheme in macro/femtocell hierarchical networks. *Computer Communications*, 36(14):1552–1558, 2013.
- [244] Eirini Eleni Tsiropoulou, Panagiotis Vamvakas, and Symeon Papavassiliou. Joint utility-based uplink power and rate allocation in wireless networks: a non-cooperative game theoretic framework. *Physical Communication*, 9:299–307, 2013.
- [245] Azamossadat Hosseinzadeh Salati, Masoumeh Nasiri-Kenari, and Parastoo Sadeghi. Distributed subband, rate and power allocation in ofdma based two-tier femtocell networks using fractional frequency reuse. In *IEEE Wireless Communications and Networking Conference (WCNC)*, pages 2626–2630, 2012.

## References

- [246] Jianmin Zhang, Zhaoyang Zhang, Kedi Wu, and Aiping Huang. Optimal distributed subchannel, rate and power allocation algorithm in ofdm-based two-tier femtocell networks. In *IEEE Vehicular Technology Conference (VTC 2010-Spring)*, pages 1–5, 2010.
- [247] Eirini Eleni Tsiropoulou, Panagiotis Vamvakas, and Symeon Papavassiliou. Energy efficient uplink joint resource allocation non-cooperative game with pricing. In *IEEE Wireless Communications and Networking Conference (WCNC)*, pages 2352–2356, 2012.
- [248] Soumya Sen, Carlee Joe-Wong, Sangtae Ha, and Mung Chiang. Smart data pricing (sdp): Economic solutions to network congestion. *Recent advances in networking*, 1:221–274, 2013.
- [249] Michail Fasoulakis, Eirini-Eleni Tsiropoulou, and Symeon Papavassiliou. A new theoretical evaluation framework for satisfaction equilibria in wireless networks. *arXiv preprint arXiv:1806.01905*, 2018.
- [250] Hajar Elhammouti, Essaid Sabir, Mustapha Benjillali, Loubna Echabbi, and Hamidou Tembine. Self-organized connected objects: Rethinking qos provisioning for iot services. *IEEE Comm. Magazine*, 55(9):41–47, 2017.
- [251] Samir M Perlaza, Hamidou Tembine, Samson Lasaulce, and Mérouane Debbah. Quality-of-service provisioning in decentralized networks: A satisfaction equilibrium approach. *IEEE Journal of Selected Topics in Signal Processing*, 6(2):104–116, 2012.
- [252] Mathew Goonewardena, Samir M Perlaza, Animesh Yadav, and Wessam Ajib. Generalized satisfaction equilibrium: A model for service-level provisioning in networks. In *22th European Wireless Conference*, pages 1–5, 2016.
- [253] François Mériaux, Samir Perlaza, Samson Lasaulce, Zhu Han, and Vincent Poor. Achievability of efficient satisfaction equilibria in self-configuring networks. In *Int. Conf. on Game Theory for Networks*, pages 1–15. Springer, 2012.
- [254] Paweł Sroka and Adrian Kliks. Playing radio resource management games in dense wireless 5g networks. *Mobile Information Systems*, 2016.
- [255] Luca Rose, Samson Lasaulce, Samir M Perlaza, and Merouane Debbah. Learning equilibria with partial information in decentralized wireless networks. *IEEE communications Magazine*, 49(8):136–142, 2011.
- [256] B. Knaster and A. Tarski. Un theoreme sur les fonctions d’ensembles. *Ann. Soc. Polon. Math.*, 6:133–134, 1928.
- [257] Xi Fang, Satyajayant Misra, Guoliang Xue, and Dejun Yang. Smart grid—the new and improved power grid: A survey. *IEEE communications surveys & tutorials*, 14(4):944–980, 2011.

