Artificial Intelligence Empowered UAVs Data Offloading in Mobile Edge Computing

Georgios Fragkos, Nicholas Kemp, Eirini Eleni Tsiropoulou
Dept. of Electrical and Computer Engineering
University of New Mexico
Albuquerque, NM, USA
gfragkos@unm.edu, nkemp@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*—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 by 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 tradeoffs 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.**

***Index Terms*—UAV Data Offloading, Mobile Edge Computing, Reinforcement Learning, Game Theory**

I. Introduction

The rise and evolution of 5G networks, as well as the latest release of 6G white paper setting the research challenges for wireless intelligence [1], have focused the interest of the research community on the communication, computing, and control of billions of connected devices, mainly placed on the scene by the explosion of the Internet of Things (IoT). Within this setting, multiple heterogeneous devices with diverse computational and communication capabilities are expected to exchange and process large amount of data in an autonomous manner. Artificial Intelligence (AI) has emerged as a powerful tool to support devices' autonomous human-like decision-making, while being founded on and supported by multi-disciplinary techniques, such as machine learning, control theory, game theory, optimization theory, and meta-heuristics [2].

At the same time, owing to the increasing computational demands of the vast number of devices connected over the Internet, Mobile Edge Computing (MEC) - representing the practice of processing data near the edge of the network [3] - is gaining significant momentum to efficiently handle the corresponding loads, while meeting the devices' Quality of Service (QoS) requirements in terms of delay, latency, and energy efficiency. Furthermore, lately, there has been heavy investment in the development of Unmanned Aerial Vehicles (UAVs) and multi-UAVs systems that can collaborate and complete missions more efficiently. Emerging technologies, including 5G networks, have significant potential on UAVs equipped with cameras and sensors in delivering IoT services, requiring the execution of computationally intensive tasks [4]. In such cases, MEC arises as a powerful tool to support their operation.

Motivated by the aforementioned arguments and observations, in this paper, we propose 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*.

A. Related Work

The problem of users' data offloading to a MEC server has been studied in [5] jointly with the interference management problem in wireless cellular networks by formulating and solving the sequential optimization problems of computation offloading decision, physical resource block allocation, and MEC computation resource allocation. A variation of this problem is studied in [6] by jointly optimizing the above metrics, as well as the content caching strategy of the MEC servers, in one holistic optimization problem that is transformed to a convex one in order to be solved. A reinforcement learning approach (Q-learning) is introduced in [7] to jointly optimize the users' offloading decision and the computational resource allocation towards minimizing the sum cost of users' delay and energy consumption. A minimization problem of users' energy consumption is introduced in [8] consisting of the energy cost for transmitting and processing data. The problem of data offloading to a MEC server is formulated as a non-cooperative game among vehicles in [9] targeting at

The research of Mr. Georgios Fragkos and Dr. Eirini Eleni Tsiropoulou was conducted as part of the NSF CRII-1849739.

reducing the latency of data offloading and the existence of a Nash Eqiulibrium (NE) is shown.

Focusing on the UAVs' data offloading to a MEC server, the authors in [4] formulate a multi-nature strategy (i.e., energy consumption, time delay, and computation cost) non-cooperative game among the UAVs, while they prove the NE's existence and propose a distributed algorithm to determine the UAVs' strategies at the NE. This work is extended in [10] to minimize a combination of energy overhead and delay for each UAV. The authors in [11] minimize the UAV's mission completion time by jointly optimizing its trajectory and the computation offloading to the ground base stations' MEC servers, while considering the UAV's maximum speed constraint and the MEC servers' computation capacity capabilities.

B. 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 [12]. 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 [2].

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 (Section II).
2. Based on the theory of submodular games, artificial intelligence is embodied in the decision of the optimal data offloading of each UAV (Section III-A). 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 III-B).
3. Towards each UAV determining the NE in an autonomous manner, three algorithms are proposed: (i) Best Response Dynamics (Section III-C), (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 III-D).
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 (Section IV).
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 (Section V). Finally, Section VI concludes the paper.

