Artificial Intelligence Enabled Distributed Edge Computing for Internet of Things Applications

Georgios Fragkos, Eirini Eleni Tsiropoulou
Dept. of Electrical and Computer Engineering
University of New Mexico
Albuquerque, NM, USA
gfragkos@unm.edu, eirini@unm.edu

Symeon Papavassiliou
School of Electrical and Computer Engineering
National Technical University of Athens
Athens, Greece
papavass@mail.ntua.gr

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

***Index Terms*—Edge Computing; Game Theory; Reinforcement Learning; Internet of Things;**

I. Introduction

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

Traditionally, Cloud-based solutions were utilized to deal with the computational, storage, and networking challenges imposed by the large streams of data. However, Cloud computing faces great challenges related to energy consumption, latency, and security, all of them being critical aspects for sensor-driven applications. On the other hand, the emerging edge computing paradigm proposes shifting the pendulum away from the traditional Cloud computing model, towards a distributed infrastructure model at the edge of the network, by offering computational resources closer to the physical location of data producers/consumers [2].

Nevertheless, in order to fully unleash the autonomous decision-making capabilities of the edge devices and users, while exploiting the distributed edge computing capabilities, there is an urgent need to push the AI frontiers to the system's edge [3]. AI mechanisms allow 5G networks to be predictive and proactive, which is essential in making the 5G vision conceivable. Nowadays, AI has extended its domain and strength, beyond the traditional machine learning approaches, by being founded on multidisciplinary techniques, such as control theory, computationally light reinforcement learning techniques, game theory, optimization theory, and meta-heuristics [4].

Motivated by the aforementioned observations and arguments, in this paper, we propose an artificial intelligence enabled multi-access edge computing (MEC) framework, supported by computing-equipped Unmanned Aerial Vehicles (UAVs) to facilitate IoT applications. The key problem at hand is to properly explore and learn the environment and the interdependence among the IoT nodes actions, so that to determine their optimal data offloading strategies to an UAV-mounted MEC server, while accounting for the IoT nodes' communication and computation overhead,

A. Related Work & Motivation

Distributed edge computing has been immensely supported by the adoption of UAV-mounted MEC servers [5], due to the UAVs' unique characteristics, i.e., fast, flexible, and effortless deployment, mobility, maneuverability, line-of-sight communication, etc. The problem of minimizing the IoT devices' communication and computation energy consumption and the UAVs' flying energy utilization is studied in [6], by jointly optimizing the devices' data offloading, transmission power, and the UAVs' trajectory. In [7], the problem of partial data offloading from the IoT devices to ground or UAV-mounted MEC servers is studied in order the devices to satisfy their minimum Quality

2325-2944/20/$31.00 ©2020 IEEE
DOI 10.1109/DCOSS49796.2020.00077

of Service (QoS) prerequisites, by adopting the novel concept of Satisfaction Equilibrium. n [8], the authors target at UAVs energy-efficiency, by jointly optimizing their hovering time, and the devices' scheduling and data offloading, while considering the constraints of the UAVs' computation capability and the devices' QoS constraints. A similar problem is studied in [9] by exploiting the uplink and downlink communication among the devices and the UAVs in terms of data offloading/receiving data respectively, while guaranteeing the energy efficient operation of the overall system. In [10], the problem of jointly optimizing the devices' association, transmission power, and data offloading to the UAVs, as well as the UAVs' trajectory is studied, aiming at minimizing the overall power consumption in the system.

A techno-economics approach is presented in [11], where the UAVs charge fees the users for the computation services that they offer to them. Also, the UAVs charge their battery over a microwave station and the authors target at maximizing the UAVs' utility by optimizing their trajectories and the data offloading process. In [12], the UAVs act as cache and edge computing nodes, and two sequentially solved optimization problems are considered, to minimize the communication and computation delay and maximize the energy efficiency of the system. In [13], the UAVs act both as MEC servers and as wireless power transfer nodes charging the IoT devices. The problem of maximizing the UAVs' computation rate is examined under the UAVs' energy provision and speed constraints.

However, despite the significant research work and advances achieved by the aforementioned research efforts, the problem of the IoT devices' distributed and autonomous decision-making with respect to their data offloading strategies, has not yet been fully exploited, especially under the the prism of artificial intelligence. In this paper, a field of IoT devices is considered supporting latency and energy sensitive IoT applications. Accordingly each IoT device has the option to execute its computation task either locally or offload part of it to a UAV-mounted MEC server, by considering the joint optimization of the involved communication and computation overhead. The focus of this paper is placed on the design of an artificial intelligence-enabled framework that drives the strategic decision of optimal data offloading to the available UAV-mounted MEC server, founded on on the power and principles of *Game Theory* and *Reinforcement Learning*.

