

# Data Offloading in UAV-assisted Multi-access Edge Computing Systems under Resource Uncertainty

Pavlos Athanasios Apostolopoulos\*, Georgios Fragkos\*, Eirini Eleni Tsiropoulou\*, and Symeon Papavassiliou†  
{pavlosapost@unm.edu, gfragkos@unm.edu, eirini@unm.edu, papavass@mail.ntua.gr}

\* Dept. of Electrical and Computer Engineering, University of New Mexico, Albuquerque, NM, USA

† School of Electrical and Computer Engineering, National Technical University of Athens, Athens, Greece

**Abstract**—In this paper, a novel data offloading decision-making framework is proposed, where users have the option to partially offload their data to a complex Multi-access Edge Computing (MEC) environment, consisting of both ground and UAV-mounted MEC servers. The problem is treated under the perspective of risk-aware user behavior as captured via prospect-theoretic utility functions, while accounting for the inherent computing environment uncertainties. The UAV-mounted MEC servers act as a common pool of resources with potentially superior but uncertain payoff for the users, while the local computation and ground server alternatives constitute safe and guaranteed options, respectively. The optimal user task offloading to the available computing choices is formulated as a maximization problem of each user's satisfaction, and confronted as a non-cooperative game. The existence and uniqueness of a Pure Nash Equilibrium (PNE) are proven, and convergence to the PNE is shown. Detailed numerical results highlight the convergence of the system to the PNE in few only iterations, while the impact of user behavior heterogeneity is evaluated. The introduced framework's consideration of the user risk-aware characteristics and computing uncertainties, results to a sophisticated exploitation of the system resources, which in turn leads to superior users' experienced performance compared to alternative approaches.

**Index Terms**—Data offloading, Multi-access Edge Computing, Unmanned Aerial Vehicles, risk-aware behavior, computing uncertainty, Prospect Theory, convex optimization

## I. INTRODUCTION

UNMANNED Aerial Vehicles (UAVs) have gained increasing research and commercial popularity due to their salient attributes, such as flexible deployment, mobility, strong line-of-sight connection links, low-cost, adjustable usage, maneuverability, and hovering ability. Their vital features have enabled them to support not only various military applications, but several civilian services as well, including transportation, industrial monitoring, agriculture services, forest fire monitoring, wireless services. For instance, focusing on the latter, the UAVs have been used among others to enhance the coverage and capacity of the wireless cellular networks, act as flying base stations, and support Internet of Things (IoT)

P.A. Apostolopoulos, G. Fragkos, and E.E. Tsiropoulou are with the Department of Electrical and Computer Engineering, University of New Mexico, Albuquerque, NM, USA, 87131. E-mail: {pavlosapost@unm.edu, gfragkos@unm.edu, eirini@unm.edu}. S. Papavassiliou is with the School of Electrical and Computer Engineering, National Technical University of Athens, Athens, Greece, 15780. E-mail: {papavass@mail.ntua.gr}.

Manuscript received April 27, 2020. Revised October 23, 2020. Revised February 16, 2021.

communications in smart cities environments [1]. Also, over the last few years, the UAVs have been considered as means to provide computing support to the end-users by acting as UAV-mounted multi-access edge computing servers [2]. The concept of multi-access edge computing (MEC), formerly known as mobile edge computing, is a network architecture concept, standardized by the European Telecommunications Standards Institute (ETSI), offering cloud-computing capabilities at the edge of the access network [3].

## A. Related Work & Motivation

Very recently (2018), the use of UAV-mounted MEC servers has been proposed in combination with the ground MEC servers to support end-users' applications' offloading in order to perform computationally intensive tasks, thus, collectively creating a fog computing system [4]. In [5], the authors investigate the stability of a cloud-based UAV system consisting of UAV-mounted MEC servers, in relationship with the acquisition rate of sensors' big data. An air-ground integrated MEC architecture is introduced in [6] consisting of both ground and UAV-mounted MEC servers. The authors highlight the benefits of the UAV-assisted network in terms of edge caching and computing. In [7], the authors consider clusters of UAV-mounted MEC servers and the problem of opportunistic computational offloading is studied to determine the tasks that should be offloaded to the neighboring UAV clusters with sufficient computing resources, in order to increase the UAVs' lifetime and decrease the overall computation time.

Focusing on the problem of users' computation tasks offloading to UAV-mounted MEC servers, the authors in [8] propose an UAV-enabled MEC system, where the UAVs act jointly as relay and data processing nodes to facilitate the communication and computing demands of the ground devices. A joint optimization problem is formulated to minimize the service delay of the ground devices and the UAVs by determining the UAVs optimal position, the communication and computing resource allocation, and the devices' task splitting. A centralized task offloading approach to the UAV-mounted and ground MEC servers is introduced in [9], where an intelligent centralized agent makes optimal decisions about the users' task offloading strategies via sensing the communication and computing conditions of the environment towards optimizing the users' Quality of Experience. The authors in [10] introduce an optimization algorithm to jointly optimize

the task offloading, the bit allocation during transmission, and the UAV trajectory. A similar problem is studied in [11], where the amount of offloaded bits to be processed by the UAVs, as well as the number of output bits returned to the users in the downlink, are jointly optimized, while taking into consideration the users' maximum latency constraints. This research work has been extended in [2] to consider orthogonal and non-orthogonal multiple access techniques to offload the users' data to the UAV-mounted MEC servers, as well as to optimize the UAVs' trajectory subject to their energy budget constraints.