II. System Model & UAV's Utility Function

We consider the communication and computing environment of a system consisting of $|S|$ MEC servers and $|D|$ UAVs, where their sets are denoted as $S = \{1, \ldots, s, \ldots, |S|\}$ and $D = \{1, \ldots, d, \ldots, |D|\}$, respectively. Each UAV $d \in D$ selects one MEC server $s \in S$ to offload either part or all of its data in order the UAV's computation task to be processed, while the rest of the computation task's data are processed locally at the UAV. Each UAV has a computation task $T_d = (I_d, C_d, \phi_d)$, where $I_d[bits]$ and $C_d[CPUcycles]$ denote the computation total input bits and the total number of CPU cycles required to accomplish the computation task T_d, respectively. The parameter $\phi_d[\frac{CPUcycles}{bits}]$ expresses the computation complexity of the task requested by the UAV, and its value depends on the nature of the application, i.e., a higher ϕ_d expresses a more computationally intensive task. Each MEC server $s \in S$ has a computation capability $F_s[\frac{CPUcycles}{sec}]$ to process all the UAVs' offloaded data. Accordingly, each UAV decides in an autonomous and distributed manner to offload $b_d, b_d \in A_d = [0, I_d]$ amount of data to a selected MEC server, while the rest of the computation task's data, i.e., $(I_d - b_d)$, are processed locally. The UAV's local computation capability is denoted by $F_d[\frac{CPUcycles}{sec}]$, while its local power consumption to process the (remaining) data locally is $\rho_d[\frac{Watts}{CPUcycles}]$. We consider that each UAV has a fixed maximum power P^{Max} to transmit its data to the MEC server that selects to be served from.

A holistic utility function is introduced for each UAV to capture its perceived QoS prerequisites' satisfaction by processing its data in the selected MEC server. The UAV's utility function is formulated as follows.

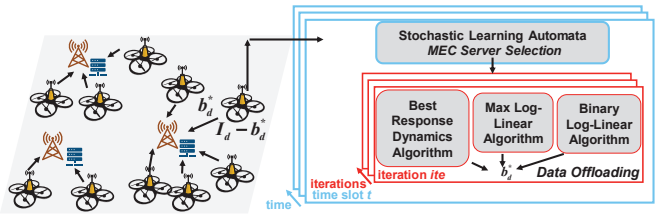

Fig. 1: Artificial Intelligence Empowered UAVs Data Offloading Framework in Mobile Edge Computing

$$
\begin{aligned}
U_d(b_d, \mathbf{b_{-d}}) = w_1 b_d \Big(\sum_{\forall s \in S} F_s - \frac{\sum_{\forall s \in S} F_s}{\sum_{\forall d \in D} I_d} e^{\frac{wF_d}{\sum_{\forall d \in D} F_d}} \sum_{\forall i \neq d} b_i \\
- \frac{\sum_{\forall s \in S} F_s}{\sum_{\forall d \in D} I_d} b_d \Big) - \frac{w_2 P^{Max} b_d}{I_d} - \frac{w_2 P^{Max} c}{I_d} \sum_{\forall i \neq d} b_i \\
- w_3 (I_d - b_d) \phi_d \rho_d + w_1 c \sum_{\forall s \in S} F_s \sum_{\forall i \neq d} b_i \\
- w_1 c \sum_{\forall i \neq d} b_i \Big[\frac{\sum_{\forall s \in S} F_s}{\sum_{\forall d \in D} I_d} \Big(\sum_{\forall i \neq d} b_i \Big) + \frac{b_d \sum_{\forall s \in S} F_s}{\sum_{\forall d \in D} I_d} e^{\frac{wF_d}{\sum_{\forall d \in D} F_d}} \Big]
\end{aligned}
\tag{1}
$$

where w, w_1, w_2 and w_3 are positive constants acting as weight parameters and selected appropriately to ensure the same order of magnitude of the individual considered terms in the UAV's utility. The first term of the UAV's utility expresses its perceived satisfaction from offloading its computation task to the selected MEC server. Specifically, the more data it offloads to the MEC server, the more it is satisfied, as it will save its personal resources by not processing them locally. However, its utility also depends on the computation capabilities of the overall MEC system and the amount of data that the rest of the UAVs offload to the MEC servers for further processing. The second term of the UAV's utility expresses the UAV's transmission cost, while the third term shows the robustness of the MEC system observed via the amount of bits that the rest of the UAVs offload ($-1 < c < 0$ is a negative constant). Specifically, if the rest of the UAVs offload a large amount of data to the MEC servers, this is a positive feedback for the examined UAV that the MEC system is robust. The fourth term of the UAV's utility expresses the UAV's local computing cost to process the remaining data locally. The fifth term of Eq. 1 also acts as a positive feedback to the UAV, by observing the amount of data that the rest of the UAVs offload relatively to the computation capabilities of the overall MEC system, while finally, the last term of Eq. 1 captures the cost that the UAV experiences by the exploitation of the MEC system's computation capabilities by itself and the rest of the UAVs.

