Fall 11-2019

Artificial Intelligence Empowered UAVs Data Offloading in Mobile Edge Computing

Nicholas Alexander Kemp
University of New Mexico

Nicholas Kemp

Candidate

Electrical and Computer Engineering

Department

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

Approved by the Thesis Committee:

Eirini Eleni Tsiropoulou, Chairperson

Xiang Sun

Michael Devetsikiotis

Artificial Intelligence Empowered UAVs Data Offloading in Mobile Edge Computing

by

Nicholas Kemp

B.S, Computer Engineering, 2018

THESIS

Submitted in Partial Fulfillment of the
Requirements for the Degree of

Master of Science
Computer Engineering

The University of New Mexico

Albuquerque, New Mexico

December, 2019

Dedication

To my loving wife, Tera.

The road was long and the work was hard;

but you kept me on track and pushed me to succeed.

I know you will be happy when I can no longer use the excuse:

"I have to work on my thesis..."

Acknowledgments

This Master's Thesis was made possible by the hard work and dedication of the PROTON team. First and foremost, I would like to Acknowledge Dr. Erini Eleni Tsiropoulou, Assistant Professor of Electrical and Computer Engineering at the University of New Mexico (UNM) for her guidance and direction throughout this process. Her support during this process was undeniably the driving force in the completion of this thesis. Additionally, I would like to thank Georgios Fragkos for his support in the formulation of this problem. His ability to be available when needed helped this thesis move not only swiftly, but efficiently. Above all else, I would love to thank my beautiful, smart, and dedicated wife, Tera and as well as my ever-growing daughter, Saoirse. Tera pushed me and encouraged me to take this step and made sure that I remained focused and dedicated. Through the many challenges, she believed me and gave me the time I needed to complete this thesis.

Artificial Intelligence Empowered UAVs Data Offloading in Mobile Edge Computing

by

Nicholas Kemp

B.S, Computer Engineering, 2018

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

Abstract

The advances introduced by Unmanned Aerial Vehicles (UAVs) are manifold and have paved the path for the full integration of UAVs, as intelligent objects, into the Internet of Things (IoT). This paper brings artificial intelligence into the UAVs data offloading process in a multi-server Mobile Edge Computing (MEC) environment, by adopting principles and concepts from game theory and reinforcement learning. Initially, the autonomous MEC server selection for partial data offloading is performed by the UAVs, based on the theory of the stochastic learning automata. A non-cooperative game among the UAVs is then formulated to determine the UAVs' data to be offloaded to the selected MEC servers, while the existence of at least one Nash Equilibrium (NE) is proven exploiting the power of submodular games. A best response dynamics framework and two alternative reinforcement learning algorithms are introduced that converge to an NE, and their trade-offs are discussed. The overall framework performance evaluation is achieved via modeling and simulation, in terms of its efficiency and effectiveness, under different operation approaches and scenarios.

Contents

List of Figures

Glossary

Artificial Intelligence the theory and development of computer systems able to perform tasks that normally require human intelligence, such as visual perception, speech recognition, decision-making, and translation between languages.

Game Theory the branch of mathematics concerned with the analysis of strategies for dealing with competitive situations where the outcome of a participant's choice of action depends critically on the actions of other participants.

Machine Learning an application of artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed.

Mobile Edge Computing a form of network architecture that enables cloud computing to be done at the edge of a mobile network.

Nash Equilibrium (in economics and game theory) a stable state of a system involving the interaction of different participants, in which no participant can gain by a unilateral change of strategy if the strategies of the others remain unchanged.

Reinforcement Learning an area of machine learning concerned with how software agents ought to take actions in an environment so as to maximize some notion of cumulative reward.

Unmanned Aerial Vehicles an aircraft piloted by remote control or onboard computers

Chapter 1

Introduction

As the age of the Internet of Things (IoT) steadily grows and the number of connected devices grows into the billions, automation and control of wireless networks is at the forefront of research. Next generation networks, like 5G and beyond are expected to handle the capacity of multiple heterogeneous devices each with diverse computational and communication capabilities; these devices are expected to exchange and process large amounts of data. In an effort to automate and control traffic through the networks these devices will perform their computing tasks in an autonomous matter. Thus, Artificial Intelligence (AI) has presented itself as a powerful tool to support autonomous human-like decision making in next generation networked devices. AI was founded on and is supported by multi-disciplinary techniques, such as machine learning, control theory, game theory, optimization theory, and meta-heuristics [1]. Many of these mathematical models have their roots in economics; however, as networks became more and more crowded it became imperative to use new techniques to design future networks. Therefore, models that already set their precedence in economics became the guiding force for AI and the evolution of next generation networks.

Nevertheless, applying these new models to the current network is not enough to make improvements on future networks; not only must we apply AI and its technology, we must also change the network architecture. As a result of the increased computational demands of the nearly billions of devices connected to the network a new architecture representing the practice of processing data near the edge of the network [2], otherwise known as Multi-Access Edge Computing or Mobile Edge Computing (MEC) has gained momentum as a solution to handle the increased load of the network. More importantly, MEC meets the devices' Quality of Service (QoS) requirements in terms of delay, latency, and energy efficiency. Because of this, MEC is utilized to reduce the amount of data offloading from IoT devices to the cloud and decrease service access latency [2]. Since IoT devices have limited processing capabilities and the overall goal is to decrease the traffic through the network and manage it in an autonomous manner [3], MEC has presented itself as a networking architecture capable of rapid analysis and immediate processing of data.

IoT devices are a large proponent of MEC and recently research has focused on Unmanned Aerial Vehicles (UAVs), also known as drones. Recently, there has been heavy investment in the development of UAVs and multi-UAV systems that can collaborate and complete missions more efficiently. Thus, new and developing technologies, like 5G and beyond have significant potential on UAVs equipped with sensors for delivering IoT services, requiring the execution of computationally intensive tasks [4]. Therefore, it is expected that UAVs will offload data in masses to the network; in such cases, MEC arises as a powerful tool to support the operation of these drones [5].

Motivated by the aforementioned arguments and observations, this thesis proposes an AI-driven data offloading approach to enable the UAVs to optimally offload part of their data to a set of MEC servers for further processing by combining key principles and methodologies from *Game Theory* and *Reinforcement Learning*.

1.1 Related Work

1.1.1 Artificial Intelligence

Intelligence is defined as the ability to acquire and apply knowledge and skills. Such skills include decision making, speech recognition, language translation, visual perception, and others. Therefore, Artificial Intelligence (AI) is the theory and development of computer systems to apply these skills in a human-like manner while at the same time learning to apply these skills better each time. The rise of 4G LTE, the release of 5G, and the latest 6G white paper have been the driving forces behind wireless networks; more specifically, mobile networks have set the tone for the rise of AI.

The role of AI in mobile networks has been researched in [6]. AI has proven to be a successful tool for applications such as computer vision, language processing, and autonomous driving; with the growing trend of AI empowered applications and IoT devices researches expect a large number of these intelligent applications to be deployed at the edge of wireless networks. Therefore, 6G wireless networks will be designed to leverage advanced wireless communications and mobile computing technologies to support AI-enabled applications at various edge mobile devices with limited communication, computation, hardware and energy resources [6]. Research into mobile networks has set the tone for research into AI enabled heterogeneous networks.

Also, the role of AI-based techniques and their use in heterogeneous networks (HetNets) is discussed in [7]. Research into AI-based techniques like self - configuration, self-healing, and self-optimization as researched in [7] can be very beneficial for architectures like MEC. In [8], the benefits that AI will have on all networking architectures are discussed. AI enabled networks can help develop a future vision of cognitive networks that will show network-wide intelligent behavior to solve problems of network heterogeneity, performance, and quality of service (QoS) [8]. Research into AI driven networking architectures is paving the way for IoT devices to smartly and autonomously offload/upload data to/from the network.

1.1.2 Game Theory

Game theory is the study of the ways in which interacting choices of economic agents produce outcomes with respect to the utilities of those agents [9]. Game theory, although original a mathematical model for studying problems in economics [10], has been applied to multiple disciplines including Political Science, Biology, and Psychology. More recently game theory has become a popular model in Computer Science and Logic design; furthermore, cognitive radio networks have benefited greatly from game theory. Because of the successful application of game theory in cognitive radio networks, other network architectures, such as cellular networks [11] and Mobile Edge Computing networks, have also adopted game theory to deal with distributed decision-making-related problems, like the resource management problem.

Game theory has been adopted in the Network Science field to deal with single-resource management problems, such power control, rate control and others. In [12], a power management control problem in Code Division Multiple Access (CDMA)-based wireless networks is introduced and a non-cooperative game among the users is formulated to determine their optimal uplink transmission power that will bring the system in a steady state, i.e., Nash Equilibrium point. This problem has been extended in [13], in order to consider the usage-based pricing (i.e., convex pricing with respect to the users' uplink transmission power) that is imposed to the users regarding their transmission and the corresponding interference that they create within the communication environment. A similar problem is addressed in [14], considering linear pricing to the users' uplink transmission power and determining the non-cooperative game's Nash Equilibrium, i.e., the users' optimal uplink transmission power. In [15], the problem of power control via adopting a game-theoretic approach is addressed in multi-tier wireless communication networks, considering macro-cells and femto-cells. Moreover, in [16], the power control problem is addressed by adopting the theory of S-modular games, i.e., when a user increases its transmission power, the rest of the users decrease their strategy due to the increased interference in the communication environment. This work has been extended in [17], by considering the application in CDMA wireless cellular networks.

Game theory has been also adopted to solve resource management problems in device-to-device communication [18], wireless powered communication networks, where the devices charge their batteries through the radio frequency signals transmitted by the transmitter within the communication environment [19], as well as in communication networks adopting single carrier frequency division multiple access technique, where on top of the power that should be determined by each user, the optimal channel allocation should also be calculated [20].

Additionally, in [21], the authors discuss game theory and its role in networking for multiple resources [22]. In the problem of multi-resources allocation [23], the goal is to find a Nash equilibrium state of the given network regarding QoS metrics such as transmission power and/or rate [24]. In [25], the authors tackle the problem of power and rate allocation in wireless cellular networks, by transforming the problem to a single-variable non-cooperative game and determining its Nash Equilibrium. This problem has been also addressed in [26], by directly addressing the two variables resource management problem as a non-cooperative game, where the authors show that it is an S-modular game and they determine its Nash Equilibrium, i.e., the optimal uplink transmission power and data rate.