In [12], the UAV-mounted MEC servers' energy saving is studied by jointly optimizing the UAVs trajectory and data offloading to ground MEC servers based on a long short term memory prediction algorithm. In [13], the UAVs energy efficient operation is also studied by jointly optimizing the UAVs trajectory, the users transmission power, and the computation load allocation via introducing a heuristic method based on the successive convex approximation technique. Furthermore, the feature of wireless powered communication has been also incorporated in the UAV-mounted MEC servers. For example in [14], [15] the UAVs are assumed to transmit energy to multiple ground users, who in turn exploit the harvested energy for local computing and computation tasks offloading.

All the aforementioned research works have examined the users computation tasks offloading problem to the UAV-mounted MEC servers and have demonstrated accordingly significant benefits and advances, by making some key assumptions, which do not necessarily hold true in real-life networking scenarios. In particular, all the users are assumed to have rational characteristics and aim to maximize some form of their perceived utility, e.g., minimizing their energy consumption, or transmission and processing time overhead of the computation tasks. However, in reality the users demonstrate a risk-aware behavior, which is driven by their personal characteristics, the actions and behavior of the other users, and the conditions in the UAV-assisted network [16], [17]. Furthermore, when users make decisions regarding offloading and particularly in a distributed manner, the uncertainties introduced by the underlying computing resource availability are not properly accounted for.

Therefore, the users tend to exhibit risk-seeking or loss-aversion behavior during their decision-making process under the presence of uncertainty stemming from the computation environment, which is an intrinsic characteristic of the MEC environment. As a result, the users computation tasks offloading problem needs to account for the human awareness and cognition within the MEC environment, capturing several underlying characteristics, e.g., risk-awareness, which are observed in real-life humans' decision-making process. For that reason, in this work we leverage Prospect Theory [18], which has emerged as a realistic model to capture how people make decisions under uncertainty, by considering and modeling many of their standard biases. As it has been argued in [19], Prospect Theory was proposed as an alternative to pure expected utility theory in order to overcome the unrealistic assumption, typically made by the latter, that risk attitudes are constant across all levels of wealth. Prospect Theory asserts

that the shape of the utility function differs in the gain and loss domains. An interesting indicative example, where the users' risk-aware behavior needs to be considered for the successful design of complex systems, is the human-robot interaction design [20], where the robots should predict the humans' behavior based on the risk-aware characteristics of the latter, in order to collaborate with each other smoothly. Another example refers to the successful design of advertisement campaigns and products' promotions based on the risk-aware bias of the customers with respect to products' negative reviews in their purchasing decision-making [21]. Regardless of the particular application domain (e.g., human-robot interaction, advertisement campaigns, etc.), Prospect Theory evangelizes that the users' payoff, resulting from their autonomous decisions under uncertainty, is determined in a probabilistic manner and deviates from the one received considering users' risk-neutral behavior, as the Expected Utility Theory commonly assumes.

As far as the users computation task offloading problem in MEC servers is concerned, very recently [22], [23], Prospect Theory was utilized in a single MEC server environment to formulate the computation task offloading problem for resource-constrained IoT devices. The authors considered the human awareness, inherent cognitive biases and behavioral characteristics into the devices' operation focusing on full [22] or partial task offloading [23] to the single MEC server. Several additional efforts have been reported in the literature where Prospect Theory has been adopted in various environments and application domains. Some of these applications include: dynamic resource management in 5G wireless networks [16], [24], public safety networks [17], anti-jamming communications in cognitive radio networks [25], users' transmission power management and anti-jamming techniques in UAV-assisted networks [26], and Quality of Experience [27] in cyber-physical social systems.

## B. Contributions & Outline

Despite the fact that several approaches, including game theoretic ones, have been proposed in the literature to study the offloading decisions in UAV-enabled MEC environments, our paper aims at exactly filling the aforementioned gap by incorporating users' behavioral factors in the users' computation task offloading problem. In particular, we consider the existence of two different types of MEC servers (i.e., ground MEC servers and UAV-mounted MEC servers, each type with different characteristics and capabilities), while taking into account users' latency and energy requirements. Towards achieving this goal, we exploit the principles of *Prospect Theory* [18], motivated by the fact that the UAV-mounted MEC servers constitute a competitive resource-constrained environment, where the users make decisions under uncertainty of the available resources and they compete among each other for the shared limited resources. On the other hand, it is noted that the ground MEC servers provide a guaranteed slice of computation resources to each user, given their powerful computing capabilities and availability. This comes in contrast to the UAV-mounted MEC servers that offer an unpredictable

slice of computation resources to each user, due to the fact that the UAV-mounted MEC servers confront the challenge of the constrained energy availability.