## References

- [258] V Cagri Gungor, Dilan Sahin, Taskin Kocak, Salih Ergut, Concettina Buccella, Carlo Cecati, and Gerhard P Hancke. A survey on smart grid potential applications and communication requirements. *IEEE Transactions on industrial informatics*, 9(1):28–42, 2012.
- [259] Wenye Wang, Yi Xu, and Mohit Khanna. A survey on the communication architectures in smart grid. *Computer networks*, 55(15):3604–3629, 2011.
- [260] J Swift, JD Carey, and DL O’Connor. Market monitor: Electric industry restructuring,”. *Division of Energy Resources, Office of Consumer Affairs and Business Regulation, MA, USA*, pages 1–6, 2002.
- [261] DL Nelson, KW Anderson, and BM Marquez. Report to the 84th texas legislature: scope of com-petition in electric markets in texas. *Public Utility Commission of Texas, Austin*, pages 1–142, 2015.
- [262] Amir-Hamed Mohsenian-Rad and Alberto Leon-Garcia. Optimal residential load control with price prediction in real-time electricity pricing environments. *IEEE Trans. Smart Grid*, 1(2):120–133, 2010.
- [263] Antonio J Conejo, Juan M Morales, and Luis Baringo. Real-time demand response model. *IEEE Transactions on Smart Grid*, 1(3):236–242, 2010.
- [264] Shengnan Shao, Manisa Pipattanasomporn, and Saifur Rahman. Demand response as a load shaping tool in an intelligent grid with electric vehicles. *IEEE Transactions on Smart Grid*, 2(4):624–631, 2011.
- [265] Marco Zugno, Juan Miguel Morales, Pierre Pinson, and Henrik Madsen. A bilevel model for electricity retailers’ participation in a demand response market environment. *Energy Economics*, 36:182–197, 2013.
- [266] Pedram Samadi, Amir-Hamed Mohsenian-Rad, Robert Schober, Vincent WS Wong, and Juri Jatskevich. Optimal real-time pricing algorithm based on utility maximization for smart grid. In *2010 First IEEE International Conference on Smart Grid Communications*, pages 415–420. IEEE, 2010.
- [267] Shuoyao Wang, Suzhi Bi, and Ying-Jun Angela Zhang. Demand response management for profit maximizing energy loads in real-time electricity market. *IEEE Transactions on Power Systems*, 33(6):6387–6396, 2018.
- [268] Shahab Bahrami, M Hadi Amini, Miadreza Shafie-Khah, and Joao PS Catalao. A decentralized renewable generation management and demand response in power distribution networks. *IEEE Transactions on Sustainable Energy*, 9(4):1783–1797, 2018.
- [269] Jun Li, Guangqing Ma, Tao Li, Wushun Chen, and Yu Gu. A stackelberg game approach for demand response management of multi-microgrids with overlapping sales areas. *Science China Information Sciences*, 62(11):212203, 2019.

## References

- [270] Alireza SoltaniNejad Farsangi, Shahrzad Hadayeghparast, Mehdi Mehdinejad, and Heidarali Shayanfar. A novel stochastic energy management of a microgrid with various types of distributed energy resources in presence of demand response programs. *Energy*, 160:257–274, 2018.
- [271] Renzhi Lu, Seung Ho Hong, and Mengmeng Yu. Demand response for home energy management using reinforcement learning and artificial neural network. *IEEE Transactions on Smart Grid*, 2019.
- [272] Radu Godina, Eduardo Rodrigues, Edris Pouresmaeil, João Matias, and João Catalão. Model predictive control home energy management and optimization strategy with demand response. *Applied Sciences*, 8(3):408, 2018.
- [273] Yao Yao, Xing He, Tingwen Huang, Chaojie Li, and Dawen Xia. A projection neural network for optimal demand response in smart grid environment. *Neural Computing and Applications*, 29(6):259–267, 2018.
- [274] Santosh Kumar Desai, Amit Dua, Neeraj Kumar, Ashok Kumar Das, and Joel JPC Rodrigues. Demand response management using lattice-based cryptography in smart grids. In *2018 IEEE Global Communications Conference (GLOBECOM)*, pages 1–6. IEEE, 2018.
- [275] Hossein Mohammadi Ruzbahani and Hadis Karimipour. Optimal incentive-based demand response management of smart households. In *2018 IEEE/IAS 54th Industrial and Commercial Power Systems Technical Conference (I&CPS)*, pages 1–7. IEEE, 2018.
- [276] Anish Jindal, Mukesh Singh, and Neeraj Kumar. Consumption-aware data analytical demand response scheme for peak load reduction in smart grid. *IEEE Transactions on Industrial Electronics*, 65(11):8993–9004, 2018.
- [277] Ehsan Saeidpour Parizy, Hamid Reza Bahrami, and Seungdeog Choi. A low complexity and secure demand response technique for peak load reduction. *IEEE Transactions on Smart Grid*, 10(3):3259–3268, 2018.
- [278] C Abreu, D Rua, P Machado, JA Peças Lopes, and Miguel Heleno. Advanced energy management for demand response and microgeneration integration. In *2018 Power Systems Computation Conference (PSCC)*, pages 1–7. IEEE, 2018.
- [279] Wei-Yu Chiu, Jui-Ting Hsieh, and Chia-Ming Chen. Pareto optimal demand response based on energy costs and load factor in smart grid. *IEEE Transactions on Industrial Informatics*, 2019.
- [280] Peicong Luo, Xiaoying Wang, Hailong Jin, Yuling Li, and Xuejiao Yang. Load management for multiple datacenters towards demand response in the smart grid integrating renewable energy. In *Proceedings of the 2018 2nd International Conference on Computer Science and Artificial Intelligence*, pages 140–144. ACM, 2018.