The proposed AI-empowered UAVs' data offloading model in a MEC environment is depicted in Fig. 1. Initially, each UAV acts as an SLA (Section IV) and selects a MEC server to partially offload its data for further processing at each time slot. Within the duration of each time slot, each UAV determines its data offloading strategy by participating in a non-cooperative game with the rest of the UAVs (Section III). Towards determining the Nash Equilibrium of the non-cooperative game, three algorithms have been proposed: the best response dynamics (BRD) and two reinforcement learning algorithms, namely the Max-logit and B-logit. The overall loop of MEC selection by the UAVs and determining the data offloading strategies of the UAVs is repeated iteratively as time evolves.

III. Artificial Intelligence Empowered UAVs' Data Offloading

A. Data Offloading: An S-Modular Game Perspective

Each UAV acts as an artificial intelligent node making decisions to which MEC server to offload part of its computing task data for further processing. In this section, we present a non-cooperative game-theoretic approach based on the theory of submodular games in order to enable the UAVs to decide the optimal amount of data to offload to a MEC server, by presenting a human-like behavior. The process of the MEC server selection by the UAVs will be presented in Section IV.

A non-cooperative game among the UAVs is formulated $G = [D, \{A_d\}_{d \in D}, \{U_d\}_{d \in D}]$, where as mentioned 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) $\mathbf{b}^* = [\mathbf{b}^*_1, \ldots, \mathbf{b}^*_{|\mathbf{D}|}]$ (denoting the amount of data that each UAV offloads), which is a stable point for the overall examined system of the multi-UAVs and the multi-MEC servers. At the NE, each UAV offloads the amount of bits to the selected MEC server in order to maximize its utility function, as follows.

$$
\begin{aligned}
&\max_{b_d \in A_d} U_d(b_d, \mathbf{b_{-d}}), \qquad \forall d \in D \\
&s.t. \qquad 0 \leq b_d \leq I_d
\end{aligned}
\tag{2}
$$

Towards proving the existence of at least one NE of the non-cooperative game G, as solution of the maximization problem (2), the theory of submodular games has been adopted.

Definition 1: The non-cooperative game G is submodular, if for all the UAVs, the following conditions hold true.
A. A_d is a compact subset of an Euclidean space.
B. $U_d(b_d, \mathbf{b_{-d}})$ 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.

B. Problem Solution

The theory of submodular games captures very well the UAVs data offloading problem given that if a UAV increases its action, i.e., offloads a large amount of data, then the interference in the communication environment increases and the MEC system has to process more data. Thus, the rest of the UAVs experience the "congestion" in both the communication and computing environment and accordingly decrease their actions, i.e., offload a lower amount of data.

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: 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. 1, 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.

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

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: $b_d^* = \text{argmax}_{b_d\in A_d} U_d(b_d, \mathbf{b_{-d}})$. ■

C. Best Response Dynamics (BRD) Approach

Towards enabling the UAVs to determine the amount of data that they should offload to the MEC server in order their strategies to converge to the NE, the best response dynamics approach is adopted. Based on the latter, the UAVs make intelligent data offloading decisions in an autonomous manner. Let us denote the UAV's best response strategy in the Euclidean space A_d, as below.

$$BR(b_d, \mathbf{b_{-d}}) = b_d^* = \underset{b_d\in A_d}{\text{argmax}}\, U_d(b_d, \mathbf{b_{-d}}) \quad (3)$$

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: 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 ;
B. If $b_d \geq b'_d$, then via Eq. 3 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's G NE is presented in Algorithm 1. The complexity of the BRD algorithm is $O(|D|Ite)$, $Ite >> |D|$ (Section V), where Ite is the total number of iterations in order the algorithm to converge to the NE.