The main contributions of our work that differentiate it from the rest of the literature, are summarized below:

1. A heterogeneous complex multi-MEC system consisting of both ground and UAV-mounted MEC servers, that jointly considers the risk-aware behavior of the individual users and the risk of failure of the shared computing resources, is introduced. Given this uncertain environment, and in contrast to the majority of the existing literature that primarily addresses the problem of binary offloading where each user may offload its whole application to one MEC server, in this work we investigate the problem of partial offloading. In the considered setting each user can offload arbitrarily parts of its application to ground and/or UAV-mounted MEC servers for remote execution [28].

2. The choice of partially locally processing the tasks or offloading them to the ground MEC servers, where a guaranteed slice of computation resources is offered to each user, are considered as a safe and guaranteed option, respectively, offering predictable performance to the users. The difference between the safe and guaranteed options, mainly stems from the fact that in the local processing option (i.e. safe option) each user's device computation capabilities depend solely on the user's local characteristics and are known before the user's decision-making and available only for its own explicit use. This differs from the ground MEC server option that, though it presents a powerful computation capability by offering a guaranteed slice of computation resource to each user that offloads parts of its data to it, still constitutes a shared resource among the users. The latter, comes in contrast to the UAV-mounted MEC servers that act as Common Pool of Resources (CPR), providing possibly superior but uncertain payoff to the users. The potential for superior expected payoff stems from the improved communication channel gains resulting from the UAVs proximity to the users compared to the ground MEC servers. On the other hand the uncertainty stems from the probability of over-exploitation of their limited computation resources as a natural outcome of the respective energy constraints, which in turn may drive the users to perceive lower satisfaction based on the principles of the *Tragedy of Commons* [29].

3. The users' risk-aware behavior in their decision-making process is captured via properly formulated prospect-theoretic utility functions, considering the users' actual utilities, their latency and energy requirements, their computation task characteristics, and their perception of the gains and losses. The latter consideration comes in contrast to the majority of existing works and fundamentally differentiates our work from current literature body, since the users are not any more treated as blind utility maximizers, but their behavioral modeling is specifically designed to reflect real life human decision-making under uncertainty.

4. Each user's overall perceived satisfaction is formulated by considering and exploiting all the different computation options available (i.e., UAV-mounted servers, ground MEC

servers, local computing). In that respect, each user's perceived satisfaction utility results from the overall expected prospect theoretic utility obtained from the UAV-mounted MEC servers, the corresponding overall time and energy overhead by offloading part of its data to the ground MEC servers, and from executing the remaining amount of data locally. It is noted that the prospect theoretic utility is of probabilistic nature, as it depends on the computing load and congestion at the UAV-mounted MEC servers, with the latter being considered as fragile computing resources that can fail to serve the users' computation demands due to their constrained energy availability.

5. The user's autonomous and optimal computation task allocation to the available computing alternatives (i.e., local, ground/UAV-mounted MEC servers), is formulated as a convex optimization problem of each user's satisfaction utility. Due to users' competition for shared resources, i.e., UAV-mounted and ground MEC servers, it is confronted as a non-cooperative game among them. The existence and uniqueness of a Pure Nash Equilibrium (PNE) is proven, and a low complexity and distributed algorithm that converges to the PNE is proposed.

The remaining of the paper is organized as follows. In Section II, the considered system model is presented, by introducing the communication and computing model, and defining the users' experienced time and energy overheads from transmitting and processing their data to the available offloading options (i.e., ground and UAV-mounted MEC servers). In Section III, the proposed prospect-theoretic utility function formalities are detailed, whereas in Section IV, the optimal user data offloading problem is formulated and analyzed. In Section V, a distributed low-complexity algorithm exploiting the properties of convex optimization to determine the game's PNE is introduced. The performance evaluation of the proposed framework is achieved via modeling and simulation, and numerical results are presented in Section VI. Section VII concludes the paper.

## II. SYSTEM MODEL

A UAV-assisted multi-MEC system is considered consisting of a set of ground MEC servers  $\mathbb{G} = \{1, \dots, g, \dots, G\}$ , a set of UAV-mounted MEC servers  $\mathbb{F} = \{1, \dots, f, \dots, F\}$ , and a set of users  $\mathbb{U} = \{1, \dots, i, \dots, U\}$ . Let us also denote the set of all available MEC servers by  $\mathbb{S} = \{1, \dots, s, \dots, S\}$ , i.e.,  $\mathbb{S} = \mathbb{G} \cup \mathbb{F}$  and  $S = G + F$ . The ground MEC servers are attached to base stations and access points located in different places on the ground. An indicative topology of the considered UAV-assisted multi-MEC system in Fig. 1.

We further denote by  $T_i = (B_i, t_i, e_i)$  the user's  $i$  computation task, which is characterized by: a) the amount of the  $B_i$  [bits] input bits (i.e., data to be processed), b) the required  $\phi * B_i$  CPU-Cycles where  $\phi > 0$  [ $\frac{\text{CPU-Cycles}}{\text{bit}}$ ] describes the level of the user's computation task's intensity (in the following, we consider that the users are requesting computation tasks with similar computation intensity and thus homogeneous  $\phi$  computation intensity factors are considered, in alignment with current literature [30], [31]), and c) the user's computation task's latency and energy requirements, denoted by  $t_i$  [sec] and

$e_i$  [J], respectively. The latency requirement  $t_i$  is related with the user's task and indicates that the latter has to be completed before this time deadline. Moreover, each user's local device is characterized by a limited energy availability (associated with the actual device's battery). For that reason, the user's device's energy requirement  $e_i$  is considered as well, and it constitutes an upper limit value for the user's overall consumed energy to complete the task. Each user can arbitrarily partition its application into distinct parts and offload them to the ground MEC servers and the UAV-mounted MEC servers, which are capable of processing the users' offloaded data in parallel, while the remaining amount of data is processed locally [15], [28]. Accordingly, the energy requirement, as used in this paper, practically reflects a threshold value that the user may set with respect to the use of its own energy resources for the execution of the specific task under consideration. It essentially refers to energy components consumed only at the user device, either for local execution or for transmission to the server.