## References

- [281] Xiao Kou, Fangxing Li, Jin Dong, Michael Starke, Jeffrey Munk, Teja Kuruganti, and Helia Zandi. A distributed energy management approach for residential demand response. Technical report, Oak Ridge National Lab.(ORNL), Oak Ridge, TN (United States), 2019.
- [282] Renzhi Lu, Seung Ho Hong, and Xiongfeng Zhang. A dynamic pricing demand response algorithm for smart grid: reinforcement learning approach. *Applied Energy*, 220:220–230, 2018.
- [283] Lijun Chen, Na Li, Steven H Low, and John C Doyle. Two market models for demand response in power networks. In *2010 First IEEE International Conference on Smart Grid Communications*, pages 397–402. IEEE, 2010.
- [284] Na Li, Lijun Chen, and Steven H Low. Optimal demand response based on utility maximization in power networks. In *2011 IEEE power and energy society general meeting*, pages 1–8. IEEE, 2011.
- [285] S Datchanamoorthy, S Kumar, Y Ozturk, and G Lee. Optimal time-of-use pricing for residential load control. In *2011 IEEE International Conference on Smart Grid Communications (SmartGridComm)*, pages 375–380. IEEE, 2011.
- [286] Amir-Hamed Mohsenian-Rad, Vincent WS Wong, Juri Jatskevich, Robert Schober, and Alberto Leon-Garcia. Autonomous demand-side management based on game-theoretic energy consumption scheduling for the future smart grid. *IEEE transactions on Smart Grid*, 1(3):320–331, 2010.
- [287] Christian Ibars, Monica Navarro, and Lorenza Giupponi. Distributed demand management in smart grid with a congestion game. In *2010 First IEEE International Conference on Smart Grid Communications*, pages 495–500. IEEE, 2010.
- [288] Chen Chen, Shalinee Kishore, and Lawrence V Snyder. An innovative rtp-based residential power scheduling scheme for smart grids. In *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5956–5959. IEEE, 2011.
- [289] Sabita Maharjan, Quanyan Zhu, Yan Zhang, Stein Gjessing, and Tamer Basar. Dependable demand response management in the smart grid: A stackelberg game approach. *IEEE Transactions on Smart Grid*, 4(1):120–132, 2013.
- [290] Zhong Fan, Parag Kulkarni, Sedat Gormus, Costas Efthymiou, Georgios Kalogridis, Mahesh Sooriyabandara, Ziming Zhu, Sangarapillai Lambotharan, and Woon Hau Chin. Smart grid communications: Overview of research challenges, solutions, and standardization activities. *IEEE Communications Surveys & Tutorials*, 15(1):21–38, 2012.
- [291] Perukrishnen Vytelingum, Sarvapali D Ramchurn, Thomas D Voice, Alex Rogers, and Nicholas R Jennings. Trading agents for the smart electricity grid. In *Proceedings*