Based on the above discussion, it is evident that Game Theory is a powerful tool to address resource management problems in various types of wireless communication networks, considering various types of resources, multi-tier architectures, and multiple access techniques. Also, Game Theory enables the mobile devices to make autonomous decisions and adopt human-like behavior, therefore, it becomes a critical part of the Artificial Intelligent initiative.

In [27, 28] the authors discuss using game theory with learning to find the pure Nash equilibrium of the network. In each of the papers game theory is used as a learning automaton to select a Wireless Internet Service Provider (WISP) or cell, respectfully. In each of the papers discussed above game theory is used to guide the network to a Nash equilibrium. Using the ideas from this research we develop a system of UAVs that will utilize game theory coupled with learning to reach a Nash equilibrium, i.e. where every drone has chosen an optimal MEC.

1.1.3 Reinforcement Learning

Reinforcement Learning is an area of Machine Learning concerned with agents taking a suitable action to maximize their reward. In an RL system, the agents attempt to learn the best action to take based on interactions with an unstable environment; the learning agent is not instructed specifically what action to take, instead it determines the the best action which maximizes its long-term reward. The selected action causes the current state of the environment to transition to the next state and the learning agent receives a scalar reward value that evaluates the effect of the state transition [29].

In [30,31] the authors present two log-linear learning algorithms, B-logit and Max-logit; both algorithms are uncoupled and sequential, have one player perform learning at a time. Log-linear learning is a learning algorithm that provides guarantees on the percentage of time that the action profile will be at a potential maximizer in potential games [32]. Binary log-linear learning (B-logit) is a variant of log-linear learning, its purpose is to handle constrained action sets. Therefore, B-logit is able to handle environments where the future actions of the players are limited based on their current action [30]. B-logit has been used in [33–35] to study Heterogeneous Networks, Wireless Communication Networks, and Opportunistic Spectrum Access (OSA) networks. These papers show that B-logit is an optimal algorithm in finding the pure Nash equilibrium of the system within a few hundred iterations, given the system is shown to be a potential game.

Max log-linear learning (Max-logit) is another variant of log-linear learning which retains the favorable equilibrium selection property with the provably fastest convergence speed over other learning algorithms in the γ-logit family, having a convergence time that is on average 33.85% faster than B-logit [31]. Max-logit has been used in [34–36] to study Heterogeneous Wireless Sensor Networks, Wireless Communication Networks, and Opportunistic Spectrum Access (OSA) networks. Similar to B-logit, these papers show that Max-logit is an optimal algorithm in finding the pure Nash equilibrium of the system within a few hundred iterations, given the system is shown to be a potential game, with the added benefit of being the fastest γ-logit learning algorithm.

1.1.4 Mobile Edge Computing

Mobile Edge Computing (MEC) is showing a rising popularity as a crucial solution to increasing IoT devices' QoS metrics by bringing computing resources to the edge of the network and in close proximity to the end users. In [5, 37–43] the authors discuss the benefits of MEC for smart mobile devices and smart objects in general. As the age of number of smart mobile devices connected to the network grew, so did the amount of traffic in the network; because of this increase in traffic a new paradigm was needed to handle this, thus the concept of MEC came to light. The main benefits of the MEC technology are: its potential to reduce the latency, provide location-awareness, improve the performance of the mobile applications, reduce the energy consumption of the mobile devices by alleviating the burden of executing their computing tasks locally, and provide accurate computing outcomes in a time-wise manner [37]. As we enter further into the age of IoT billions of devices have come online and further increased traffic through the network; therefore, MEC has presented itself as a necessary tool for IoT devices' data offloading.

In [5] the authors discuss using game theory and reinforcement learning to support the autonomous and distributed operation of the MEC servers as well as the process of data offloading by the devices. In [38] the authors research a MEC system with multiple mobile users; a single-user MEC system is a highly researched system and is an easy environment to control. With this research a multi-user MEC system that utilizes game theoretic and reinforcement learning approaches to intelligently manage the network is proposed.

1.1.5 Unmanned Aerial Vehicles

An unmanned aerial vehicle (UAV) , or drone is an aircraft with no human pilot on board. UAVs are one component in an unmanned aircraft system; the entire systems consists of the UAV, a ground-based controller, and a system of communication between the two. UAVs were originally designed and used for military applications; however, there has been a recent influx in drones being used for both private, public, and commercial projects. Like most IoT devices, UAVs are constrained by their battery power and computational capabilities. Because of these constraints we must consider ways to reduce transmission power and reduce the amount of data that is processed locally. Therefore, UAVs will benefit greatly from the MEC architecture.

In [44–47] the authors discuss utilizing UAVs in Public Safety Networks (PSNs) and how to control and automate the drones to improve energy efficiency. In [44, 45] the authors utilize the theory of minority games to have the drones perform their tasks in an independent and distributed manner, then they apply a non-cooperative game theoretic approach to optimize the drones' uplink transmission power. In [46] the authors utilize prospect theory to find a pure Nash equilibrium of the system. Lastly, in [47] the authors suggest a UAV-assisted public safety system based on game theory and reinforcement learning; using a binary log-linear learning algorithm the authors were able to prove a pure Nash equilibrium state of the system with regards to the drones' cost function.

The above literature shows one use for drones, that being for disaster stricken areas. However, as 5G networks and beyond evolve further the use of drones becomes diverse. Drones can be used in private, public, or commercial projects to perform certain compute intensive tasks (i.e. facial recognition or detection and prevention).

1.2 Contributions & Outline

Despite the significant advances that have been obtained in each of the aforementioned areas in isolation, limited research work has been performed in empowering the UAVs' operation and decision-making with adopting the AI technology. AI techniques have been traditionally focused on machine learning frameworks with applications primarily in robotics and image processing, by mainly adopting the artificial neural networks [48]. Game theory has arisen as a crucial element and aspect in AI today, gaining ground in particular in multi-agent systems. In principle, multiple agents can either compete or collaborate to accomplish a task with accuracy and efficiency - the foundation for reinforcement learning in AI. In this paper we adopt a similar philosophy and perspective to support the UAVs autonomous intelligent decision making by adopting game theory and reinforcement learning [1].

To the best of our knowledge, this is the first work in the existing literature where the use of AI techniques, e.g., reinforcement learning and game theory, enables the UAVs to promote human-like decision-making, in terms of selecting a MEC server to offload their computational tasks, and determining the optimal amount of offloaded data to maximize the perceived QoS. The key scientific contributions of our work, that differentiate it from the rest of the existing literature, are summarized as follows:

1. A multi-UAVs and multi-MEC servers environment is considered. The utility of each UAV is formulated as a function of the amount of data that is offloaded to a selected MEC server considering the UAV's transmission cost, the local computing cost, as well as the impact on its perceived QoS by the transmission cost of the rest of the UAVs in combination with the exploitation of the MEC server's computing resources (chapter 2).

2. Based on the theory of submodular games, artificial intelligence is embodied in the decision of the optimal data offloading of each UAV (Section 3.1). A non-cooperative game among the UAVs is formulated with the objective to maximize each UAV's utility function. The game is proven to be submodular, and thus the existence of an NE is shown (Section 3.2).

3. Towards each UAV determining the NE in an autonomous manner, three algorithms are proposed: (i) Best Response Dynamics (Section 3.3), (ii) Max Log-Linear (Max-logit) learning, and (iii) Binary Log-Linear (B-logit) learning. The latter two algorithms are based on the principles of reinforcement learning (Section 3.4).

4. The MEC server selection by each UAV is achieved by intelligently considering each server's reward function depending on its relative computing capability and distance from the UAVs, as well as the QoS that it can potentially provide

to the UAVs. Each UAV acts as a stochastic learning automaton (SLA), which intelligently selects a MEC server to process its data (chapter 4).

5. A series of simulation experiments are realized to evaluate the performance and the inherent attributes of the proposed artificial intelligent UAVs' data offloading approach in the mobile edge computing environment, while a detailed comparative numerical study is presented to demonstrate its benefits (chapter 5). Finally, chapter 6 concludes the paper.

Chapter 2

System Model and UAV's Utility Function

The communication and computing environment described within is defined as:

1. the set of MEC servers $S = \{1, \ldots, s, \ldots, |S|\}$, and
2. the set of UAVs $D = \{1, \ldots, d, \ldots, |D|\}$

denoted as $|S|$ and $|D|$, respectively. Each UAV has a computing task defined as:

$$T_d = (I_d, C_d, \phi_d)$$

where $I_d[bits]$ denotes the total input bits of the computation and $C_d[CPU cycles]$ denotes the number of CPU cycles required to carry out the computing task T_d. Additionally, the parameter $\phi_d[\frac{CPU cycles}{bits}]$ designates the computational complexity of the computing task requested by the UAV; the value of ϕ_d depends on the nature of the application, therefore, a higher ϕ_d value indicates a more computationally intensive task. Additionally, each MEC server $s \in S$ has a computational capability denoted by $F_s[\frac{CPU cycles}{sec}]$, this computational capability F_s defines the MEC server's ability to process all of the UAVs offloaded data. Likewise, each UAV $d \in D$ has a local computational capability denoted by $F_d[\frac{CPU cycles}{sec}]$; additionally, the parameter $\rho_d[\frac{Watts}{CPU cycles}]$ denotes the UAVs local power consumption to process the (remaining) data from the computing task. We consider that each UAV has a fixed maximum power to transmit its data to the chosen MEC server denoted by P^{Max}.

Each UAV $d \in D$ selects one MEC server $s \in S$ to offload either a portion or all of its data in order for the UAVs computing task to be processed, while the remaining data of the computing task are processed locally by the UAV. Therefore, each UAV decides in an autonomous and distributed manner to offload b_d bits of data to the selected MEC server, while the rest of the computational task's data i.e., $(I_d - b_d)$ bits, are processed locally by the UAV, where $b_d \in A_d = [0, I_d]$.

A holistic utility function for each UAV is defined below. The utility function captures the UAV's perceived QoS prerequisites' satisfaction by processing its data in the selected MEC server. The holistic utility function is introduced in six factors, however the factors are not meant to be viewed as just parts but instead understood and then viewed as a whole.