The users' communication overhead of associating with multiple UAV-mounted and/or ground MEC servers is assumed negligible compared to the corresponding data transmission and processing overhead. Nevertheless, it is noted that it can be easily incorporated in our model and framework, by considering an additional constant factor - which would typically be of smaller magnitude compared to the rest of the involved overhead factors - in the formulation of the corresponding communication overhead, each time that a user is associated with a sever.

We denote by  $\mathbf{b}_i = (b_{i,1}, \dots, b_{i,s}, \dots, b_{i,S})$  the user's  $i$  offloading vector, where  $b_{i,s}$  [bits] is the amount of data that user  $i$  offloads to the MEC server  $s$  (either ground or UAV-mounted MEC server). Accordingly, the total amount of data that user  $i$  offloads to the MEC servers equals to  $\sum_{s \in \mathbb{S}} b_{i,s} \leq B_i, \forall i \in \mathbb{U}$ , while the rest  $L_i = B_i - \sum_{s \in \mathbb{S}} b_{i,s}$  amount of data is processed locally at the user's device. The data offloading strategy of all the users is  $\mathbf{b} = (\mathbf{b}_1, \dots, \mathbf{b}_U)$ . For practical purposes, and assuming single-communication interface at each user, we consider that each user transmits sequentially its data  $b_{i,s}, \forall s \in \mathbb{S}$ , and each MEC server has sufficient memory to store the received data. Each UAV-mounted MEC server's  $s, s \in \mathbb{F}$ , energy availability is denoted as  $E_s$  [J], a part of which is used for the UAV's operation (e.g., accurately maintaining its position above the ground) and the rest  $E_s^p$  is consumed for the users' offloaded data processing.

### A. Communication & Computing Model

A multi-channel interference limited wireless communication environment is considered, where the system's bandwidth is divided in wireless communication channels, i.e., frequency bands. Each MEC server (ground or UAV-mounted) is assigned and occupies one such wireless communication channel and receives the users' offloaded data through it [32]–[34]. Thus, the users communicating with the same MEC server share the same channel and accordingly they experience intra-channel interference, while avoiding the inter-channel interference stemming from users offloading their data to other MEC servers. Thus, the user's  $i$  uplink data rate to the MEC server

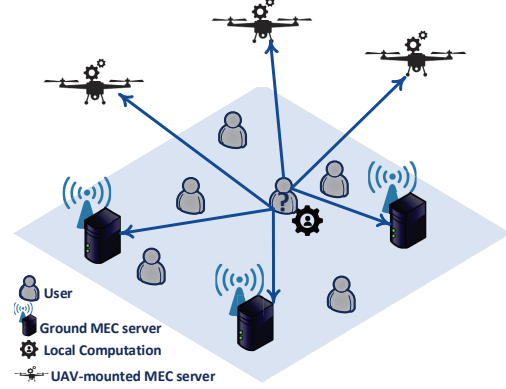


Fig. 1: UAV-assisted multi-MEC System

$s$  is  $R_{i,s} = W_s * \log(1 + \frac{p_{i,s} * g_{i,s}}{\sigma_0^2 + \sum_{j \in U_s, j \neq i} p_{j,s} * g_{j,s}})$ , where  $W_s$  is the MEC server's  $s$  channel bandwidth,  $p_{i,s}$  is the user's  $i$  transmission power to offload its amount of data to the MEC server  $s$ ,  $g_{i,s}$  is the channel gain between the user  $i$  and the MEC server  $s$ ,  $\sigma_0^2$  is the variance of the Additive White Gaussian Noise, and  $U_s = \{i \in \mathbb{U} : b_{i,s} \neq 0\}$  is the set of users that offload a non-zero amount of data to the MEC server  $s$ .

It should be noted that in practice some users may complete their data transmission earlier than others, which means that they may no longer contribute to the interference term (i.e.,  $\sum_{j \in U_s, j \neq i} p_{j,s} * g_{j,s}$ ) of the rest users, i.e.,  $U_s$ , who still transmit their data to the MEC server  $s$ . To fully characterize each user's perceived transmission rate when the user actually performs the data offloading to the MEC server  $s$ , would imply that during the decision-making process, the user  $i$  is aware of specific individual information about the rest of the users, both in terms of individual user offloading strategies as well as communication information (i.e., transmission power, channel gain), such that the actual interference term is evaluated. Such an approach, though would be more accurate by fully exploiting the time dimension as well, it would be rather complex and impractical, or even infeasible in most cases. Moreover, the way that the user's  $i$  uplink data rate  $R_{i,s}$  to the MEC server  $s$  is defined above constitutes a lower bound, i.e., the worst-case, transmission rate that a user perceives by offloading  $b_{i,s}$  data to the MEC server  $s$ . This worst-case formulation of the transmission rate for the purposes of computation offloading, is well aligned with commonly assumed research efforts in the literature [34], [35].