D. Reinforcement Learning Approach

As alternatives to the best response dynamics approach described above, we utilize two artificial intelligent algorithms, namely the Binary Log-Linear (B-logit) and the Max Log-Linear (Max-logit) algorithms, in order each UAV to decide in an autonomous and distributed manner the amount of data that it should offload to the MEC server. These approaches require no information exchange among the UAVs to converge to the NE of the formulated non-cooperative game and their convergence to a NE is proven in [13]. 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 every iteration, one UAV is selected randomly to conduct exploration and learning. Therefore, at the *ite* iteration the UAV d explores an alternative amount of information $b'^{(ite)}_d$ as its new strategy with equal probability $\frac{1}{|A_d|}$, receiving thus a respective utility $U'^{(ite)}_d(b'^{(ite)}_d, \mathbf{b^{(ite)}_{-d}})$ (exploration phase). At the *ite* iteration, UAV d updates its strategy, i.e., the amount of information that it will offload to the MEC server, according to the following probabilistic learning rule, i.e., Eq. 4a, 4b regarding the B-logit approach, and Eq. 4c, 4d with refrence to the Max-logit approach, while the rest UAVs maintain their previously selected actions unchanged (learning phase).

$$Pr(b_d^{(ite)} = b_d^{'(ite)}) = \frac{e^{U_d^{'(ite)}\cdot\beta}}{e^{U_d^{(ite-1)}\cdot\beta} + e^{U_d^{'(ite)}\cdot\beta}} \quad (4a)$$

Algorithm 1 Best Response Dynamics

```
Input: S, D, T_d, ρ_d, ∀d ∈ D
Output: Profile Strategy at NE: b_d^*
Initialization: ite = 0, Convergence = 0, b_d^(ite=0)
while Convergence == 0 do
    ite = ite + 1;
    for d=1 to D do
        UAV d determines b_d^*(ite) w.r.t. b_-d^*(ite-1) (Eq.3)
        and receives U_d^(ite)
    end for
    if b_d^*(ite) == b_d^*(ite-1) then
        Convergence = 1
    end if
end while
```

$$Pr(b_d^{(ite)} = b_d^{(ite-1)}) = \frac{e^{U_d^{(ite-1)}\cdot\beta}}{e^{U_d^{(ite-1)}\cdot\beta} + e^{U_d^{'(ite)}\cdot\beta}} \quad (4b)$$

$$Pr(b_d^{(ite)} = b_d^{'(ite)}) = \frac{e^{U_d^{'(ite)}\cdot\beta}}{max(e^{U_d^{(ite-1)}\cdot\beta}, e^{U_d^{'(ite)}})} \quad (4c)$$

$$Pr(b_d^{(ite)} = b_d^{(ite-1)}) = \frac{e^{U_d^{(ite-1)}\cdot\beta}}{max(e^{U_d^{(ite-1)}\cdot\beta}, e^{U_d^{'(ite)}})} \quad (4d)$$

where $b_d^{(ite-1)}$, $U_d^{(ite-1)}$ are the UAV's d strategy and utility at the $(ite-1)$ iteration, respectively. Due to the space limitation, the B-logit and Max-logit algorithms are jointly presented in Algorithm 2. The complexity of the Max-logit/B-logit algorithm is $O(Ite')$, $Ite' >> |D|$ (Section V), where Ite' is the total number of iterations in order the algorithms to converge to the NE.

IV. MEC Server Selection Through Reinforcement Learning

In this section, we introduce a reinforcement learning approach based on the theory of stochastic learning automata (SLA) to enable each UAV to select the most beneficial MEC server to process its data. Each MEC server is characterized by a reputation score, which increases as the MEC server's relative computation capability and the utilities of the users served by the examined MEC server increase, and if the MEC server's relative distance from the users decrease. The formal definition of the MEC server's reputation score is provided in Eq. 5.

$$r_s = \Big(\frac{F_s}{\sum\limits_{\forall s\in S} F_s}\frac{\sum\limits_{\forall d\in D} U_{d,s_d}}{\sum\limits_{\forall s\in S}\sum\limits_{\forall d\in D} U_{d,s_d}}\Big) \Big/ \frac{\sum\limits_{\forall d\in D} d_{d,s_d}}{\sum\limits_{\forall s\in S}\sum\limits_{\forall d\in D} d_{d,s_d}} \quad (5)$$

where s_d denotes the MEC server that the UAV d offloads its data and $d_{d,s_d}[m]$ denotes the distance of the UAV d from the MEC server s_d that it is served from.