The first term of the UAV's utility function represents its perceived satisfaction from offloading a part or all of its computational task's data and is defined as follows:

$$w_1 b_d(\sum_{\forall s \in S} F_s - \frac{\sum_{\forall s \in S} F_s}{\sum_{\forall d \in D} I_d} e^{\frac{wF_d}{\sum_{\forall d \in D} F_d}} \sum_{\forall i \neq d} b_i - \frac{\sum_{\forall s \in S} F_s}{\sum_{\forall d \in D} I_d} b_d) \tag{2.1a}$$

Term 2.1a is an increasing term, i.e. as the value of b_d increases, the UAV's perceived satisfaction also increases; this is because, as the UAV offloads more of its computational task's data, it will save more of its personal resources. Nevertheless, this term is also driven by the overall computational capability of the entire MEC system and the amount of data the other UAVs offload to the MEC system. That is to say, as the other UAVs offload more data to the MEC system the less the system will be able to serve the UAV, therefore driving the UAV to offload less data. The second term of the UAV's utility function represents the UAV's transmission cost and is defined as follows:

$$-\frac{w_2 P^{Max} b_d}{I_d} \tag{2.1b}$$

Term 2.1b is a decreasing term, i.e. as the percentage of bits offloaded increases, the power required to transmit those bits from the UAV to the MEC server increases, thus lowering the UAV's perceived satisfaction. The third term of the UAV's utility function represents the robustness of the MEC system by observing the amount of bits that the other UAVs offload and is defined as follows:

$$-\frac{w_2 P^{Max} c}{I_d} \sum_{\forall i \neq d} b_i \tag{2.1c}$$

In term 2.1c the constant c is a negative constant where $-1 < c < 0$; therefore, term 2.1c is an increasing term. With that said, if the rest of the UAVs tend to offload large amounts of their computational task's bits to be processed by the MEC system then the examined UAV receives positive feedback, i.e. the UAV perceives the MEC system as being robust. The fourth term of the UAV's utility function represents the UAV's local computing cost associated with processing the remaining data locally and is defined as follows:

$$-w_3 (I_d - b_d) \phi_d \rho_d \tag{2.1d}$$

Term 2.1d is a decreasing term, i.e. as the number of bits processed locally by the UAV increases, the UAV's perceived satisfaction decreases. The fifth term of the UAV's utility function observes the amount of data offloaded by the other UAVs relative to the computational capability of the entire MEC system as is defined as follows:

$$w_1 c \sum_{\forall s \in S} F_s \sum_{\forall i \neq d} b_i \tag{2.1e}$$

Term 2.1e is an decreasing term that serves as positive feedback to the UAV, i.e. the examined UAV tends to offload more of its computational task's bits if it observes the other UAVs tending to the same behavior. Lastly, the sixth term of the UAV's utility function captures the cost that the UAV experiences by the exploitation of the MEC system's computational capabilities by itself and the other UAVs. The sixth term is defined as follows:

$$-w_1 c \sum_{\forall i \neq d} b_i \left[\frac{\sum_{\forall s \in S} F_s}{\sum_{\forall d \in D} I_d} \left(\sum_{\forall i \neq d} b_i \right) + \frac{b_d \sum_{\forall s \in S} F_s}{\sum_{\forall d \in D} I_d} e^{\frac{wF_d}{\sum_{\forall d \in D} F_d}} \right] \tag{2.1f}$$

Term 2.1f is an increasing term, i.e. as the examined UAV and the other UAVs tend to utilize the MEC system's overall computational capabilities, the examined UAV's perceived satisfaction increases.

As stated, the terms are not meant to be viewed as individual equations or separate parts, they are meant to be viewed as pieces that go together to form one holistic utility function as shown below:

$$\begin{aligned}
U_d(b_d, \mathbf{b_{-d}}) = {} & w_1 b_d \Big(\sum_{\forall s \in S} F_s - \frac{\sum_{\forall s \in S} F_s}{\sum_{\forall d \in D} I_d} e^{\frac{wF_d}{\sum_{\forall d \in D} F_d}} \sum_{\forall i \neq d} b_i \\
& - \frac{\sum_{\forall s \in S} F_s}{\sum_{\forall d \in D} I_d} b_d \Big) - \frac{w_2 P^{Max} b_d}{I_d} - \frac{w_2 P^{Max} c}{I_d} \sum_{\forall i \neq d} b_i \\
& - w_3 (I_d - b_d) \phi_d \rho_d + w_1 c \sum_{\forall s \in S} F_s \sum_{\forall i \neq d} b_i \\
& - w_1 c \sum_{\forall i \neq d} b_i \Big[\frac{\sum_{\forall s \in S} F_s}{\sum_{\forall d \in D} I_d} \Big(\sum_{\forall i \neq d} b_i \Big) + \frac{b_d \sum_{\forall s \in S} F_s}{\sum_{\forall d \in D} I_d} e^{\frac{wF_d}{\sum_{\forall d \in D} F_d}} \Big]
\end{aligned} \tag{2.2}$$

In equation 2.2 w, w_1, w_2 and w_3 are positive constants which represent weighting parameters. These weighting parameters are specifically selected to ensure each individual term of the UAVs utility function has the same order of magnitude. Thus, presented is a holistic utility function utilized by each UAV to capture its perceived QoS prerequisites' satisfaction based on the amount of data that is offloaded to a selected MEC server considering the UAV's transmission cost, the local computing cost, as well as the impact on its perceived QoS by the transmission cost of the rest of the UAVs in combination with the exploitation of the MEC server's computing resources.

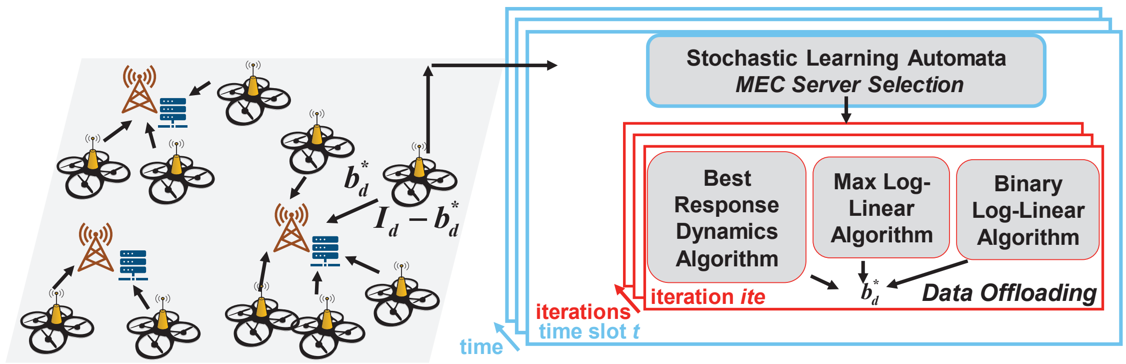

Figure 2.1: Artificial Intelligence Empowered UAVs Data Offloading Framework in Mobile Edge Computing

Figure 2.1 depicts the proposed AI-empowered UAVs data offloading model in a mobile edge computing environment; the proposed model consists of UAVs determining in an intelligent and autonomous matter how much of their computing task's data to offload and to which MEC server to offload that data. In the first stage, each UAV acts as a stochastic learning automaton (SLA). In the SLA stage each UAV determines the optimal MEC server to offload its data in each time slot (chapter 4). After completion of SLA and within the duration of the same time slot, each UAV determines its data offloading strategy, i.e. to which MEC server to offload to (determined in SLA stage) and how much of the computing task's data to offload. This is determined via the UAVs participating in a non-cooperative game (chapter 3). In order to determine the Nash Equilibrium of the non-cooperative game we propose three algorithms: the best response dynamics, max log-linear, and binary log-linear.

Chapter 3

Artificial Intelligent UAV's Data Offloading

In the network, each UAV acts as an artificial intelligence node by making decisions in an autonomous matter as to which MEC server to offload to as well as how much of its computing task's data to offload to that server. In this chapter, a non-cooperative game-theoretic approach based on the theory of submodular games is presented. This environment will enable the UAVs to decide the optimal amount of data to offload to the optimal MEC server in a human-like manor via learning. The process of the MEC server selection via SLA will be discussed in chapter 4.

As a non-cooperative game the proposed environment consists of agents (i.e. the UAVs) seeking to maximize their utility in a selfish manner. In this chapter we will discuss the consequences of an agent attempting to increase their utility by changing their strategy on the other agents in the system. Additionally, we will discuss three algorithms, i.e. Best Response Dynamics and two RL approaches – Max Log-linear and Binary Log-linear – that enable to UAVs to update their strategies and ultimately maximize their utility in a distributed and intelligent manner.

3.1 Data Offloading: An S-Modular Game Perspective

The game, $G = [D, \{A_d\}_{d \in D}, \{U_d\}_{d \in D}]$ is a non-cooperative game formulated by the UAVs; as stated before, D is the set of UAVs, $A_d = [0, I_d]$ is the set of data that the UAV d needs to process for the computation task T_d, and U_d denotes the UAV's utility function. The outcome of the game is a Nash Equilibrium (NE) $\mathbf{b}^* = [\mathbf{b}_1^*, \ldots, \mathbf{b}_{|\mathbf{D}|}^*]$, where b^* denotes the amount of data that each UAV offloads. The NE is a stable point for the overall multi-UAVs and multi-MEC servers system examined herein. At the NE, each UAV offloads an amount of its computing task's data to the selected MEC server in order to maximize its utility function, as follows:

$$\begin{aligned} &\max_{b_d \in A_d} U_d(b_d, \mathbf{b_{-d}}), \qquad \forall d \in D, \\ &\qquad s.t. \qquad 0 \leq b_d \leq I_d \end{aligned} \tag{3.1}$$

Towards proving the existence of at least one NE of the non-cooperative game G as a solution to the maximization problem represented by equation 3.1, we will adopt one theory of S-modular games.

We propose that the non-cooperative game presented herein has the S-modular type structure introduced by Topkis in [49]. These forms of non-cooperative games exhibit interesting properties that are important in applications; these properties include:

1. a Nash equilibrium exists
2. it (the NE) can be attained using greedy best-response type algorithms, and
3. best response policies are monotone in other players' policies [50].