The user  $i$  by offloading  $b_{i,s}$  data to the MEC server  $s, s \in \mathbb{S}$  experiences a transmission time overhead  $O_{i,s}^t|_{tr} = \frac{b_{i,s}}{R_{i,s}}$ , and a transmission energy overhead  $O_{i,s}^e|_{tr} = \frac{b_{i,s}}{R_{i,s}} * p_{i,s}$ . We denote as  $l_i^c$  [ $\frac{\text{CPU-Cycles}}{\text{sec}}$ ] and  $l_i^e$  [ $\frac{\text{Joules}}{\text{CPU-Cycle}}$ ] the user's  $i$  device's local computation capability and energy consumption, respectively. Thus, the user's perceived local time overhead is  $O_i^t|_l = \frac{L_i * \phi}{l_i^c}$  [sec] and its local energy consumption is  $O_i^e|_l = L_i * \phi * l_i^e$  [J]. It should be noted here that in our setting, without loss of generality, we assume that both energy and time overheads are of equal and high importance. Accordingly, taking into account normalization aspects to guarantee the same order of magnitude of the jointly considered time and energy overhead

[34], [35], the user's  $i$  overall local overhead is formulated as follows.

$$O_{i|l} = \frac{O_{i|l}^t}{t_i} + \frac{O_{i|l}^e}{e_i} \quad (1)$$

In Sections II-B and II-C, the ground and UAV-mounted MEC server's computing models are introduced. It should be clarified that in this research work, we assume that a MEC server is capable of parallel processing the users' offloaded data. The latter is commonly considered in the literature [34]–[37], where a MEC server is able of computing the users' offloaded tasks independently through virtualization techniques.

### B. Ground MEC Servers and Actual Overhead

Each ground MEC server  $s$ ,  $s \in \mathbb{G}$  has a powerful computation capability (e.g., high speed CPU). We consider that each ground MEC server offers a guaranteed slice of computation resources  $f_s^G$  [ $\frac{\text{CPU-Cycles}}{\text{sec}}$ ] to each user that offloads part of its data to the specific ground MEC server  $s$ . Thus, the ground MEC server acts as a guaranteed option for the user to process its data. Considering the user's transmission time  $O_{i,s|tr}^t$  and energy  $O_{i,s|tr}^e$  overhead, as well as the processing time for the  $b_{i,s}$  data at the ground MEC server, i.e.,  $\frac{b_{i,s} \cdot \phi}{f_s^G}$ , the user's  $i$  actual overhead for offloading  $b_{i,s}$  data to the ground MEC server  $s$  is given as:

$$O_{i,s|gr} = \frac{O_{i,s|tr}^t + \frac{b_{i,s} \cdot \phi}{f_s^G}}{t_i} + \frac{O_{i,s|tr}^e}{e_i} \quad (2)$$

Thus, the user's  $i$  overall actual overhead by the data offloading and processing to the ground MEC servers is given as:

$$O_{i|gr} = \sum_{s \in \mathbb{G}} O_{i,s|gr} = \sum_{s \in \mathbb{G}} b_{i,s} \left( \frac{1}{R_{i,s} t_i} + \frac{\phi}{f_s^G t_i} + \frac{p_{i,s}}{R_{i,s} e_i} \right) \quad (3)$$

### C. UAV-mounted MEC servers and Expected Overhead

The UAV-mounted MEC servers offer an attractive choice to the users by possibly providing superior (compared to ground MEC servers) payoff to them, due to the potential establishment of better communication channel gains as an outcome of their closer proximity to the users. In this research work, we consider that the UAVs trajectory is a priori known and the UAVs have the ability to hover closer to the users, in comparison to the users' corresponding distance from the ground MEC servers. However, each UAV-mounted MEC server  $s$ ,  $s \in \mathbb{F}$  has limited energy availability  $E_s^p$  to be used for the processing of the users' offloaded data. Furthermore, each UAV-mounted MEC server is considered as a Common Pool of Resources (CPR) and its computation capability, which is shared among the users, is a decreasing function of the overall amount of received data, as the more data are offloaded to the UAV-mounted MEC server, the less computation capability is assigned to each user.