Each UAV acts as an SLA and learns the most beneficial MEC server to offload its data in order to be processed, while dynamically adapting to the changes of the multi-UAVs multi-MEC servers environment. Each UAV selects a MEC server to offload its data in a probabilistic manner by using the following action probabilities.

$$Pr_{d,s}(t+1) = Pr_{d,s}(t) + br_s(t)(1 - Pr_{d,s}(t)), \quad s^{(t+1)} = s^{(t)} \quad (6a)$$

$$Pr_{d,s}(t+1) = Pr_{d,s}(t) - br_s(t)Pr_{d,s}(t), \quad s^{(t+1)} \neq s^{(t)} \quad (6b)$$

where $b, 0 < b < 1$ is a step-size parameter that controls the convergence time of the SLA algorithm. Eq. 6a presents the probability $Pr_{d,s}(t+1)$ of the UAV d in the time slot $t+1$ to select the same MEC server to be served from as in time slot t, while Eq. 6b shows 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 (Section III) by following any of the three alternative approaches, i.e., best response dynamics, B-logit, and Max-logit.

V. Numerical Results

In this section, a detailed numerical evaluation of the proposed data offloading framework in a multi-UAVs multi-MEC environment is conducted. The performance evaluation initially focuses on the pure operation characteristics of the proposed game theoretic data offloading framework (Section V-A), under the best response dynamics (BRD) algorithm. Subsequently, the performance 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 V-B. Additionally, a comparative analysis of the performance of the best response dynamics against these two reinforcement learning approaches is also presented. In the following, we considered a multi-UAVs multi-MEC servers environment consisting of $|S|$= 3 MEC servers and $|D|$= 80 UAVs, where each UAV's distance from each MEC server is randomly and uniformly distributed in the interval $(10m, 400m)$. Also, for demonstration only purposes, we have assumed the following system parameterization: $F_s \in [1,5]10^{12} CPUcycles/sec$, $I_d = [20,100] MBytes$, $C_d = [1,5]10^9 CPUcycles$, $\phi_d = C_d/I_d$, $\rho_d = 130W/CPUcycles$, $w = 50, w_1 = 1, w_2 = 1.47\cdot 10^{20}$, $w_3 = 10^6$, and $P^{Max} = 2W$ 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.

A. Pure Game Theoretic Framework Operation Evaluation

Fig. 2 presents the UAV's average achieved utility (left vertical axis) and the average amount of offloaded data to the MEC servers (right vertical axis), as a function of the BRD algorithm's iterations (bottom horizontal axis) and

Algorithm 2 B-logit (Max-logit)

1: **Input:** S, D, T_d, ρ_d, $\forall d \in D$
2: **Output:** Profile Strategy at NE: $\mathbf{b_d^*}$
3: **Initialization:** $\beta = 1000$, $\epsilon = 10^{18}$, T, $ite = 0$, $Convergence = 0$, $\mathbf{b_d}^{(ite=0)}$
4: **while** $Convergence == 0$ **do**
5: $ite = ite + 1$;
6: *UAV d selects* $b_d^{'(ite)}$ *with equal probability* $\frac{1}{|A_d|}$, *receives* $U_d^{'(ite)}$ *and updates* $b_d^{(ite)}$ based on Eq.4a, 4b (Eq.4c, 4d)
7: *The other UAVs keep their previous actions, i.e.,* $\mathbf{b_{-d}^{(ite)}} = \mathbf{b_{-d}^{(ite-1)}}$
8: **if** $|(\frac{\sum_{ite=0}^{T}\sum_{d=1}^{|D|}(U_d^{(ite)})}{T} - \sum_{d=1}^{|D|} U_d^{ite})| \leq \epsilon$ **then**
9: $Convergence = 1$
10: **end if**
11: **end while**

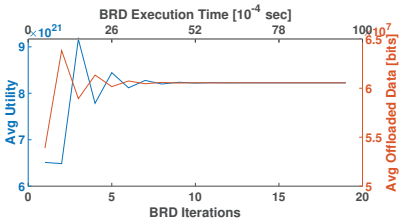

Fig. 2: Best Response Dynamics

the actual execution time required for convergence to the NE. The results reveal that the BRD algorithm converges to the NE in less than 10 iterations which corresponds to less than 1 msec, indicating that each UAV determines its data offloading strategy in a fast manner.