Games that are S-modular fall into two categories, either supermodular or submodular. We start of by defining supermodular games as games characterized by strategic complementarities, i.e. when one player increases its strategy, the other players follow suite. Supermodular games are simple and well behaved and are known to have pure strategy Nash equilibrium. Supermodular games are analytically appealing and they have an outstanding property, that being that many solutions yield the same predictions [51].

In [51,52] the authors attempt to utilize the theory of supermodularity to show the existence of at least one unique Nash equilibrium. In [51] the theory of supermodular games is used to analyze a game with a multidimensional strategy space, i.e. the users' uplink transmission power and data rate allocation; likewise, in [52] the theory of supermodular games is used to analyze a game with a single dimensional strategy space, i.e. the users' transmission power. Albeit, in [52] the proposed game was not supermodular and in [51] the proposed game was not supermodular without first modifying the strategy space, this is because the games did not meet the following requirement:

Definition 1 the utility f_i for player i is supermodular if and only if $\forall x, y \in S$ the following holds true [50]

$$f_i(x \wedge y) + f_i(x \vee y) \geq f_i(x) + f_i(y)$$

Remark 1 if it is submodular, then the opposite inequality holds true.

We also note that if f_i is twice differentiable, then supermodularity is equivalent to:

$$\frac{\partial^2 f_i(x)}{\partial x_i \partial x_j} \geq 0$$

for all $x \in S$ and $j \neq i$.

Remark 2 if it is submodular, then the opposite inequality holds true.

A submodular game is a game characterized by diminishing returns, i.e. adding an element to a smaller subset of S makes a bigger difference to the function values than adding it to a larger subset of S [53]. In [53] the authors utilize the theory of submodular games to show the existence of a pure Nash Equilibrium in their non-cooperative game. Additionally, submodular games exhibit the characteristic of strategic substitutes, i.e. when one player decides to increase its action, the other players follow up by lowering their action since they perceive a negative feedback from the system. We propose that our non-cooperative game displays characteristics of submodular games.

Definition 2 The non-cooperative game G is submodular, if for all the UAVs, the following conditions hold true.

1. A_d is a compact subset of an Euclidean space.
2. $U_d(b_d, \mathbf{b_{-d}})$ is smooth, submodular in b_d, and has non-increasing differences in $(b_d, \mathbf{b_{-d}})$, i.e., $\frac{\partial^2 U_d(b_d,\mathbf{b_{-d}})}{\partial b_d \partial b_i} \leq 0$.

The submodular games are characterized by strategic substitutes implying that an increase in the actions of one UAV leads the other UAVs to decrease their actions, i.e., amount of offloaded data, accordingly. In a submodular game, there always exist external equilibria: a largest element $\overline{b_d} = sup\{b_d \in A_d : BR(b_d, \mathbf{b_{-d}}) \geq b_d\}$ and a smallest element $\underline{b_d} = inf\{b_d \in A_d : BR(b_d, \mathbf{b_{-d}}) \leq b_d\}$ of the equilibrium set, where $BR(\cdot)$ denotes the UAV's $d, d \in D$ best response strategy to other UAVs' strategies.

3.2 Problem Solution

The theory of submodularity captures the UAVs data offloading problem very well, given that if a UAV increases its action, i.e. decides to offload a larger percentage of its computing task's bits, then the interference in the communication environment increases and the MEC system has to process more data. Therefore, the rest of the UAVs experience congestion in both the communication and computing environments and accordingly lower their actions, i.e. decide to offload a lower percentage of their computing task's bits.

Theorem 1 The non-cooperative game $G = [D, \{A_d\}_{d \in D}, \{U_d\}_{d \in D}]$ is submodular for all $b_d \in A_d$ and has at least one Nash Equilibrium.

Proof 1 The strategy space $A_d = [0, I_d]$ is a compact subset of an Euclidean space. The UAV's utility function $U_d(b_d, \mathbf{b_{-d}})$, as defined in Eq. 2.2, is smooth, as it has derivatives of all orders everywhere in its domain A_d. Towards showing that the utility function $U_d(b_d, \mathbf{b_{-d}})$ is submodular and has non-increasing differences in $(b_d, \mathbf{b_{-d}})$, we determine its second order partial derivative, as follows.

$$\frac{\partial^2 U_d(b_d, \mathbf{b_{-d}})}{\partial b_d \partial b_i} = -\frac{\sum\limits_{\forall s \in S} F_s}{\sum\limits_{\forall d \in D} I_d} \cdot e^{\frac{F_d}{\sum\limits_{\forall d \in D} F_d} \cdot w} (1+c) w_1$$

We conclude that $\frac{\partial^2 U_d(b_d, \mathbf{b_{-d}})}{\partial b_d \partial b_i} \leq 0$, as $1 + c \geq 0$, thus the non-cooperative game G is submodular and has at least one Nash Equilibrium, which is defined as:

$$b_d^* = \underset{b_d \in A_d}{\operatorname{argmax}} \, U_d(b_d, \mathbf{b_{-d}})$$

Thus, the UAV data offloading problem proposed is a non-cooperative, submodular game. As stated above, S-modular type games are known to have at least one Nash equilibrium and the NE can be found using greedy best response type algorithms. Therefore, described below is the best response dynamics approach used to determine the NE of the UAVs data offloading problem.

3.3 Best Response Dynamics (BRD) Approach

A best response dynamics approach is adopted in order to enable the UAVs to determine the optimal amount of their computing task's bits to offload to the MEC server. The best response algorithm will allow the UAVs' strategies to converge to the NE. Based on this, the UAVs make intelligent, human-like data offloading decisions in an autonomous matter. The UAVs best response strategy in the Euclidean space A_d is denoted as:

$$BR(b_d, \mathbf{b_{-d}}) = b_d^* = \underset{b_d \in A_d}{\operatorname{argmax}} U_d(b_d, \mathbf{b_{-d}}) \tag{3.2}$$

Theorem 2 In the non-cooperative game $G = [D, \{A_d\}_{d \in D}, \{U_d\}_{d \in D}]$, the UAVs' strategies converge to a Nash Equilibrium.

Proof 2 In order to prove that the UAVs' strategies converge to a NE, we have to prove that each UAV's best response strategy is a standard function. A function f is standard, if the following three conditions hold true.

A Positivity: $f(\mathbf{x}) > 0$;

B Monotonicity: if $\mathbf{x} \geq \mathbf{x}'$, then $f(\mathbf{x}) \geq f(\mathbf{x}')$, and

C Scalability: for all $a > 1, af(\mathbf{x}) \geq f(a\mathbf{x})$ for all $\mathbf{x} > \mathbf{0}$, where $\mathbf{x} = [x_1, \ldots, x_{|D|}]$ is a NE.

Regarding the non-cooperative game $G = [D, \{A_d\}_{d \in D}, \{U_d\}_{d \in D}]$, we can easily show that the above three conditions hold true, as follows.

A $b_d > 0$, thus $BR(b_d, \mathbf{b_{-d}}) > 0$, via Eq. 3.2;

B If $b_d \geq b_d'$, then via Eq. 3.2 we have $BR(b_d, \mathbf{b_{-d}}) \geq BR(b_d', \mathbf{b_{-d}})$, and

C For all $a > 1$, $BR(b_d, \mathbf{b_{-d}})$ is monotonous with respect to b_d in A_d, thus $aBR(b_d, \mathbf{b_{-d}}) \geq BR(ab_d, \mathbf{b_{-d}})$.

The algorithm that implements the aforementioned UAVs' best response dynamics converging to the non-cooperative game G's NE is presented in Algorithm 1. The complexity of the BRD algorithm is $O(|D|Ite)$, $Ite >> |D|$ (Chapter 5), where Ite is the total number of iterations required for the algorithm to converge to the NE.

Algorithm 1 Best Response Dynamics

```
1: Input: S, D, T_d, ρ_d, ∀d ∈ D
2: Output: Profile Strategy at NE: b_d^*
3: Initialization: ite = 0, Convergence = 0, b_d^(ite=0)
4: while Convergence == 0 do
5:     ite = ite + 1;
6:     for d=1 to D do
7:         UAV d determines b_d^{*(ite)} w.r.t. b_{-d}^{*(ite-1)} (Eq.3.2) and receives U_d^(ite)
8:     end for
9:     if b_d^{*(ite)} == b_d^{*(ite-1)} then
10:        Convergence = 1
11:    end if
12: end while
```

3.4 Reinforcement Learning Approach

As alternatives to the Best Response Dynamics approach described above, two Reinforcement Learning algorithms will be utilized, namely the Binary Log-Linear (B-logit) and Max Log-Linear (Max-logit) algorithms. These algorithms will be utilized as artificial intelligence algorithms to enable the UAVs to decide in an autonomous and distributed manner the amount of their computing task's data that each one should offload to their chosen MEC server. These approaches require no information exchange between the UAVs in order for the non-cooperative game G to converge to the NE. The Binary Log-Linear and Max Log-Linear algorithms convergence to a NE is proven in [54]. In B-logit and Max-logit algorithms, we assume that each UAV has a discrete space of strategies from which it can choose from, i.e. $b_d \in A_d = \{b_d^{min}, \ldots, b_d^{max}\}$ and initially it selects a random amount of information $b_d^{(ite=0)}$ with equal probability $Pr(b_d^{(ite=0)}) = \frac{1}{|A_d|}$. At each iteration the algorithm selects a random UAV to perform exploration and learning, while the other UAVs maintain their previous strategy. Therefore, at the ite iteration the UAV d explores an alternative amount of information $b_d'^{(ite)}$ as its new strategy with equal probability $\frac{1}{|A_d|}$; the UAV then receives a respective utility $U_d'^{(ite)}(b_d'^{(ite)}, \mathbf{b_{-d}^{(ite)}})$ associated with exploring the chosen strategy (exploration phase). At the ite iteration, UAV d updates its strategy, i.e. the amount of its computing task's data that it will offload to the MEC server, according to the following probabilistic learning rules, i.e., Eq. 3.3a and 3.3b regarding the B-logit approach, and Eq. 3.3c and 3.3d with reference to the Max-logit approach, while the rest of the UAVs maintain their previously selected strategy (learning phase).

$$Pr(b_d^{(ite)} = b_d'^{(ite)}) = \frac{e^{U_d'^{(ite)} \cdot \beta}}{e^{U_d^{(ite-1)} \cdot \beta} + e^{U_d'^{(ite)} \cdot \beta}} \tag{3.3a}$$

$$Pr(b_d^{(ite)} = b_d^{(ite-1)}) = \frac{e^{U_d^{(ite-1)} \cdot \beta}}{e^{U_d^{(ite-1)} \cdot \beta} + e^{U_d'^{(ite)} \cdot \beta}} \tag{3.3b}$$

$$Pr(b_d^{(ite)} = b_d'^{(ite)}) = \frac{e^{U_d'^{(ite)} \cdot \beta}}{max(e^{U_d^{(ite-1)} \cdot \beta}, e^{U_d'^{(ite)}})} \tag{3.3c}$$

$$Pr(b_d^{(ite)} = b_d^{(ite-1)}) = \frac{e^{U_d^{(ite-1)} \cdot \beta}}{max(e^{U_d^{(ite-1)} \cdot \beta}, e^{U_d'^{(ite)}})} \tag{3.3d}$$

where $b_d^{(ite-1)}$, $U_d^{(ite-1)}$ are the UAV's d strategy and utility at the $(ite - 1)$ iteration, respectively. The B-logit and Max-logit algorithms are presented below in Algorithm

2 and Algorithm 3. The complexity of the Max-logit/B-logit algorithm is $O(Ite')$, $Ite' >> |D|$ (Chapter 5), where Ite' is the total number of iterations required for the algorithms to converge to the NE.