B. Contributions & Outline

The key technical contributions of this research work are summarized as follows.

- The IoT devices' communication, computation and energy overhead due to data offloading is properly captured and modeled (Section II), while based on this, the utility of each device by offloading and processing its computation task's data to the UAV-mounted MEC server is reflected in representative functions (Section III).
- A non-cooperative game is formulated among the IoT devices aiming at maximizing their own utility at every timeslot, while considering the experienced communication and computation time overhead, from offloading and processing their data at the UAV (Section IV-A). This process enables the devices to learn from history, scrutinize the performance of other nodes, and adjust their own behavior accordingly. The existence of at least one Pure Nash Equilibrium (PNE) point is shown by proving that the game is submodular (Section IV-B). A best response dynamics approach is introduced that converges to a PNE (Section IV-C).
- Different operation points (i.e. offloading strategies) are obtained and studied, based on alternative reinforcement learning approaches (i.e. gradient ascent, log-linear, and Q-learning algorithms), which explore and learn the environment towards determining the users' stable data offloading strategies (Section V). The operation and convergence strategies realized by the various reinforcement algorithms, are critically compared against each other and against the corresponding one at the PNE point (Section VI).

II. Communication & Computation Overhead

A distributed edge computing system is considered consisting of a set of IoT devices $D = \{1, \ldots, d, \ldots, |D|\}$ spread in an area $x[m] \times y[m]$ and a UAV-mounted MEC server hovering above the area. Each IoT device has a computation task $\mathcal{T}_d^{(t)}$ to be completed at timeslot t, which is defined as $\mathcal{T}_d^{(t)} = (I_d^{(t)}, \phi_d^{(t)})$, where $I_d^{(t)}[bits]$ denotes the total amount of data of the IoT device's computation task, and the parameter $\phi_d^{(t)}[\frac{CPU-Cycles}{bit}]$ represents the computation intensity of the device's task (i.e., a higher value of $\phi_d^{(t)}$ expresses a more computing demanding application). At each timeslot t, each IoT device offloads part of its computation task's data to the UAV-mounted MEC server for further processing, while aiming at minimizing its experienced communication and computation latency and energy cost. The IoT device's d set of data offloading strategies at timeslot t is denoted as $A_d^{(t)} = \{a_{d,min}^{(t)}, \ldots, a_{d,j}^{(t)}, \ldots, a_{d,max}^{(t)}\}$, where $a_{d,j}^{(t)} \in [0, 1]$ is a percentage of the overall amount of the device's computation task's data.

Moreover, a non-orthogonal multiple access (NOMA)-based wireless communication environment is considered to enable each IoT device to offload its computation task's data $a_{d,j}^{(t)} \cdot I_d^{(t)}[bits]$ to the UAV at each timeslot t. Also, the Successive Interference Cancellation (SIC) technique is implemented at the UAV to improve the interference management in the congested IoT environment [14]. Each IoT device's d uplink data rate to the UAV-mounted

MEC server at each timeslot t is calculated through the Shannon's formula, as follows.

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

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

The UAV-mounted MEC server is assumed to have a computation capability $F_{UAV}[\frac{CPU-Cycles}{sec}]$ that is shared among the IoT devices to process their offloaded data. Also, the UAV can process in parallel a total amount of data $B_{UAV}[bits]$ at each timeslot. Based on the above, the time overhead that the IoT device d experiences at timeslot t by offloading $a_{d,j}^{(t)} \cdot I_d^{(t)}$ is given as follows [15]:

$$O_{time,d}^{(t)} = \frac{a_{d,j}^{(t)} \cdot I_d^{(t)}}{R_d^{(t)}} + \frac{\phi_d^{(t)} \cdot a_{d,j}^{(t)} \cdot I_d^{(t)}}{[1 - \frac{\sum\limits_{d' \neq d} a_{d',j'}^{(t)} \cdot I_{d'}^{(t)}}{B_{UAV}}] \cdot F_{UAV}} \quad (2)$$

The first term of Eq.2 represents the communication time overhead that the IoT device experiences to offload its data to the UAV, while the second term captures the experienced computation time overhead. Also, as it is observed by the denominator of the second term in Eq.2, each IoT device exploits only a portion of the UAV's computation capability, as the latter is shared in a fair manner among the IoT devices.