By denoting as  $e_s$  [ $\frac{\text{Joules}}{\text{CPU-Cycle}}$ ] each UAV-mounted MEC server's  $s$  energy consumption, then based on the users' level of computation task intensity  $\phi$ , the threshold data value that each UAV-mounted MEC server can receive for remote processing is  $\bar{B}_s = \frac{E_s^p}{\phi}$ . Let  $\bar{b}_s = \sum_{i \in \mathbb{U}_s} b_{i,s}$  denote the

TABLE I: Summary of Key Notations

Notation	Description [Units]
$\mathbb{S}$	Set of MEC servers
$\mathbb{G}$	Set of ground MEC servers
$\mathbb{F}$	Set of UAV-mounted MEC servers
$\mathbb{U}$	Set of users
$T_i$	User's $i$ computation task
$B_i$	Total input bits of user $i$ [bits]
$\phi$	Computation task's level of intensity of users [CPU-Cycles/bit]
$t_i, e_i$	User's $i$ latency [sec] and energy [J] requirements
$b_{i,s}$	Offloaded data of user $i$ to MEC server $s$ [bits]
$E_s$	UAV-mounted MEC server's energy availability [J]
$R_{i,s}$	Uplink data rate of user $i$ to MEC server $s$
$W_s$	MEC server's channel bandwidth [Hz]
$p_{i,s}$	Transmission power of user $i$ to MEC server $s$
$g_{i,s}$	Channel gain between user $i$ and MEC server $s$
$\sigma_0^2$	Variance of the Additive White Gaussian Noise
$\mathbf{b}_i$	User's $i$ data offloading vector
$L_i$	User's $i$ amount of locally processed data [bits]
$\mathbf{b}$	Data offloading strategy of all users
$\mathbb{U}_s$	Set of users offloading data to MEC server $s$
$O_{i,s tr}^t$	User's $i$ transmission time overhead to offload data to MEC server $s$ [sec]
$O_{i,s tr}^e$	User's $i$ transmission energy overhead to offload data to MEC server $s$ [J]
$l_i^c$	User's $i$ local computation capability [CPU-Cycles/sec]
$l_i^e$	User's $i$ local computation energy consumption [J/CPU-Cycles]
$O_{i l}^t$	User's $i$ local time overhead [sec]
$O_{i l}^e$	User's $i$ local energy consumption [J]
$O_{i l}$	User's $i$ overall local overhead
$F_s^U$	UAV-mounted MEC server's computation capability [CPU-Cycles/sec]
$f_s^G$	Guaranteed computation resources slice assigned to a user by the ground MEC server $s$ [CPU-Cycles/sec]
$f_s^U$	UAV-mounted MEC server's $s$ computation resources slice assigned to a user [CPU-Cycles/sec]
$O_{i,s gr}$	User's $i$ overall overhead by a ground MEC server $s$
$O_{i gr}$	User's $i$ overall overhead by the ground MEC servers
$e_s$	UAV-mounted MEC server's $s$ energy consumption [J/CPU-Cycles]
$E_s^p$	UAV-mounted MEC server's data processing energy availability [J]
$\bar{b}_s$	Overall data received by a UAV-mounted MEC server $s$ [bits]
$\bar{B}_s$	Threshold data value of a UAV-mounted MEC server $s$ [bits]
$p_s(\bar{b}_s)$	Probability of failure of UAV-mounted MEC server $s$
$\alpha_i, \gamma_i$	Sensitivity to the gains and losses of user $i$ , respectively
$k_i$	Loss aversion parameter of user $i$
$u_{i,s}$	User's $i$ prospect-theoretic utility
$O_i$	User's $i$ total overhead
$O_{i,s fl}$	User's $i$ overall overhead by a UAV-mounted MEC server
$O_{i fl}$	User's overall overhead by the UAV-mounted MEC servers
$q_{i,r}$	User's reference point
$s_i(\mathbf{b}_1, \mathbf{b}_{-i})$	User's satisfaction utility
$\Gamma_i$	User's $i$ strategy space
$\mathbf{b}_i^*$	User's $i$ optimal data offloading vector
$\mathbf{b}^*$	Pure Nash Equilibrium point

UAV-mounted MEC server's total received amount of data. Each UAV-mounted MEC server's  $s$  computation resources slice, denoted by  $f_s^U$ , that is allocated to each user is a portion of the server's overall computation capability  $F_s^U$  [CPU-Cycles/sec] to be shared among all users, and is formulated as follows.

$$f_s^U = (1 - \frac{\bar{b}_s}{\bar{B}_s}) F_s^U \quad (4)$$

Each UAV-mounted MEC server  $s$ ,  $s \in \mathbb{F}$  constitutes a rivalrous and subtractable resource, since all the users can arbitrarily offload part of their data for remote execution. This means that its utilization by one user reduces the degree that is exploited and utilized by another user. Thus, it is observed in Eq. 4 that each user computation resources slice  $f_s^U$  decreases

as the overall data  $\bar{b}_s$  received by a UAV-mounted MEC server  $s$  increases due to the fact that the server becomes more congested, especially given the UAV-mounted MEC server's limited energy availability. Also, based on Eq. 4, it is evident that if  $\bar{b}_s \geq \bar{B}_s$ , then the UAV-mounted MEC server is unable to process the received amount of data due to its limited energy availability. It is worth mentioning that even in the case of  $\bar{b}_s \geq \bar{B}_s$ , there may still exist users' offloaded data that could be processed by the UAV-mounted MEC server  $s$  with an appropriate scheduling. However, this is not deterministically known by the users, when the latter ones are making their offloading decisions (Section V). For that reason, in this research work, considering the importance of each user's  $i$  latency ( $t_i$ ) and energy ( $e_i$ ) requirements' fulfilment, we adopt a worst-case scenario approach, where each user considers that with probability  $p_s(\bar{b}_s)$  its offloaded data are unable to be processed by the UAV-mounted MEC server  $s$ . This phenomenon is well known in the literature as the *Tragedy of the Commons* [29]. In the case of the UAV-mounted MEC server's failure, it is more beneficial for the user to offload its data to another MEC server (ground or UAV-mounted) or to process them locally on its device. Moreover, each UAV's overall energy availability  $E_s$  decreases over time, as part of it is consumed for the UAV's operation, thus  $E_s^p, \bar{B}_s$  decrease over time as well. The latter constitutes a computing uncertainty for the users decision-making offloading, as the UAV-mounted MEC server's capability to process the offloaded data by the users is not known in prior. As a result, the uncertainty of each UAV-mounted MEC server's failure is captured by its probability of failure, thus, the users exhibit a risk-aware offloading behavior.