Algorithm 2 B-logit

Input: S, D, T_d, ρ_d, $\forall d \in D$

Output: Profile Strategy at NE: $\mathbf{b_d^*}$

Initialization: $\beta = 1000$, $\epsilon = 10^{18}$, T, $ite = 0$, $Convergence = 0$, $\mathbf{b_d}^{(ite=0)}$

while $Convergence == 0$ **do**

  $ite = ite + 1$;

  UAV d selects $b_d^{\prime(ite)}$ *with equal probability* $\frac{1}{|A_d|}$ *, receives* $U_d^{\prime(ite)}$ *and updates* $b_d^{(ite)}$ based on Eq.3.3a, 3.3b

  The other UAVs keep their previous actions, i.e., $\mathbf{b_{-d}^{(ite)}} = \mathbf{b_{-d}^{(ite-1)}}$

  if $|(\frac{\sum_{ite=0}^{T} \sum_{d=1}^{|D|} (U_d^{(ite)})}{T} - \sum_{d=1}^{|D|} U_d^{ite})| \leq \epsilon$ **then**

    $Convergence = 1$

  end if

end while

Algorithm 3 Max-logit

Input: S, D, T_d, ρ_d, $\forall d \in D$

Output: Profile Strategy at NE: $\mathbf{b_d^*}$

Initialization: $\beta = 1000$, $\epsilon = 10^{18}$, T, $ite = 0$, $Convergence = 0$, $\mathbf{b_d}^{(ite=0)}$

while $Convergence == 0$ **do**

  $ite = ite + 1$;

  UAV d selects $b_d^{\prime(ite)}$ *with equal probability* $\frac{1}{|A_d|}$ *, receives* $U_d^{\prime(ite)}$ *and updates* $b_d^{(ite)}$ based on Eq.3.3c, 3.3d

  The other UAVs keep their previous actions, i.e., $\mathbf{b_{-d}^{(ite)}} = \mathbf{b_{-d}^{(ite-1)}}$

  if $|(\frac{\sum_{ite=0}^{T} \sum_{d=1}^{|D|} (U_d^{(ite)})}{T} - \sum_{d=1}^{|D|} U_d^{ite})| \leq \epsilon$ **then**

    $Convergence = 1$

  end if

end while

Chapter 4

MEC Server Selection Through Reinforcement Learning

This chapter introduces a reinforcement learning algorithm based on the theory of stochastic learning automata (SLA). In the SLA algorithm, the game is played once in every slot according to the mixed strategy profile of the players. In every slot, the players receive a payoff and update their strategy profile based on the payoff. If the chosen action receives a positive payoff, then the probability the player will choose the same action again increases; whereas, the probability of choosing the other actions decreases. This updating strategy is known in the literature as linear reward-inaction and is the update strategy used in the following SLA algorithm. Since the strategy update of each player solely relies on that player's individual information, the linear reward-inaction is considered to be completely distributed. For this reason, the SLA algorithm is an efficient solution for the incomplete, dynamic and uncertain information in wireless communication networks [55]. In [55] the authors further investigate the usage of the SLA algorithm in distributed wireless games; it is determined that the SLA algorithm is a powerful tool for wireless networks and can be applied to various wireless optimization problems. For this reason, we explore SLA as a means for enabling the UAVs to select and optimal MEC server as described below.

The proposed SLA algorithm will enable the UAVs to select the most beneficial MEC server to process their computing task's bits. Each MEC server will be characterized by a reputation score. This reputation score increases as the MEC server's relative computational capability increases, as well as when the utilities of the users served by the examined MEC server increase. In addition, if the MEC server's relative distance from the users decreases, the reputation score increases. The formal definition of the MEC servers' reputation score is presented in Eq. 4.1.

$$r_s = \frac{\left(\frac{F_s}{\sum\limits_{\forall s \in S} F_s} \frac{\sum\limits_{\forall d \in D} U_{d,s_d}}{\sum\limits_{\forall s \in S} \sum\limits_{\forall d \in D} U_{d,s_d}} \right)}{\frac{\sum\limits_{\forall d \in D} d_{d,s_d}}{\sum\limits_{\forall s \in S} \sum\limits_{\forall d \in D} d_{d,s_d}}} \tag{4.1}$$

where s_d denotes the MEC server that UAV d chooses to offload a portion of its computing task's data to and $d_{d,s_d}[m]$ denotes the distance of UAV d from the MEC server s_d that is serving it.

Initially, each UAV acts as an SLA gathering information from the system and learning the most beneficial MEC server to offload a part of its computing task's data for further processing, while dynamically adapting to the multi-UAVs multi-MEC servers environment. In each iteration of the SLA, each UAV selects a MEC server to offload its data in a probabilistic manner by using the following action probabilities:

$$Pr_{d,s}(t+1) = Pr_{d,s}(t) + br_s(t)(1 - Pr_{d,s}(t)), \quad s^{(t+1)} = s^{(t)} \tag{4.2a}$$

$$Pr_{d,s}(t+1) = Pr_{d,s}(t) - br_s(t)Pr_{d,s}(t), \quad s^{(t+1)} \neq s^{(t)} \tag{4.2b}$$

where b, defined as $0 < b < 1$ is a step-size parameter that determines the convergence time of the SLA algorithm. Eq. 4.2a presents the probability $Pr_{d,s}(t+1)$ of UAV d in the time slot $t+1$ of selecting the same MEC server to be served from as in time slot t, while eq. 4.2b presents the probability of a UAV to select a different MEC server than the one that was serving the UAV in the previous time slot. It is noted that as the time evolves, each UAV selects per time slot a MEC server to partially offload its data, and within the time slot, each UAV determines the NE (Chapter 3) by following any of the three alternative approaches, i.e., best response dynamics, B-logit, and Max-logit.

The algorithm that implements the aforementioned UAVs' MEC server selection is presented below in Algorithm 4. Assuming the SLA component uses the BRD algorithm for the data offloading decision-making component, which as shown in Chapter 3.3 has a complexity of $O(|D|Ite)$, then the algorithm's complexity is $O(T(|D|Ite))$, where T is the total number of time slots required for the SLA to converge. Since the total number of time slots, T, scales well with the total number of drones, $|D|$ (Chapter 5), the SLA approach is characterized by low complexity.

Algorithm 4 SLA

Input: S; D; $F_s, \forall s \in S$; $d_{d,s_d}, \forall d \in D, \forall s \in S$; $U_{d,s_d}, \forall d \in D, \forall s \in S$; b
Output: Profile Strategy at NE: MEC server $s \in S$ that each UAV $d \in D$ will be served by
Initialization: $t = 0$; $Pr_{d,s}(0) = \frac{1}{|S|}$; $Convergence = 0$
while $Convergence == 0$ **do**
  for $d = 1$ to $|D|$ **do**
    UAV d chooses a MEC server s to offload its data to based on its action probability vector $Pr_{d,s}(t) = [Pr_{d,1}, \ldots, Pr_{d,|S|}]$
  end for
  Run BRD (or Max-logit/B-logit)
  for $s = 1$ to $|S|$ **do**
    MEC s determines the corresponding reputation score r_s *(4.1) based on the UAVs that want to offload their data to it*
  end for
  for $d = 1$ to $|D|$ **do**
    for $\forall s \in S$ **do**
      if $s^{(t+1)} = s^t$ **then**
        Eq. 4.2a
      else
        Eq. 4.2b
      end if
    end for
  end for
  Check for convergence
  if $\forall d \in D$, $\exists s_d \in S : |Pr_{d,s}(t) - 1| \leq \epsilon, \epsilon \to 0$ **then**
    $Convergence = 0$
  else
    $t = t + 1$
  end if
end while

Chapter 5

Numerical Results

In this chapter, a detailed numerical evaluation of the proposed data offloading framework in a multi-UAVs multi-MEC environment is conducted. Initially, the performance evaluation focuses on the pure operation characteristics of the proposed game theoretic data offloading framework (Section 5.1), under the best response dynamics (BRD) algorithm. Afterwards, the performance evaluation of the two alternative reinforcement learning approaches (i.e., Max-logit and B-logit) to determine the optimal amount of offloaded data for each UAV, is studied in Section 5.2. Additionally, a comparative analysis of the performance of the best response dynamics approach against the Max-logit and B-logit algorithms is also presented.

In the subsequent analysis, considered is a multi-UAVs multi-MEC servers environment consisting of $|S|= 3$ MEC servers and $|D|= 80$ UAVs, where each UAV's distance is randomly and uniformly distributed in the interval $(10m, 400m)$. Also, for demonstration purposes only, the following system parameterizations are assumed:

- $F_s \in [1, 5]10^{12} CPU cycles/sec$

for each MEC server, and:

- $I_d = [20, 100] MBytes$
- $C_d = [1, 5] 10^9 CPUcycles$
- $\phi_d = C_d / I_d$
- $\rho_d = 130 W/CPUcycles$
- $w = 50, w_1 = 1, w_2 = 1.47 \cdot 10^{20}$, and $w_3 = 10^6$,
- $P^{Max} = 2W$

for each UAV. The proposed framework's evaluation was conducted via modeling and simulation and executed in a MacBook Pro Laptop, 2.5GHz Intel Core i7, with 16GB LPDDR3 available RAM.