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

$$O_{energy,d}^{(t)} = \frac{a_{d,j}^{(t)} \cdot I_d^{(t)}}{R_d^{(t)}} \cdot p_d^{(t)} \quad (3)$$

The duration of a timeslot t is assumed $T[sec]$ and the energy availability of an IoT device d at timeslot t is $e_d^{(t)}[J]$. Based on Eq.2, 3, the total normalized overhead that the IoT device d experiences at timeslot t is given as follows.

$$O_d^{(t)} = \frac{O_{time,d}^{(t)}}{T} + \frac{O_{energy,d}^{(t)}}{e_d^{(t)}} \quad (4)$$

III. IoT Devices Utilities

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

$$U_d^{(t)}(a_{d,j}^{(t)}, \boldsymbol{a}_{-\boldsymbol{d,j}}^{(\boldsymbol{t})}) = b \cdot e^{\frac{a_{d,j}^{(t)}}{\sum_{\forall d' \neq d, d' \in D} a_{d',j'}^{(t)}}} - c \cdot e^{O_d^{(t)}} \quad (5)$$

where $\boldsymbol{a}_{-\boldsymbol{d,j}}^{(\boldsymbol{t})}$ is the data offloading strategy vector of all the devices residing in the examined system except for the IoT device d. Also, the weights $b, c \in [0, 1]$ are configurable parameters representing how much the IoT device weighs the satisfaction that it receives by processing its data at the UAV (first term of Eq.5), as compared to the corresponding cost to perform this action (second term of Eq.5). Moreover, given that small changes in the devices' data offloading strategies can dramatically influence the stable operation of the distributed edge computing system due to the large number of devices, we have adopted the exponential form to capture the devices' satisfaction and cost tradeoffs and trends in Eq.5.

IV. Game-theoretic Edge Distributed Computing

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

A. Problem Formulation

Each IoT device aims at maximizing its perceived utility, as expressed in Eq.5, at each timeslot in order to improve its perceived benefit from offloading and processing its data at the UAV-mounted MEC server, while mitigating its personal cost, as expressed by its experienced overhead (Eq.4). Thus, the corresponding optimization problem for each IoT device, is expressed as the maximization of each IoT device's utility, as follows.

$$\max U_d^{(t)}(a_{d,j}^{(t)}, \boldsymbol{a}_{-\boldsymbol{d,j}}^{(\boldsymbol{t})}) = b \cdot e^{\frac{a_{d,j}^{(t)}}{\sum_{\forall d' \neq d, d' \in D} a_{d',j'}^{(t)}}} - c \cdot e^{O_d^{(t)}} \quad (6)$$
$$\text{s.t. } a_{d,j}^{(t)} \in A_d^{(t)}$$

Based on the maximization problem in Eq.6, we observe that the IoT devices' data offloading strategies are interdependent, and the devices demonstrate competitive behavior in terms of exploiting the UAV's computing capabilities. Thus, the utility maximization problem in Eq.6 is confronted as a non-cooperative game among the IoT devices. Let $G = [D, \{A_d^{(t)}\}_{d \in D}, \{U_d^{(t)}\}_{d \in D}]$ denote the

Distributed Data Offloading (DDO) game played among the IoT device's at each timeslot t, where as mentioned before D is the set of IoT devices, $A_d^{(t)}$ is the data offloading strategy set of each device $d \in D$, and $U_d^{(t)}$ denotes the device's utility.

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

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

Based on Definition 1, we conclude that the existence of a PNE in the DDO game guarantees the stable operation of the distributed edge computing system, while the IoT devices maximize their perceived utility. On the other hand, if the DDO game does not have at least one PNE, that is translated to an unsteady and unstable state of the examined system.

B. Problem Solution

The theroy of S-modular games is adopted in order to show the existence of at least one PNE for the DDO game [16]. Specifically, we show that the DDO game is submodular, which means that when an IoT device tends to offload a large amount of data to the UAV-mounted MEC server, the rest of the devices follow the exact opposite philosophy, i.e., they become more conservative in terms of their data offloading, as the MEC server is congested with tasks. Thus, in general a submodular game is characterized by strategic substitutes and has at least one PNE [16], [17]. Considering the DDO game with strategy space $A_d^{(t)}$, we can prove the following theorem.