*Assumption 1:* Each UAV-mounted MEC server's  $s, s \in \mathbb{F}$  probability of failure  $p_s(\bar{b}_s)$  is strictly increasing, convex and twice differentiable with respect to  $\bar{b}_s \in [0, \bar{B}_s)$ , with  $p_s(\bar{b}_s) = 1, \forall \bar{b}_s \geq \bar{B}_s$ .

In this paper, we consider a linear probability of failure function, thus  $p_s(\bar{b}_s) = \bar{b}_s/\bar{B}_s, \forall \bar{b}_s < \bar{B}_s$ , while  $p_s(\bar{b}_s) = 1, \forall \bar{b}_s \geq \bar{B}_s$ . The physical meaning of this model is that the UAV-mounted MEC server will deterministically fail to serve the users' computation demands, if their total amount of offloaded data exceeds the server's computation capacity, i.e.,  $p_s(\bar{b}_s) = 1, \forall \bar{b}_s \geq \bar{B}_s$ . In the case however, where the users' total amount of offloaded data does not exceed the server's computation capacity, i.e.,  $\forall \bar{b}_s < \bar{B}_s$ , then, the UAV-mounted MEC server's probability of failure is not zero, but probabilistically depends on the amount of offloaded data that it needs to process, i.e.,  $p_s(\bar{b}_s) = \bar{b}_s/\bar{B}_s$ . This holds true since each UAV-mounted MEC server's actual threshold data value  $\bar{B}_s$  decreases over time, thus is not deterministically known by the users, when they make their data offloading decisions.

It is noted that the rest of the paper's analysis still holds true for any other probability of failure function that follows the Assumption 1 and the selection of a linear probability of failure function is made for presentation purposes. Studying the behavior of additional probability of failure functions, such as the one resulting from a Poisson process regarding the arrival data from all users, is also of high research interest and part of our future work. The probability for the UAV-mounted MEC server to survive and process the users' offloaded data

is  $(1 - p_s(\bar{b}_s))$ . Thus, the user's expected perceived overhead by offloading  $b_{i,s}$  to the UAV-mounted MEC server  $s$  is:

$$\mathbb{E}(O_{i,s}|_{fl}) = (1 - p_s(\bar{b}_s))O_{i,s}|_{fl} + p_s(\bar{b}_s)(O_{i,s}|_l + \frac{O_{i,s}|_{tr}}{t_i} + \frac{O_{i,s}|_{tr}}{e_i}) \quad (5)$$

where

$$O_{i,s}|_{fl} = \frac{O_{i,s}|_{tr} + \frac{b_{i,s} \cdot \phi}{f_s}}{t_i} + \frac{O_{i,s}|_{tr}}{e_i} \quad (6)$$

is the actual overall overhead that user  $i$  experiences by offloading part of its data to a UAV-mounted MEC server  $s$ , where  $O_{i,s}|_{tr} = \frac{b_{i,s}}{R_{i,s}}$  and  $O_{i,s}|_{tr} = \frac{b_{i,s}}{R_{i,s}} \cdot p_{i,s}, \forall s, s \in \mathbb{F}$ . The last two terms in Eq. 5 indicate the user's additional time and energy overhead (accounting for the need to transmit the data before the UAV-mounted MEC server's failure is finally observed). As a result, the user's  $i$  overall expected overhead by the UAV-mounted MEC servers is  $\mathbb{E}(O_i|_{fl}) = \sum_{s \in \mathbb{F}} \mathbb{E}(O_{i,s}|_{fl})$ , and its overall overhead based on its offloading strategy  $\mathbf{b}_i$  is formulated as follows.

$$\mathbb{E}(O_i) = \mathbb{E}(O_i|_{fl}) + O_i|_{gr} + O_i|_l \quad (7)$$

### III. THE PROSPECT OF DATA OFFLOADING

To address the users' subjectivity in the data offloading decision-making under the uncertainty of each UAV-mounted MEC server failure, and considering that in real life users are not risk-neutral, we adopt the principles of *Prospect Theory*. Prospect Theory was introduced by Kahneman and Tversky [38], and it is a behavioral model where the users make decisions under risk and uncertainty of the associated payoff of their choices, which is estimated in a probabilistic manner. Prospect Theory captures users' behavioral patterns, where a user perceives greater dissatisfaction from a potential loss compared to its satisfaction from gains of the same magnitude (*loss aversion property*). The user's losses and gains are evaluated with respect to a reference point, which implies a safe outcome that the user can perceive (*reference dependence property*). Moreover, the users' associated utility function is concave for gains (i.e., users are risk averse in gains) and convex for losses (i.e., users are risk seeking in losses), i.e., *diminishing sensitivity property*.