With reference to the MEC server selection component of our framework, in Fig. 3 we present the operation of the SLA algorithm, which enables the UAVs to select a MEC server to offload their data. For the following numerical results, we consider the SLA algorithm's learning parameter $b = 0.7$. The convergence of the action probabilities to the three MEC servers for an indicative UAV is presented in Fig. 3a showing that the UAVs conclude to the selection of a MEC server relatively fast (requires less than one second), while in the included subfigure a Monte Carlo analysis is performed for $10,000$ runs of the SLA algorithm for each value of the learning parameter $b = 0.1, 0.2, \ldots, 1$. The results demonstrate that as the learning parameter b increases, the UAVs explore less the available options of MEC servers, making a faster decision and requiring fewer iterations for convergence. Fig. 3b depicts the evolution of the MEC servers' reputation score (left vertical axis) according to Eq. 5, and the corresponding UAVs' average action probability per MEC server (right vertical axis). It is observed that the MEC server with the higher reputation score achieves a higher average probability to attract more UAVs to offload their data to it. This is confirmed in Fig. 3c where the MEC server with the highest reputation score, i.e., MEC server 3, attracts more UAVs, while those UAVs achieve higher average utility. Consequently, MEC server 3 receives increased offloaded data from the UAVs (Fig. 3d) compared with the other servers.

B. Reinforcement Learning and Comparative Evaluation

In this section, we initially study and analyze the convergence and behavior of the two alternative (with reference to the BRD algorithm) reinforcement learning approaches (i.e., Max-logit and B-logit) as they were introduced in Section III. D, towards determining the optimal amount of offloaded data for each UAV. In particular Fig. 4a (Fig. 4c) and Fig. 4b (Fig. 4d), 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 both reinforcement learning algorithms, the results reveal that both reinforcement learning algorithms converge to the NE, by following the exploration and the learning phases, however this is achieved at a slower manner compared to the BRD algorithm, i.e., order of magnitude of sec compared to msec. The latter phenomenon is observed as the learning algorithms perform the exploration phase in order to learn the data offloading strategy, while the BRD algorithm determines it by performing the optimization presented in Eq. 3. Moreover, it is confirmed that the Max-logit algorithm converges faster than the B-logit to the NE, while for greater values of the learning parameter β, the UAVs converge to a better NE in terms of the amount of offloaded data [14] (Fig. 4b and 4d). Therefore, by offloading a greater amount of data to the MEC servers for further processing, they achieve greater individual UAV utilities, and consequently, their overall welfare is larger (as shown in Fig. 4a and 4c respectively).

Subsequently, a comparative analysis of the game-theoretic 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. 5a 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. 5b). In that respect the reinforcement learning approach (i.e., Max-logit) can achieve similar results with 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. 5c) as they offload more data to the MEC servers for further processing (Fig. 5a).

Moreover, Fig. 5d and Fig. 5e 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 the greater average utility under the Max-logit algorithm, as they converge to a better NE among the available ones as explained before (Fig. 5a). Also, the BRD algorithm has the smallest execution time, as it practically solves a closed-form optimization problem, i.e., Eq. 3, and