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

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

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

Proof. Towards proving that the DDO game is submodular, we consider that the IoT device can partition its task in any feasible set of data and offload them to the UAV-mounted MEC server. Thus, the strategy space $A_d^{(t)} = (0, 1]$ is continuous and a compact subset of the Euclidean space and $U_d^{(t)}$ is a smooth function. Also we have: $\frac{\partial^2 U_d^{(t)}}{\partial a_{d,j}^{(t)} \cdot \partial a_{d',j'}^{(t)}} = b \cdot \lambda - c \cdot \mu$ where we set $\lambda = \frac{a_{d,j}^{(t)}}{e^{\sum_{\forall d' \neq d, d' \in D} a_{d',j'}^{(t)}}} \cdot (\frac{-1}{(\sum_{\forall d' \neq d, d' \in D} a_{d',j'}^{(t)})^2} + \frac{-1}{(\sum_{\forall d' \neq d, d' \in D} a_{d',j'}^{(t)})^3} \cdot a_{d,j}^{(t)})$ and $\mu = e^{O_d^{(t)}} \cdot (\frac{\phi_d^{(t)} \cdot I_d^{(t)} \cdot \frac{I_{d'}^{(t)}}{B_{UAV}}}{\sum_{a_{d',j'}^{(t)} \cdot I_{d'}^{(t)}} [1 - \frac{d' \neq d}{B_{UAV}}]^2 \cdot F_{UAV} * T}) \cdot (1 + O_d^{(t)})$. Thus, we observe that $\lambda < 0$ and $\mu > 0$. Therefore, we conclude that $\frac{\partial^2 U_d^{(t)}}{\partial a_{d,j}^{(t)} \cdot \partial a_{d',j'}^{(t)}} < 0$ and the DDO game is submodular. ■

Consequently, taking into account that a submodular game has a non-empty set of Pure Nash Equilibrium points [16], [17], we conclude that the DDO game has at least one PNE $\boldsymbol{a^{(t)*}} = (a_{1,j'}^{(t)*}, \ldots, a_{|D|,j'}^{(t)*}), a_{d,j}^{(t)*})$.

C. Best Response Dynamics

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

The best response strategy of each IoT device is defined as follows.

$$BR_d(\boldsymbol{a_{-d,j}^{(t)*}}) = a_{d,j}^{(t)*} = \arg\max_{a_{d,j}^{(t)} \in A_d^{(t)}} U_d^{(t)}(a_{d,j}^{(t)}, \boldsymbol{a_{-d,j}^{(t)}}) \quad (7)$$

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

V. Reinforcement Learning-enabled Edge Distributed Computing

In this section, an artificial intelligence approach is introduced based on reinforcement learning algorithms to enable the IoT devices to determine their stable data offloading strategies, while mitigating their experienced overhead. The need for adopting these learning approaches versus the game-theoretic model (as expressed via the BRD framework), arises in several realistic cases including the ones where: a) the devices are not fully aware of the closed-form solution (Eq. 7), and/or b) the devices' data

Algorithm 1 Asynchronous BRD Algorithm

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

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

A. Gradient Ascent Learning

In the gradient ascent reinforcement learning approach, the IoT devices act as Learning Automata (LA) and they learn their environment by performing gradient updates of their perceived utility. Each device's data offloading decisions are characterized by an action probability vector $\mathbf{P_d^{(ite)}} = [P^{(ite)}_{a^{(t)}_{d,min}}, \ldots, P^{(ite)}_{a^{(t)}_{d,j}}, \ldots, P^{(ite)}_{a^{(t)}_{d,max}}]$. At each iteration of the gradient ascent algorithm, each device probabilistically chooses its potential data offloading strategy. The IoT devices make their stable data offloading decision, if $P^{(ite)}_{a^{(t)}_{d,j}} \geq P_{thres}, \forall d \in D$, where P_{thres} is a threshold value of the action probability. The most commonly applied gradient ascent learning algorithm is called Linear Reward-Inaction (LRI) and the corresponding action probability updating rule is given as follows [18].

$$P^{(ite+1)}_{a^{(t)}_{d,j}} = P^{(ite)}_{a^{(t)}_{d,j}} + \eta[\hat{U_d^{(t)}}]^{(ite)}(1 - P^{(ite)}_{a^{(t)}_{d,j}}), \quad if \quad a^{(t)}_{d,j}|_{ite} = a^{(t)}_{d,j}|_{ite+1} \tag{8a}$$

$$P^{(ite+1)}_{a^{(t)}_{d,j}} = P^{(ite)}_{a^{(t)}_{d,j}} - \eta[\hat{U_d^{(t)}}]^{(ite)} P^{(ite)}_{a^{(t)}_{d,j}}, \quad if \quad a^{(t)}_{d,j}|_{ite} \neq a^{(t)}_{d,j}|_{ite+1} \tag{8b}$$