## References

- of the 9th International Conference on Autonomous Agents and Multiagent Systems: volume 1-Volume 1*, pages 897–904. International Foundation for Autonomous Agents and Multiagent Systems, 2010.
- [292] Bo Chai, Jiming Chen, Zaiyue Yang, and Yan Zhang. Demand response management with multiple utility companies: A two-level game approach. *IEEE Transactions on Smart Grid*, 5(2):722–731, 2014.
- [293] Peter Palensky and Dietmar Dietrich. Demand side management: Demand response, intelligent energy systems, and smart loads. *IEEE transactions on industrial informatics*, 7(3):381–388, 2011.
- [294] Murat Erkok, Eeyad Al-Ahmadi, Nurcin Celik, and Walid Saad. A game theoretic approach for load-shifting in the smart grid. In *2015 IEEE International Conference on Smart Grid Communications (SmartGridComm)*, pages 187–192. IEEE, 2015.
- [295] Naouar Yaagoubi and Hussein T Mouftah. A comfort based game theoretic approach for load management in the smart grid. In *2013 IEEE Green Technologies Conference (GreenTech)*, pages 35–41. IEEE, 2013.
- [296] Peng Yang, Gongguo Tang, and Arye Nehorai. A game-theoretic approach for optimal time-of-use electricity pricing. *IEEE Transactions on Power Systems*, 28(2):884–892, 2012.
- [297] Wenbo Shi, Na Li, Xiaorong Xie, Chi-Cheng Chu, and Rajit Gadh. Optimal residential demand response in distribution networks. *IEEE journal on selected areas in communications*, 32(7):1441–1450, 2014.
- [298] Yunjian Xu, Na Li, and Steven H Low. Demand response with capacity constrained supply function bidding. *IEEE Transactions on Power Systems*, 31(2):1377–1394, 2015.
- [299] Mengmeng Yu and Seung Ho Hong. A real-time demand-response algorithm for smart grids: A stackelberg game approach. *IEEE Transactions on Smart Grid*, 7(2):879–888, 2015.
- [300] Khaled Alshehri, Ji Liu, Xudong Chen, and Tamer Başar. A stackelberg game for multi-period demand response management in the smart grid. In *2015 54th IEEE Conference on Decision and Control (CDC)*, pages 5889–5894. IEEE, 2015.
- [301] Eirini Eleni Tsiropoulou, Panagiotis Vamvakas, Georgios K Katsinis, and Symeon Papavassiliou. Combined power and rate allocation in self-optimized multi-service two-tier femtocell networks. *Computer Communications*, 72:38–48, 2015.
- [302] Roy D. Yates et al. A framework for uplink power control in cellular radio systems. *IEEE Journal on selected areas in communications*, 13(7):1341–1347, 1995.

## References

- [303] T. Qiu, B. Chen, A. K. Sangaiah, J. Ma, and R. Huang. A survey of mobile social networks: Applications, social characteristics, and challenges. *IEEE Systems J.*, 12(4):3932–3947, Dec 2018.
- [304] Kyle Chard, Kris Bubendorfer, Simon Caton, and Omer F Rana. Social cloud computing: A vision for socially motivated resource sharing. *IEEE Transactions on Services Computing*, 5(4):551–563, 2012.
- [305] Lei Jiao, Jun Lit, Wei Du, and Xiaoming Fu. Multi-objective data placement for multi-cloud socially aware services. In *IEEE INFOCOM*, pages 1–9, 2014.
- [306] Alfonso Pérez, Germán Moltó, Miguel Caballer, and Amanda Calatrava. Serverless computing for container-based architectures. *Future Generation Computer Systems*, 83:50–59, 2018.
- [307] Wes Lloyd, Shruti Ramesh, Swetha Chinthalapati, Lan Ly, and Shrideep Pallickara. Serverless computing: An investigation of factors influencing microservice performance. In *Int. Conf. on Cloud Eng.*, pages 159–169. IEEE, 2018.
- [308] Tarek Elgamal. Costless: Optimizing cost of serverless computing through function fusion and placement. In *IEEE/ACM Symposium on Edge Computing (SEC)*, pages 300–312. IEEE, 2018.
- [309] AWS Lambda.
- [310] Google Cloud Functions.
- [311] G van Rossum. Python tutorial, technical report cs-r9526, centrum voor wiskunde en informatica (cwi), amsterdam.". 1995.
- [312] MATLAB. *version 7.10.0 (R2010a)*. The MathWorks Inc., Natick, Massachusetts, 2019.
- [313] Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, September 2020.
- [314] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris,