5.1 Pure Game Theoretic Framework Operation Evaluation

Below we present the outcome of the BRD algorithm in Fig. 5.1. On the left vertical axis is the UAVs' average achieved utility and on the right vertical axis is the average amount of offloaded data to the MEC servers; this is presented as a function of the BRD algorithm's iterations (bottom horizontal axis) and the actual execution time required for the algorithm to converge to the NE (top horizontal axis). The results reveal that the BRD algorithm converges to the NE is in less than 10 iterations which corresponds to less than 1 millisecond, indicating that each UAV determines its data offloading strategy in a relatively fast manner.

Referencing the MEC server selection part of the framework, Fig. 5.2 presents the operation of the SLA algorithm, which enables the UAVs to select a MEC server

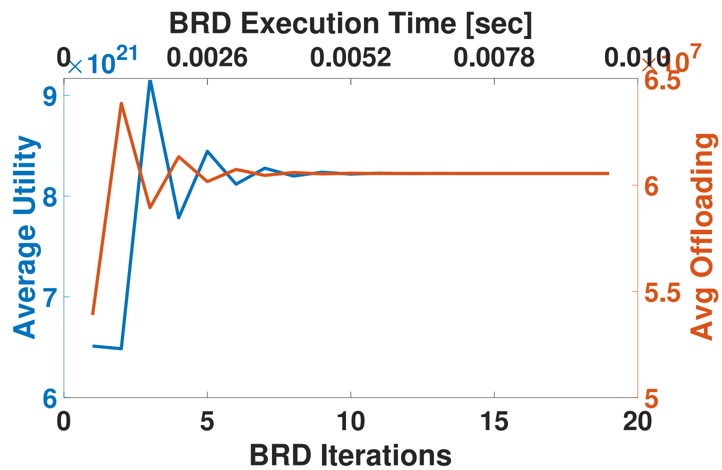

Figure 5.1: Best Response Dynamics

to offload a percentage of their computing task's bits. For the following numerical results, the SLA algorithm's learning parameter is defined as $b = 0.7$. Fig. 5.2a presents, for a randomly selected UAV, the convergence of the action probabilities towards one of the three MEC servers; Fig. 5.2a shows that the UAVs conclude to the selection of an optimal MEC server relatively fast, occurring in less than 40 iterations (equivalent to less than 1 second). Additionally, in the included subfigure a Monte Carlo analysis is performed for $10,000$ runs of the SLA algorithm for the following range of values of the learning parameter: $b = 0.1, 0.2, \ldots, 1$. From the results of this Monte Carlo analysis we conclude that as the learning parameter b increases, the UAVs do not take as much time to explore the available MEC server options, and thus converge to a selection faster, requiring less time and less iterations.

Fig. 5.2b depicts the evolution of the MEC servers' reputation score (left vertical axis) according to Eq. 4.1 and the corresponding UAVs' average action probability per MEC server (right vertical axis). From the data, we observe that the MEC server with the highest reputation score also achieves a higher average probability of being chosen to be served by a UAV over the other MEC servers; thus, the MEC server with the highest reputation score attracts more UAVs to offload their data to it. Fig. 5.2c confirms this observation; in Fig. 5.2c, the MEC server with the highest reputation score, i.e. MEC server 3, attracts more UAVs. Additionally, those UAVs that are served by MEC server 3 achieve a higher average utility than the rest. Consequently, MEC server 3 (or in general, the MEC server with the highest reputation score) will also receive an increase in the amount of data offloaded to it from the UAVs it serves compared with the other MEC servers (see Fig. 5.2d).

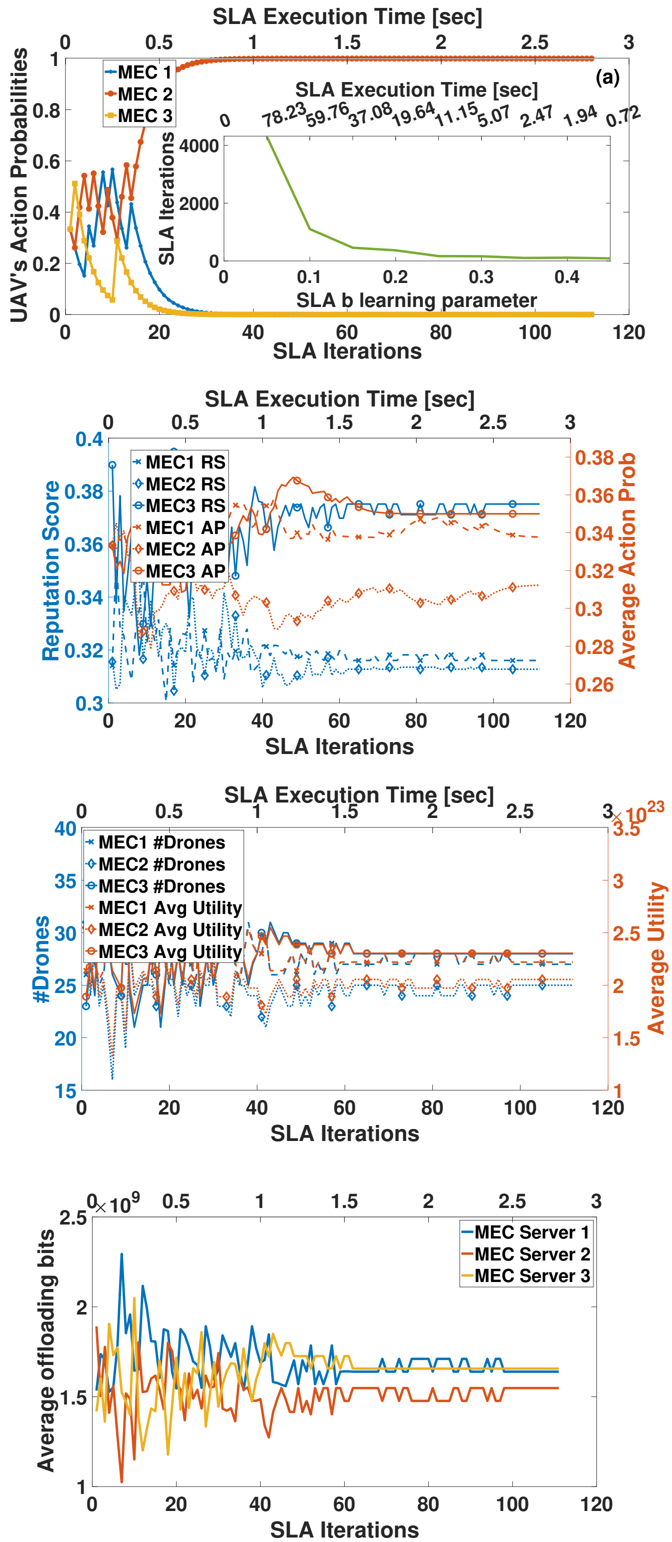

Figure 5.2: Stochastic Learning Automata: MEC Server Selection

5.2 Reinforcement Learning and Comparative Evaluation

Initially, in this section the behavior of the two reinforcement learning approaches (i.e., Max-logit and B-logit) introduced in Section 3.4 is studied and analyzed. Additionally, the convergence of these two algorithms, used as alternatives to the BRD algorithm, towards determining the optimal amount of each UAV's computing task's data to offload is also analyzed. In particular Fig. 5.3a (Fig. 5.3c) and Fig. 5.3b (Fig. 5.3d), present the UAVs' welfare i.e., summation of all the UAVs' utilities, and the UAVs' average amount of offloaded data respectively, for the Max-Logit (B-Logit) algorithm, as a function of the corresponding required iterations (bottom horizontal axis) and actual execution time (upper horizontal axis) and for different values of the learning parameter β.

Regarding the two RL approaches, the results reveal that both converge to the NE, by following the exploration and learning phases; however, the time required to converge to the NE is achieved in a slower manner than compared to the BRD algorithm, e.g. the BRD algorithm converges in milliseconds whereas the two RL approaches converge in seconds. This increased convergence time is explained by the exploration phase performed by the learning algorithms in order to learn the data offloading strategy; whereas the BRD algorithm learns the data offloading strategy by performing the optimization presented in Eq. 3.2. Moreover, it is confirmed that the Max-logit algorithm converges to the NE faster than the B-logit algorithm. Additionally, both algorithms show that for greater values of the learning parameter β, the UAVs converge to a better NE in terms of amount of data offloaded [55] (Fig. 5.3b and 5.3d). Therefore, by offloading a greater portion of their computing task's data, each UAV achieves a greater utility, and consequently, their overall welfare is also greater (as shown in Fig. 5.3a for Max-logit and 5.3c for B-logit).

Subsequently, a comparative analysis of the BRD algorithm against the aforementioned reinforcement learning paradigm, in terms of the performance of the overall proposed framework, is presented. For the comparison, we choose the Max-logit algorithm among the reinforcement learning ones, since it presented better results compared to the B-logit algorithm, as discussed above. Specifically, Fig. 5.4a presents the UAVs' amount of offloaded data at the NE as a function of the UAVs' IDs for the game-theoretic BRD algorithm and Max-logit reinforcement learning algorithm, considering different action space sizes, i.e., $10, 1,000$, and $10,000$ available actions.

The results reveal that as the number of available actions increases, the Max-logit algorithm converges to values of the amount of offloaded data closer to the BRD algorithm's values, thus, the corresponding mean square error decreases (Fig. 5.4b). In that respect the reinforcement learning approach (i.e., Max-logit) can achieve similar results as the game-theoretic approach (i.e., BRD); however, without requiring any information exchange among the UAVs, i.e., the data offloading vector of the rest of the UAVs $\mathbf{b_{-d}}$. Specifically, it is also observed that the Max-logit algorithm converges to a better NE among the available ones compared to the BRD algorithm, even for a small number of available data offloading actions. Accordingly, the UAVs achieve greater utilities under the Max-logit algorithm (Fig. 5.4c) as they offload more data to the MEC servers for further processing (Fig. 5.4a).