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

B. Log-Linear Learning

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

$$P^{(ite+1)}_{a^{(t)}_{d,j}} = \frac{e^{[U_d^{(t)}]'^{(ite)} \cdot \beta}}{e^{[U_d^{(t)}]'^{(ite)} \cdot \beta} + e^{[U_d^{(t)}]^{(ite)} \cdot \beta}}, \quad if \quad a^{(t)}_{d,j}|_{ite+1} = a^{(t)}_{d,j'}|_{ite} \tag{9a}$$

$$P^{(ite+1)}_{a^{(t)}_{d,j}} = \frac{e^{[U_d^{(t)}]^{(ite)} \cdot \beta}}{e^{[U_d^{(t)}]'^{(ite)} \cdot \beta} + e^{[U_d^{(t)}]^{(ite)} \cdot \beta}}, \quad if \quad a^{(t)}_{d,j}|_{ite+1} \neq a^{(t)}_{d,j'}|_{ite} \tag{9b}$$

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

C. Q-Learning

An alternative reinforcement learning approach, known as stateless Q-Learning, is studied in this subsection. The stateless Q-Learning utilizes the stochastic approximation methods in order to allow the IoT devices to explore and learn their environment by following a Markov Decision Process (MDP) policy, thus converging eventually to their stable data offloading decisions. Specifically, each IoT device d preserves an action values vector $Q_d^{(ite)}(\boldsymbol{a}) = [Q^{(ite)}_{a^{(t)}_{d,min}}, \ldots, Q^{(ite)}_{a^{(t)}_{d,j}}, \ldots, Q^{(ite)}_{a^{(t)}_{d,max}}]$, where $Q^{(ite)}_{a^{(t)}_{d,j}}$ denotes the

estimated value of that action $a_{d,j}^{(t)}$ up to the iteration ite, i.e., it depicts the expected utility $U_d^{(t,ite)}$ given that $a_{d,j}^{(t)}$ is selected:

$$Q_{a_{d,j}^{(t)}}^{(ite)} \cong \mathbb{E}[U_d^{(t,ite)} | a_{d,j}^{(t)}|_{ite}] \tag{10}$$

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

$$Q_{a_{d,j}^{(t)}}^{(ite)} = Q_{a_{d,j}^{(t)}}^{(ite)} + \theta \cdot (U_d^{(t,ite)} - Q_{a_{d,j}^{(t)}}^{(ite)}) \tag{11}$$

where $\theta \in (0,1]$ is the learning parameter. Since each IoT device selects an offloading strategy at each iteration ite, we introduce the widely used action selection rule known as the greedy approach. According to the greedy rule, the IoT devices select the offloading strategies with the highest expected utility (Eq.12), thus they only exploit the knowledge that is acquired up to the iteration ite.

$$a_{d,j}^{(t)}|_{ite+1} = \underset{a_{d,j}^{(t)} \in A_d^{(t)}}{\arg\max}\ Q_d^{(ite)}(\boldsymbol{a}) \tag{12}$$

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

VI. Numerical Results

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

We consider an environment consisting of $|D| = 250$ IoT devices, where each IoT device's distance from the UAV-mounted MEC server is randomly and uniformly distributed in the interval $(10m, 400m)$. The simulation parameters are as follows: $I_d^{(t)} \in [20,100] MBytes$, $C_d^{(t)} \in [1,5] \cdot 10^9 CPUcycles$, $\phi_d^{(t)} = \frac{C_d^{(t)}}{I_d^{(t)}}$, $p_d^{(t)} \in [1.2,2] Watts$, $W = 5MHz$, $b = 0.74$, $c = 0.0043$, $B_{UAV} \geq \sum_{d \in D} I_d^{(t)}$ and $F_{UAV} = 15 \cdot 10^9 \frac{CPUcycles}{sec}$. Unless otherwise explicitly stated, we consider $a_{d,min}^{(t)} \in (0,0.2]$, $a_{d,max}^{(t)} \in [0.8,1.0]$ with an intermediate step of 0.05, $\eta = 0.3$, $\beta = 1000$ and $\theta = 0.6$. The proposed framework's evaluation was conducted via modeling and simulation and was executed in a MacBook Pro Laptop, 2.5GHz Intel Core i7, with 16GB LPDDR3 available RAM.

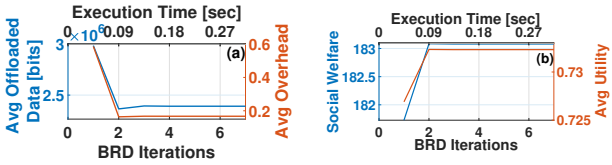

Fig. 1: Best Response Dynamics

A. Pure Operation Performance

In this subsection, we examine the operation performance of the proposed artificial framework under the game-theoretic and the reinforcement learning models, in terms of: the IoT devices' data offloading strategies, the corresponding experienced overhead and utility, the overall system's achieved social welfare, as well as the required iterations and time (execution time) for convergence.