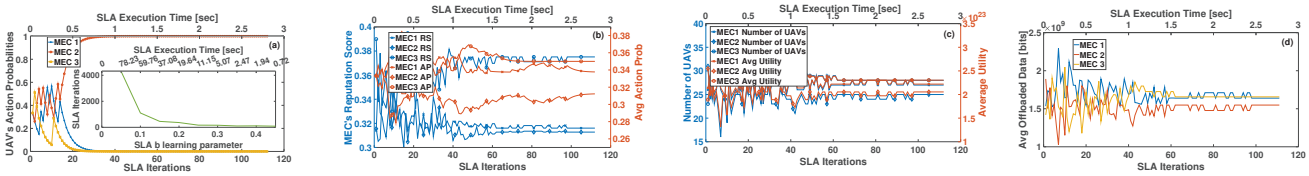

Fig. 3: Stochastic Learning Automata: MEC Server Selection

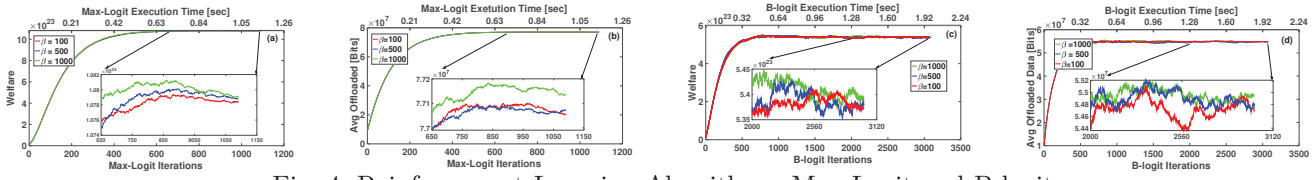

Fig. 4: Reinforcement Learning Algorithms: Max-Logit and B-logit

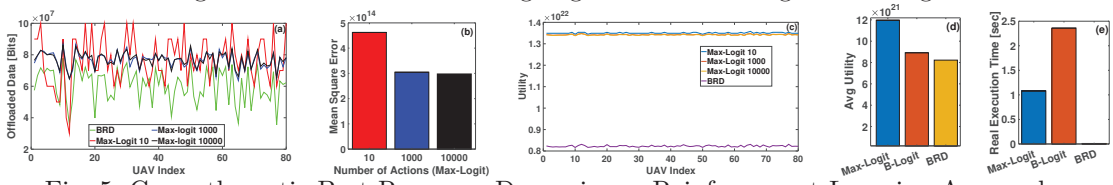

Fig. 5: Game-theoretic Best Response Dynamics vs Reinforcement Learning Approaches

the UAVs do not invest time in the exploration phase, as it happens in any of the reinforcement learning approaches. The B-logit algorithm has the slowest execution time, as it slowly updates the action probabilities (Eq. 4a, 4b) compared to the Max-logit algorithm (Eq. 4c, 4d).

VI. Conclusions

In this paper, an artificial intelligence enabled mechanism to support the UAVs' data offloading in a multi-MEC servers environment, by exploiting the power of game theory and reinforcement learning, is devised and evaluated. In particular, a non-cooperative game among the UAVs is formulated to determine the UAVs' data offloading to the MEC servers and the existence of at least one NE is proven. A best response dynamics framework is initially introduced that is shown to converge to an NE point, while two alternative reinforcement learning algorithms are presented that also achieve to converge to the NE point without requiring to exchange any information among the UAVs, at the cost however of lower convergence speed. Moreover, a reinforcement learning algorithm is proposed based on the theory of the stochastic learning automata to enable the autonomous MEC server selection by the UAVs. The overall framework was evaluated via modeling and simulation, in terms of its efficiency and effectiveness, by studying multiple operation approaches and scenarios.

Part of our current and future work contains the extension of this model under the principles of Contract Theory, where each UAV acts as an "employer" and the MEC servers as "employees" offering their data processing capabilities and being rewarded by the UAVs.

References

[1] L.-a. Matti and L. Kari, "Key drivers and research challenges for 6g ubiquitous wireless intelligence," *University of Oulu*, ISBN: 978-952-62-2354-4, 2019.

[2] 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.

[3] 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.

[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] C. Wang, F. R. Yu, C. Liang, Q. Chen, and L. Tang, "Joint computation offloading and interference management in wireless cellular networks with mobile edge computing," *IEEE Trans. on Veh. Tech.*, vol. 66, no. 8, pp. 7432–7445, 2017.

[6] C. Wang, C. Liang, F. R. Yu, Q. Chen, and L. Tang, "Joint computation offloading, resource allocation and content caching in cellular networks with mobile edge computing," in *2017 IEEE Intern. Conference on Communications*. IEEE, 2017, pp. 1–6.

[7] J. Li, H. Gao, T. Lv, and Y. Lu, "Deep reinforcement learning based computation offloading and resource allocation for mec," in *2018 IEEE Wir. Com. and Net. Conf.* IEEE, 2018, pp. 1–6.

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

[9] Y. Liu, S. Wang, J. Huang, and F. Yang, "A computation offloading algorithm based on game theory for vehicular edge networks," in *IEEE Int. Conf. on Com.* IEEE, 2018, pp. 1–6.

[10] M.-A. Messous, H. Sedjelmaci, N. Houari, and S.-M. Senouci, "Computation offloading game for an uav network in mobile edge computing," in *IEEE Int. Conf. on Com.*, 2017, pp. 1–6.

[11] X. Cao, J. Xu, and R. Zhangt, "Mobile edge computing for cellular-connected uav: Computation offloading and trajectory optimization," in *2018 IEEE 19th International Workshop on Signal Proc. Advances in Wireless Com.* IEEE, 2018, pp. 1–5.

[12] 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.

[13] 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.* ACM, 2011, pp. 325–336.

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