Moreover, Fig. 5.4d and Fig. 5.4e present the UAVs' average utility and the execution time of the BRD, Max-logit, and B-logit algorithms. The results illustrate that the UAVs achieve a greater average utility under the Max-logit algorithm, as they converge to a better NE among the available ones as explained before (Fig. 5.4a). Also, the BRD algorithm has the smallest execution time, as it practically solves a closed-form optimization problem, i.e., Eq. 3.2, and the UAVs do not invest time in the exploration phase, unlike the reinforcement learning approaches. The B-logit algorithm has the slowest execution time, as it slowly updates the action probabilities (Eq. 3.3a, 3.3b) compared to the Max-logit algorithm (Eq. 3.3c, 3.3d).

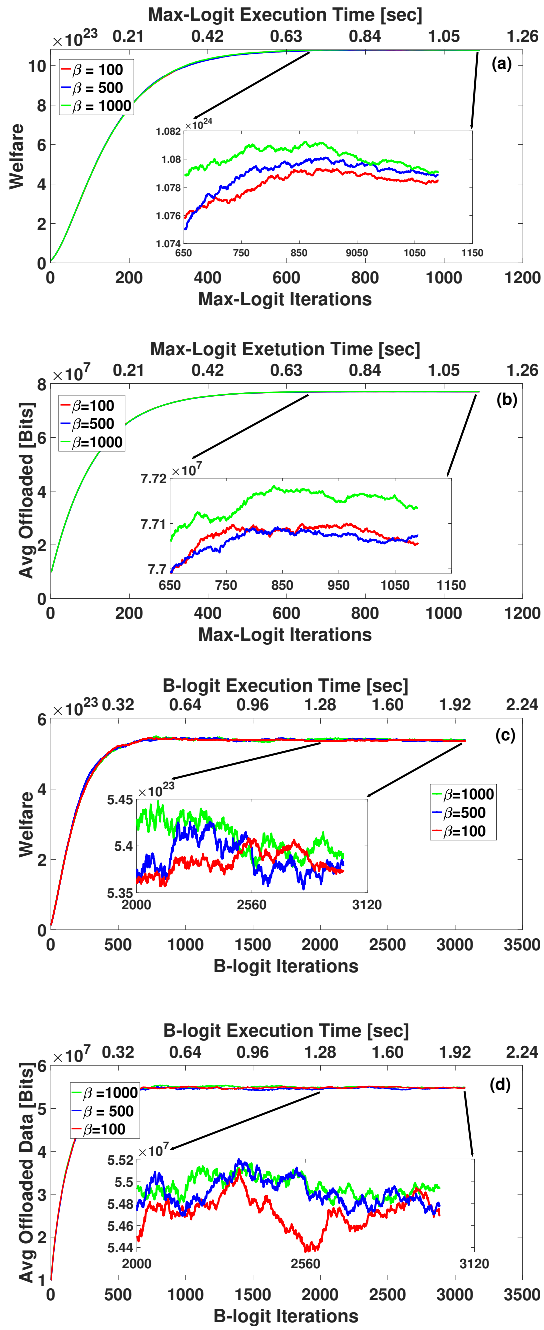

Figure 5.3: Reinforcement Learning Algorithms: Max-Logit and B-logit

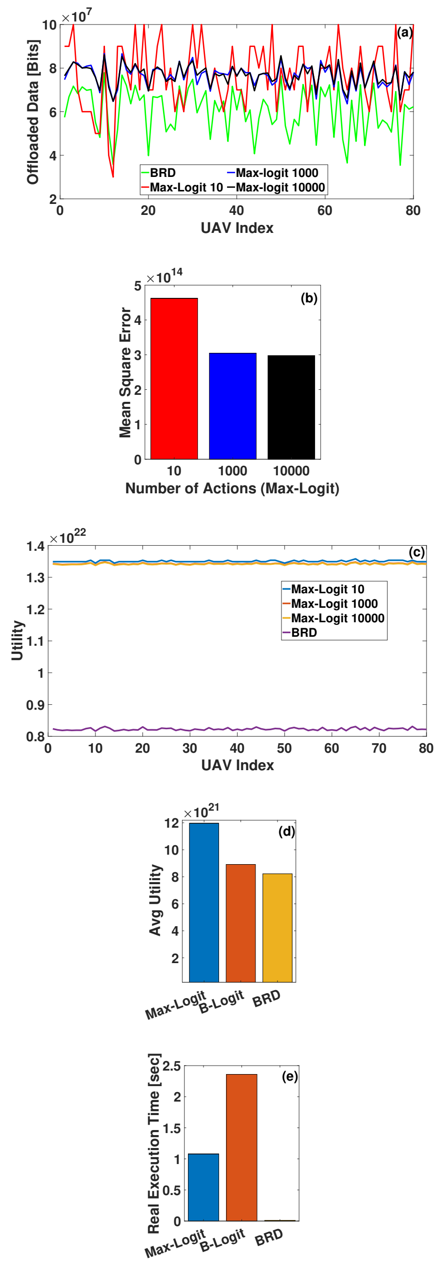

Figure 5.4: Game-theoretic Best Response Dynamics vs Reinforcement Learning Approaches

Chapter 6

Conclusions

In this work, an artificially intelligent system to support the data offloading system for UAVs in a multi-MEC server environment is devised and evaluated through the use of game theory and reinforcement learning. In particular, a non-cooperative game among the UAVs is formulated to determine the UAVs data offloading scheme to the MEC servers and the existence of at least on NE is proven. A best response dynamics framework as well as two alternative reinforcement learning algorithms were introduced towards proving the existence of a NE point for the data offloading game; additionally, a reinforcement learning algorithm based on the theory of stochastic learning automata was introduced for the purpose of autonomous MEC server selection by the UAVs.

To handle the UAVs data offloading scheme we initially introduced the BRD algorithm. This algorithm proved to converge to a NE point with respect to the UAVs average offloaded bits and corresponding utility. The BRD algorithm converged in a relatively fast manner, and the algorithm presented is characterized by a low complexity. Alternatively, two reinforcement learning algorithms, namely Max-logit and B-logit, were introduced; these algorithms have no data exchange between the

UAVs and, instead have an exploration phase to better learn the available strategy space. Both RL approaches proved to converge to an NE point; however, the cost of removing the information exchange between the UAVs is a longer convergence time. Furthermore, it is shown that the Max-logit algorithm not only converges faster than the B-logit algorithm, but also it converges to an overall better NE point with regards to the amount of offloaded data. Lastly, when comparing the BRD algorithm to the Max-logit algorithm the data shows that Max-logit tends to once again converge to a better NE point with regards to the amount of offloaded data.

To handle the MEC server selection portion of the system a reinforcement learning algorithm based on the theory of stochastic learning automata was introduced. In each time slot of the SLA either the BRD or one of the RL algorithms is run. It has been proven that under the Max-logit algorithm the UAVs achieve a better NE among those available, whereas under the BRD algorithm the convergence time is much faster. Therefore, the optimal solution for the mulit-UAV multi-MEC server environment in terms of autonomous MEC server selection and data offloading is to run Max-logit inside the SLA per time slot. The overall framework was evaluated via modeling and simulation, in terms of its efficiency and effectiveness,by studying multiple operation approaches and scenarios.

References

[1] R. Li, Z. Zhao, X. Zhou, G. Ding, Y. Chen, Z. Wang, and H. Zhang, "Intelligent 5g: When cellular networks meet artificial intelligence," *IEEE Wir. Com.*, vol. 24, no. 5, pp. 175–183, 2017.

[2] N. Hassan, S. Gillani, E. Ahmed, I. Yaqoob, and M. Imran, "The role of edge computing in internet of things," *IEEE Communications Magazine*, vol. 56, no. 11, pp. 110–115, 2018.

[3] E. E. Tsiropoulou, G. Mitsis, and S. Papavassiliou, "Interest-aware energy collection & resource management in machine to machine communications," *Ad Hoc Networks*, vol. 68, pp. 48–57, 2018.

[4] M.-A. Messous, S.-M. Senouci, H. Sedjelmaci, and S. Cherkaoui, "A game theory based efficient computation offloading in an uav network," *IEEE Trans. on Veh. Tech.*, vol. 68, no. 5, pp. 4964–4974, 2019.

[5] P. A. Apostolopoulos, E. E. Tsiropoulou, and S. Papavassiliou, "Game-theoretic learning-based qos satisfaction in autonomous mobile edge computing," in *2018 Global Information Infrastructure and Networking Symposium (GIIS)*, pp. 1–5, IEEE, 2018.

[6] Y. S. J. Z. K. B. Letaief, W. Chen and Y.-J. A. Zhang, "The roadmap to 6g – ai empowered wireless networks," *arXiv preprint arXiv:1904.11686*, 2019.

[7] V. L. X. Wang, X. Li, "Artificial intelligence-based techniques for emerging heterogeneous network: State of the arts opportunities and challenges," *IEEE Access*, vol. 3, pp. 1379–1391, 2015.

[8] J. Q. et al., "Artificial intelligence enabled networking," *IEEE Access*, vol. 3, pp. 3079–3082, 2015.

[9] D. Ross, "Game theory," in *The Stanford Encyclopedia of Philosophy* (E. N. Zalta, ed.), Metaphysics Research Lab, Stanford University, spring 2019 ed., 2019.

[10] D. Fudenberg and J. Tirole, "Game theory mit press," *Cambridge, MA*, p. 86, 1991.

[11] E. E. Tsiropoulou, G. K. Katsinis, and S. Papavassiliou, "Distributed uplink power control in multiservice wireless networks via a game theoretic approach with convex pricing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 1, pp. 61–68, 2011.

[12] T. Kastrinogiannis, E.-E. Tsiropoulou, and S. Papavassiliou, "Utility-based uplink power control in cdma wireless networks with real-time services," in *International Conference on Ad-Hoc Networks and Wireless*, pp. 307–320, Springer, Berlin, Heidelberg, 2008.

[13] E. E. Tsiropoulou, G. K. Katsinis, and S. Papavassiliou, "Utility-based power control via convex pricing for the uplink in cdma wireless networks," in *2010 European Wireless Conference (EW)*, pp. 200–206, IEEE, 2010.

[14] C. U. Saraydar, N. B. Mandayam, D. J. Goodman, *et al.*, "Efficient power control via pricing in wireless data networks," *IEEE transactions on Communications*, vol. 50, no. 2, pp. 291–303, 2002.