In particular, Fig.1a presents the IoT devices' average offloaded data to the UAV and the corresponding experienced overhead as a function of the BRD algorithm's iterations and real execution time (lower and upper horizontal axis respectively). The results reveal that the BRD algorithm converges fast to a PNE (i.e., practically in less than 4 iterations, equivalent to 0.18 sec). Also, the IoT devices converge to a PNE, where they experience low average overhead (Fig.1a) and high levels of utility (Fig. 1b). Moreover, by studying the BRD framework from the system's perspective, we observe that at the PNE high levels of social welfare are obtained (Fig.1b).

Fig.2a presents the convergence of the data offloading strategies of one indicative IoT device to a stable data offloading decision following the LRI algorithm. It is observed that the devices' data offloading converge to a stable decision in less than 100 iterations i.e., 0.32 sec. Also, Fig. 2b, 2c present the convergence of the IoT devices' average offloaded data, overhead, and utility, as well as the system's social welfare. The results show that the IoT devices learn in a distributed manner their surrounding environment and they strategically decide their data offloading strategies in order to achieve low overhead and high utility, while collectively enjoy high levels of social welfare. Furthermore, the results presented in Fig.2d, reveal that for increasing values of the learning parameter η, the devices learn faster their environment and make a data offloading decision. However, this comes at the cost of lower achieved utility, as they underexplore their available data offloading decisions.

Fig.3a-3d examine the behavior of the BLLL algorithm, for different values of the learning parameter β, as a function of the iterations and the real execution time. The results show that the BLLL algorithm converges to the PNE with high probability, bearing however the cost of longer convergence time. Thus, the IoT devices converge close to the PNE and they achieve high utility levels (Fig.3b), and low overhead (Fig.3d), while intelligently deciding their data offloading strategies (Fig.3c). Furthermore, the system converges to high levels of social welfare

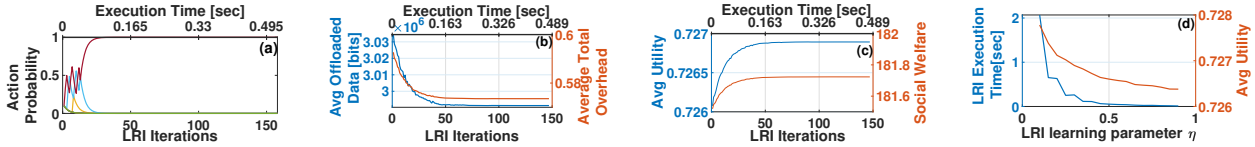

Fig. 2: Gradient Ascent Reinforcement Learning Framework

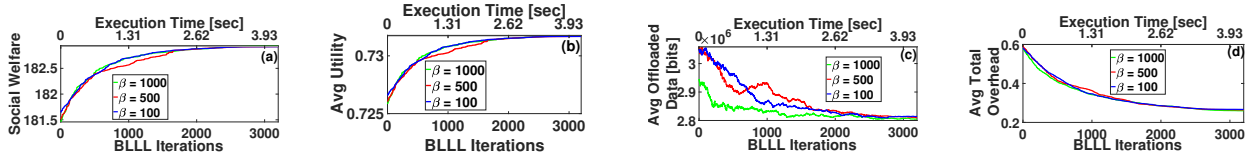

Fig. 3: Log-Linear Reinforcement Learning Framework

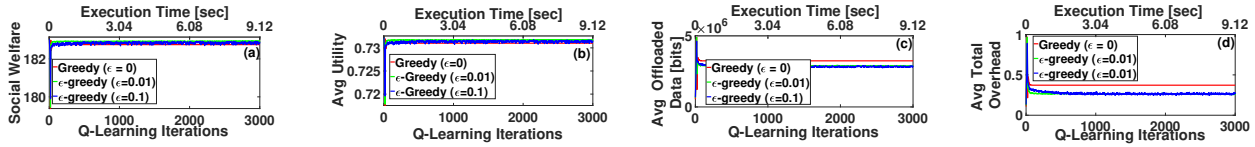

Fig. 4: Q-Learning Framework

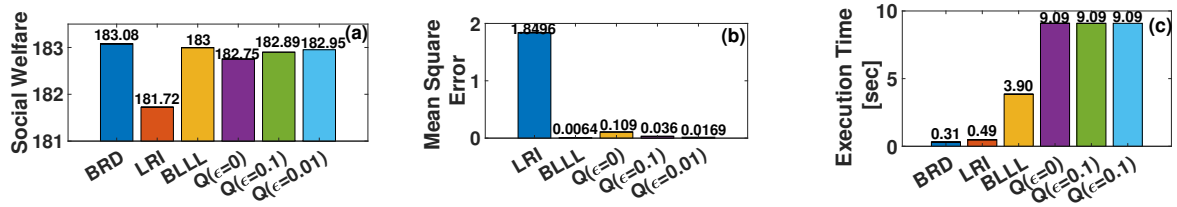

Fig. 5: Comparative Evaluation of the Reinforcement Learning Methods

(Fig.3a). Moreover, it is observed that better results are achieved for higher values of the learning parameter β.