## References

- Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020.
- [315] The pandas development team. pandas-dev/pandas: Pandas, February 2020.
- [316] J. D. Hunter. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95, 2007.
- [317] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.
- [318] Abhinav Goel, Caleb Tung, Yung-Hsiang Lu, and George K Thiruvathukal. A survey of methods for low-power deep learning and computer vision. In *2020 IEEE 6th World Forum on Internet of Things (WF-IoT)*, pages 1–6. IEEE, 2020.
- [319] Xu Chen, Lei Jiao, Wenzhong Li, and Xiaoming Fu. Efficient multi-user computation offloading for mobile-edge cloud computing. *IEEE/ACM Transactions on Networking*, 24(5):2795–2808, 2015.
- [320] Michael Till Beck, Martin Werner, Sebastian Feld, and S Schimper. Mobile edge computing: A taxonomy. In *Proc. of the Sixth International Conference on Advances in Future Internet*, pages 48–55. Citeseer, 2014.
- [321] Alvin E Roth and Marilda Sotomayor. Two-sided matching. *Handbook of game theory with economic applications*, 1:485–541, 1992.
- [322] Walid Saad, Zhu Han, Rong Zheng, Mérouane Debbah, and H Vincent Poor. A college admissions game for uplink user association in wireless small cell networks. In *IEEE INFOCOM 2014-IEEE Conference on Computer Communications*, pages 1096–1104. IEEE, 2014.
- [323] S. Sardellitti, M. Merluzzi, and S. Barbarossa. Optimal association of mobile users to multi-access edge computing resources. In *2018 IEEE International Conference on Communications Workshops (ICC Workshops)*, pages 1–6, 2018.
- [324] Pavlos Athanasios Apostolopoulos, Eirini Eleni Tsiropoulou, and Symeon Papavasiliou. Game-theoretic learning-based qos satisfaction in autonomous mobile edge computing. In *2018 Global Information Infrastructure and Networking Symposium (GIIS)*, pages 1–5. IEEE, 2018.

## References

- [325] Y. Dai, D. Xu, S. Maharjan, and Y. Zhang. Joint computation offloading and user association in multi-task mobile edge computing. *IEEE Transactions on Vehicular Technology*, 67(12):12313–12325, 2018.
- [326] Tuyen X Tran and Dario Pompili. Joint task offloading and resource allocation for multi-server mobile-edge computing networks. *IEEE Transactions on Vehicular Technology*, 68(1):856–868, 2018.
- [327] Chen Xu, Guangyuan Zheng, and Liangrui Tang. Energy-aware user association for noma-based mobile edge computing using matching-coalition game. *IEEE Access*, 8:61943–61955, 2020.
- [328] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [329] Hado van Hasselt, Arthur Guez, and David Silver. Deep reinforcement learning with double q-learning, 2015.
- [330] Ziyu Wang, Tom Schaul, Matteo Hessel, Hado Hasselt, Marc Lanctot, and Nando Freitas. Dueling network architectures for deep reinforcement learning. In *International conference on machine learning*, pages 1995–2003. PMLR, 2016.
- [331] Ying Huang, GuoLiang Wei, and YongXiong Wang. Vd d3qn: the variant of double deep q-learning network with dueling architecture. In *2018 37th Chinese Control Conference (CCC)*, pages 9130–9135. IEEE, 2018.
- [332] Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*, 2020.
- [333] Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative q-learning for offline reinforcement learning. *arXiv preprint arXiv:2006.04779*, 2020.
- [334] Rishabh Agarwal, Dale Schuurmans, and Mohammad Norouzi. An optimistic perspective on offline reinforcement learning. In *International Conference on Machine Learning*, pages 104–114. PMLR, 2020.