[15] E. E. Tsiropoulou, G. K. Katsinis, P. Vamvakas, and S. Papavassiliou, "Efficient uplink power control in multi-service two-tier femtocell networks via a game theoretic approach," in *2013 IEEE 18th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*, pp. 104–108, IEEE, 2013.

[16] E. Altman and Z. Altman, "S-modular games and power control in wireless networks," *IEEE Transactions on Automatic Control*, vol. 48, no. 5, pp. 839–842, 2003.

[17] T. Alpcan, T. Başar, R. Srikant, and E. Altman, "Cdma uplink power control as a noncooperative game," *Wireless Networks*, vol. 8, no. 6, pp. 659–670, 2002.

[18] G. Katsinis, E. E. Tsiropoulou, and S. Papavassiliou, "A game theoretic approach to the power control in d2d communications underlay cellular networks," in *2014 IEEE 19th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*, pp. 208–212, IEEE, 2014.

[19] P. Vamvakas, E. E. Tsiropoulou, M. Vomvas, and S. Papavassiliou, "Adaptive power management in wireless powered communication networks: A user-centric approach," in *2017 IEEE 38th Sarnoff Symposium*, pp. 1–6, IEEE, 2017.

[20] E. E. Tsiropoulou, A. Kapoukakis, and S. Papavassiliou, "Energy-efficient subcarrier allocation in sc-fdma wireless networks based on multilateral model of bargaining," in *2013 IFIP Networking Conference*, pp. 1–9, IEEE, 2013.

[21] M. R. Musku, A. T. Chronopoulos, D. C. Popescu, and A. Stefanescu, "A game-theoretic approach to joint rate and power control for uplink cdma communications," *IEEE Transactions on Communications*, vol. 58, no. 3, pp. 923–932, 2010.

[22] E. E. Tsiropoulou, P. Vamvakas, and S. Papavassiliou, "Joint customized price and power control for energy-efficient multi-service wireless networks via s-modular theory," *IEEE Transactions on Green Communications and Networking*, vol. 1, no. 1, pp. 17–28, 2017.

[23] E. E. Tsiropoulou, P. Vamvakas, and S. Papavassiliou, "Energy efficient uplink joint resource allocation non-cooperative game with pricing," in *2012 IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 2352–2356, IEEE, 2012.

[24] E. E. Tsiropoulou, P. Vamvakas, and S. Papavassiliou, "Joint utility-based uplink power and rate allocation in wireless networks: A non-cooperative game theoretic framework," *Physical Communication*, vol. 9, pp. 299–307, 2013.

[25] M. R. Musku, A. T. Chronopoulos, and D. C. Popescu, "Joint rate and power control with pricing," in *GLOBECOM'05. IEEE Global Telecommunications Conference, 2005.*, vol. 6, pp. 5–pp, IEEE, 2005.

[26] E. E. Tsiropoulou, P. Vamvakas, and S. Papavassiliou, "Supermodular game-based distributed joint uplink power and rate allocation in two-tier femtocell networks," *IEEE Transactions on Mobile Computing*, vol. 16, no. 9, pp. 2656–2667, 2016.

[27] P. Vamvakas, E. E. Tsiropoulou, and S. Papavassiliou, "Dynamic provider selection power resource management in competitive wireless communication markets," *Mobile Networks and Applications*, vol. 23, no. 1, pp. 86–99, 2018.

[28] E. E. Tsiropoulou, G. K. Katsinis, A. Filios, and S. 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*, pp. 114–127, Springer, Cham, 2014.

[29] X. X. Chunming Liu and D. Hu, "Multiobjective reinforcement learning: A comprehensive overview," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 45, no. 3, pp. 385–398, 2015.

[30] Y. Y. Arjun Muralidharan and Y. Mostofi, "Binary log-linear learning with stochastic communication links," in *MILCOM 2015 - 2015 IEEE Military Communications Conference*, pp. 1348–1353, IEEE, 2015.

[31] Y. Song, S. H. Wong, and K.-W. Lee, "Optimal gateway selection in multi-domain wireless networks: A potential game perspective," in *Proceedings of the 17th Annual International Conference on Mobile Computing and Networking*, pp. 325–336, ACM, 2011.

[32] J. S. S. Jason R. Marden, "Revisiting log-linear learning: Asynchrony, completeness and payoff-based implementation," *Games and Economic Behavior*, vol. 75, no. 2, pp. 788–808, 2012.

[33] P. C. Mohd. Shabbir Ali and M. Coupechoux, "Load balancing in heterogeneous networks based on distributed learning in near-potential games," *IEEE Transactions on Wireless Communications*, vol. 15, no. 7, pp. 5046–5059, 2016.

[34] Y. Xu, J. Wang, and Q. Wu., "Distributed learning of equilibria with incomplete, dynamic, and uncertain information in wireless communication networks," in *Game Theory Framework Applied to Wireless Communication Networks*, pp. 63–86, IGI Global, 2016.

[35] J. W. L. S. Yuhua Xu, Qihui Wu and A. Anpalagan, "Opportunistic spectrum access using partially overlapping channels: Graphical game and uncoupled learning," *IEEE Transactions on Communications*, vol. 61, no. 9, pp. 3906–3918, 2013.

[36] Y. B. Hajar El Hammouti, Loubna Echabbi, "Maximizing qos in heterogeneous wireless sensor networks: Memory and time considerations," in *2015 10th International Conference on Intelligent Systems: Theories and Applications (SITA)*, pp. 1–6, ACM, 2015.

[37] G. Mitsis, P. A. Apostolopoulos, E. E. Tsiropoulou, and S. Papavassiliou, "Intelligent dynamic data offloading in a competitive mobile edge computing market," *Future Internet*, vol. 11, 2019.

[38] Y. Mao, J. Zhang, S. Song, and K. B. Letaief, "Power-delay tradeoff in multi-user mobile-edge computing systems," in *IEEE Global Commu-nications Conference (GLOBECOM)*, pp. 1–6, IEEE, 2016.

[39] X. Chen, L. Jiao, W. Li, and X. Fu, "Efficient multi-user computation offloading for mobile-edge cloud computing," *IEEE/ACM Transactions on Networking*, vol. 24, no. 5, pp. 2795–2808, 2016.

[40] K. Zhang, Y. Mao, S. Leng, Q. Zhao, L. Li, X. Peng, S. M. L. Pan, and Y. Zhang, "Energy-efficient offloading for mobile edge computing in 5g heterogeneous networks," *IEEE Access*, vol. 4, pp. 5896–5907, 2016.

[41] H. Yu, Q. Wang, and S. Guo, "Energy-efficient task offloading and resource scheduling for mobile edge computing," in *2018 IEEE Inter-national Conference on Networking, Architecture and Storage (NAS)*, pp. 1–4, IEEE, 2018.

[42] T.-Y. Kan, Y. Chiang, and H.-Y. Wei, " "task offloading and resource allocation in mobile-edge computing system," in *27th IEEE Wireless and Optical Communication Conference (WOCC)*, pp. 1–4, IEEE, 2018.

[43] S. Sardellitti, G. Scutari, and S. Barbarossa, "Joint optimization of radio and computational resources for multicell mobile-edge computing," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 1, no. 2, pp. 89–103, 2015.

[44] D. Sikeridis, E. E. Tsiropoulou, M. Devetsikiotis, and S. Papavassiliou, "Wireless powered public safety iot: A uav-assisted adaptive-learning approach towards energy efficiency," *Journal of Network and Computer Applications*, vol. 123, pp. 69–79, 2018.

[45] D. Sikeridis, E. EleniTsiropoulou, M. Devetsikiotis, and S. Papavassiliou, "Self-adaptive energy efficient operation in uav-assisted public safety networks," in *2018 IEEE 19th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, pp. 1–5, IEEE, 2018.

[46] P. Vamvakas, E. E. Tsiropoulou, and S. Papavassiliou, "On the prospect of uav-assisted communications paradigm in public safety networks," in *IEEE INFOCOM WKSHPS: WCNEE 2019: Wireless Communications and Networking in Extreme Environments*, 2019.

[47] G. Fragkos, E. E. Tsiropoulou, and S. Papavassiliou, "Disaster management and information transmission decision-making in public safety systems," in *IEEE GLOBECOM*, 2019.

[48] L. Jian, Z. Li, X. Yang, W. Wu, A. Ahmad, and G. Jeon, "Combining unmanned aerial vehicles with artificial-intelligence technology for traffic-congestion recognition: Electronic eyes in the skies to spot clogged roads," *IEEE Consumer Electronics Magazine*, vol. 8, no. 3, pp. 81–86, 2019.

[49] D. M. Topkis, "Equilibrium points in nonzero-sum n-person submodular games," *SIAM J. Control Optim.*, vol. 17, no. 6, p. 773–787, 1979.

[50] E. Altman and Z. Altman, "S-modular games and power control in wireless networks," *IEEE Transactions on Automatic Control*, vol. 48, no. 5, pp. 839–842, 2003.

[51] E. E. Tsiropoulou, P. Vamvakas, and S. Papavassiliou, "Supermodular game-based distributed joint uplink power and rate allocation in two-tier femtocell networks," *IEEE Transactions on Mobile Computing*, vol. 16, no. 9, pp. 2656–2667, 2017.

[52] C. U. Saraydar, N. B. Mandayam, and D. J. Goodman, "Efficient power control via pricing in wireless data networks," *IEEE Transactions on Communications*, vol. 50, no. 2, pp. 291–303, 2002.

[53] Chengjie Wu, Y. Xu, Y. Chen, and C. Lu, "Submodular game for distributed application allocation in shared sensor networks," in *2012 Proceedings IEEE INFOCOM*, pp. 127–135, 2012.

[54] Y. Song, S. H. Wong, and K.-W. Lee, "Optimal gateway selection in multi-domain wireless networks: A potential game perspective," in *Proceedings of the 17th annual international conference on Mobile Comp. and Net.*, pp. 325–336, ACM, 2011.

[55] Y. Xu, J. Wang, and Q. Wu, "Distributed learning of equilibria with incomplete, dynamic, and uncertain information in wireless communication networks," in *Game Theory Framework Applied to Wireless Communication Networks*, pp. 63–86, IGI Global, 2016.