Similarly, Fig.4a-4d present the corresponding operation performance of the Q-learning approach, i.e., both the greedy and the ϵ-greedy. The results reveal that the Q-learning algorithms converge to stable data offloading decisions for all the IoT devices (Fig.4c) achieving high utilities (Fig.4b), low overhead (Fig.4d), and high social welfare values (Fig.4a). It is also observed that the ϵ-greedy algorithm by allowing with small probability ($\epsilon = 0.01$) the IoT devices to explore other data offloading strategies than the ones that maximize the expected utilities, achieve the best results among the different Q-learning implementations. This is due to the fact that the IoT devices can explore alternative actions compared to the greedy Q-learning algorithm ($\epsilon = 0$) where they myopically choose the strategies that offer them the maximum expected utility. On the other hand, if the devices overexplore alternative strategies, i.e., $\epsilon = 0.1$, they deviate from good outcomes, being "lost" in the exploration phase.

B. Comparative Evaluation

In this subsection, a comparative evaluation among the examined learning models (i.e. game theoretic model and reinforcement learning ones) utilized to determine the IoT devices' data offloading strategies is performed. Fig.5a-5c present the system's social welfare, the social welfare's mean square error with respect to the the BRD model, and the execution time of all the examined algorithms, respectively. The results reveal that the game-theoretic model - as reflected by the BRD algorithm - illustrates the best results, both in terms of achieved social welfare and execution time. Then, the BLLL algorithm achieves the highest social welfare among all the reinforcement learning algorithms, given its inherent attribute to converge to a PNE with high probability. On the other hand, the LRI approach, given its simplistic action update rule (Eq.8a,8b) converges fast (Fig.5c) to a stable data offloading vector for all the IoT devices, while sacrificing the achieved welfare (Fig.5a). The Q-Learning approaches, i.e., $\epsilon = 0, 0.01, 0.1$ illustrate similar execution time (Fig.5c) and high levels of social welfare (Fig.5a) with the BRD algorithm's PNE outcome. In a nutshell, based on the results in Fig.5b, we observe that the smallest mean square error of the social welfare with respect to the BRD algorithm's outcome is achieved by the BLLL algorithm and then by the ϵ-greedy Q-learning algorithms with $\epsilon = 0.01$ and $\epsilon = 0.1$, respectively.

C. Discussion on Learning Methods Applicability

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

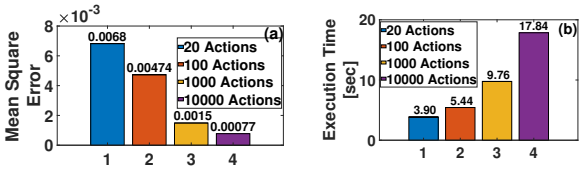

Fig. 6: Evaluation for various strategy space sizes

demonstrated the best results among all the examined reinforcement learning frameworks. Fig.6a presents the mean square error of the BLLL algorithm's achieved social welfare compared to the outcome of the BRD algorithm for 20, 100, 1,000, 10,000 data offloading strategies, while Fig.6b shows the corresponding execution time of the BLLL algorithm. The results illustrate that as the devices' strategy space increases, the achieved social welfare by the BLLL algorithm approaches the corresponding one by the BRD algorithm, at the cost of increased execution time.

Based on the results provided in the latter two subsections, we observe that, the game-theoretic BRD algorithm converges to better results both from the devices' and the system's perspective, primarily due to the use of the closed-form used to determine the PNE (Eq. 7). Nevertheless, this requires that the devices are aware of the closed-form solution or can extrapolate it, which bears additional overhead. The reinforcement learning algorithms on the other hand, eliminate this assumption, by enabling the devices to learn their environment without having a priori knowledge of the optimal strategy rule. Last but not least, it should be noted that the reinforcement learning approaches can be better applied in realistic cases where the devices' strategy space is not continuous as considered in the game-theoretic model (i.e the devices may arbitrarily select any percentage of their data to offload), but instead the devices are allowed to select their data offloading strategies from a discrete predefined strategy space.

VII. Conclusions

In this paper, an artificial intelligence-enabled distributed edge computing framework is proposed, by exploiting the computing capabilities of a UAV-mounted MEC server. The communication and computation overhead experienced by the IoT devices is modeled, and appropriate utility functions are designed for the IoT devices to measure their satisfaction from offloading their computation tasks. A non-cooperative game is formulated among the IoT devices and its PNE, i.e., devices' optimal data offloading strategies, is determined following the theory of submodular games. Alternative reinforcement learning algorithms are adopted, i.e., gradient ascent, log-linear, and Q-learning, to determine the devices' stable data offloading strategies. Detailed numerical results are presented that demonstrate the operational characteristics and performance of the different models and algorithms, while they are compared against each other. Part of our future work is to extend and evaluate the presented framework, while considering a multi-UAV-mounted servers setup, where the IoT devices can exploit the different computation choices of the environment.

Acknowledgment

The research of Mr. Fragkos and Dr. Tsiropoulou was supported under the NSF CRII-1849739. This research work was supported by the Hellenic Foundation for Research and Innovation (H.F.R.I.) under the "First Call for H.F.R.I. Research Projects to support Faculty members and Researchers and the procurement of high-cost research equipment grant" (Project Number: HFRI-FM17-2436).

References

[1] N. Hassan, S. Gillani, E. Ahmed, I. Yaqoob, and M. Imran, "The role of edge computing in internet of things," *IEEE Communications Magazine*, no. 99, pp. 1–6, 2018.
[2] M. Satyanarayanan, "The emergence of edge computing," *Computer*, vol. 50, no. 1, pp. 30–39, 2017.
[3] Y. Dai, D. Xu, S. Maharjan, G. Qiao, and Y. Zhang, "Artificial intelligence empowered edge computing and caching for internet of vehicles," *IEEE Wirel. Comm.*, vol. 26, no. 3, pp. 12–18, 2019.
[4] 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.
[5] P. A. Apostolopoulos, E. E. Tsiropoulou, and S. Papavassiliou, "Cognitive data offloading in mobile edge computing for internet of things," *IEEE Access*, vol. 8, pp. 55 736–55 749, 2020.
[6] T. Zhang, Y. Xu, J. Loo, D. Yang, and L. Xiao, "Joint computation and communication design for uav-assisted mobile edge computing in iot," *IEEE Trans. on Ind. Inform.*, pp. 1–1, 2019.
[7] P. A. Apostolopoulos, M. Torres, and E. E. Tsiropoulou, "Satisfaction-aware data offloading in surveillance systems," in *14th Workshop on Challenged Networks*, 2019, pp. 21–26.
[8] Y. Du, K. Wang, K. Yang, and G. Zhang, "Energy-efficient resource allocation in uav based mec system for iot devices," in *2018 IEEE GLOBECOM*, 2018, pp. 1–6.
[9] H. Guo and J. Liu, "Uav-enhanced intelligent offloading for internet of things at the edge," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 4, pp. 2737–2746, 2020.
[10] Z. Yang, C. Pan, K. Wang, and M. Shikh-Bahaei, "Energy efficient resource allocation in uav-enabled mobile edge computing networks," *IEEE Tran. on Wir. Com.*, vol. 18, no. 9, pp. 4576–4589, 2019.
[11] Y. Liu, M. Qiu, J. Hu, and H. Yu, "Incentive uav-enabled mobile edge computing based on microwave power transmission," *IEEE Access*, vol. 8, pp. 28 584–28 593, 2020.
[12] Z. Tan, H. Qu, J. Zhao, S. Zhou, and W. Wang, "Uav-aided edge/fog computing in smart iot community for social augmented reality," *IEEE Internet of Things Journal*, pp. 1–1, 2020.
[13] F. Zhou, Y. Wu, R. Q. Hu, and Y. Qian, "Computation rate maximization in uav-enabled wireless-powered mobile-edge computing systems," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 9, pp. 1927–1941, 2018.
[14] G. Fragkos, E. E. Tsiropoulou, and S. Papavassiliou, "Disaster management and information transmission decision-making in public safety systems," in *IEEE GLOBECOM*, 2019, pp. 1–6.
[15] P. A. Apostolopoulos, E. E. Tsiropoulou, and S. Papavassiliou, "Risk-aware data offloading in multi-server multi-access edge computing environment," *IEEE/ACM Transactions on Networking*, pp. 1–14, 2020.
[16] Y. Zhang and M. Guizani, *Game theory for wireless communications and networking*. CRC press, 2011.
[17] 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.
[18] G. Fragkos, P. A. Apostolopoulos, and E. E. Tsiropoulou, "Escape: Evacuation strategy through clustering and autonomous operation in public safety systems," *Future Internet*, vol. 11, no. 1, p. 20, 2019.