Some research works have focused on examining users' behavior under the cases of observing only gains or losses in the examined system, i.e., concave and convex part of user's utility function, respectively [22], [39]. However, in this research work, we examine the users risk-aware behavior (i.e., with respect to both gains and losses) under the principles of Prospect Theory, jointly with the risk of failure of the shared UAV-mounted MEC servers' computing resources, as reflected by the theory of the Tragedy of the Commons. Following the prospect-theoretic behavioral model, each user's perceived actual overhead (Eq. 6) by offloading  $b_{i,s}$  data to the UAV-mounted MEC server is evaluated with respect to a reference point  $q_{i,r}$ . In our work, the reference point expresses the corresponding overhead that the user would have obtained if processed locally the  $b_{i,s}$  data, i.e.,  $q_{i,r} = O_i|_l(b_{i,s})$  (Eq. 1). Moreover, following the diminishing sensitivity property, the user's prospect-theoretic utility function is concave with respect to the user's actual overhead (Eq. 6) above the reference

point  $q_{i,r}$ , i.e., gains curve, while it is convex bellow it, i.e., losses curve. Also, the prospect-theoretic utility function has a greater slope in the losses compared to the gains, as the user weighs more the losses (i.e., experiencing a higher actual overhead  $O_{i,s}|_{fl}$  compared to its reference point) compared to the gains (*loss aversion property*).

Based on the above analysis, we combine the properties of reference dependence, diminishing sensitivity, and loss aversion, and we define each user's  $i$  prospect-theoretic utility function, following the general form of the prospect-theoretic utility function [18], as follows.

$$u_{i,s}(q_{i,s}) = \begin{cases} (q_{i,r} - q_{i,s})^{\alpha_i} & , \text{ if } q_{i,s} \leq q_{i,r} \\ -k_i \cdot (q_{i,s} - q_{i,r})^{\gamma_i} & , \text{ if } q_{i,s} > q_{i,r} \end{cases} \quad (8)$$

where  $q_{i,s} = O_{i,s}|_{fl}$  if the UAV-mounted MEC server survives, otherwise  $q_{i,s} = O_i|_l + \frac{O_{i,s}|_{tr}}{t_i} + \frac{O_{e,s}|_{tr}}{e_i}$ , as the  $b_{i,s}$  data are executed locally, while an additional communication overhead is generated by their transmission to the UAV-mounted MEC server (despite its eventual failure). Each user aims to maximize its prospect-theoretic utility (Eq. 8). If the UAV-mounted MEC server survives, the user targets at its gains' maximization (first branch of Eq. 8), i.e., its actual overhead minimization, while in the opposite case, the maximization of the user's prospect-theoretic utility indicates the user's losses' minimization (second branch of Eq. 8).

The user's risk seeking behavior in losses and risk averse behavior in gains are reflected by small values of the parameter  $\alpha_i \in [0, 1]$ . Also, small values of the parameter  $\gamma_i \in [0, 1]$  reflect a higher decrease in the user's prospect-theoretic utility, when its actual overhead is close to the reference point. Without loss of generality, we consider that the users follow similar behavior both in losses and gains, i.e.,  $\alpha_i = \gamma_i, \forall i \in \mathbb{U}$ . Moreover, the parameter  $k_i$  captures the users' loss aversion behavior. Specifically, a user weighs the losses more than ( $k_i > 1$ ) or equal to ( $k_i = 1$ ) the gains, while the opposite holds if  $k_i < 1$ .

Considering the case that  $\bar{b}_s \leq \bar{B}_s = \frac{E_s^p/e_s}{\phi}$ , then the UAV-mounted MEC server's limited energy  $E_s^p$  is expected to be sufficient to process the users' offloaded data  $\bar{b}_s$ . To this end, we assume that the user's perceived actual overhead  $q_{i,s}$  is lower than the reference point ( $q_{i,s} \leq q_{i,r}$ ), given that a UAV-mounted MEC server is considered to have significantly higher computation capability compared to the corresponding one of the users' devices themselves [2], [11] (indicative realistic values are provided in Section VI). Based on Eq. 6 and the first branch of Eq. 8, the user's prospect-theoretic utility is  $u_{i,s} = [b_{i,s}(\frac{\phi}{t_i \cdot l_i^c} + \frac{l_i^e \cdot \phi}{e_i} - \frac{1}{t_i \cdot R_{i,s}} - \frac{\phi}{t_i \cdot f_s^U} - \frac{p_{i,s}}{e_i \cdot R_{i,s}})]^{\alpha_i}$ . In the case of the UAV-mounted MEC server's failure (i.e.,  $\bar{b}_s > \bar{B}_s$ ), the user's actual overhead  $q_{i,s}$  is greater than the reference point  $q_{i,r}$ , as  $q_{i,s} = q_{i,r} + \frac{O_{i,s}|_{tr}}{t_i} + \frac{O_{e,s}|_{tr}}{e_i}$ , so following the second branch of Eq. 8, the user's prospect-theoretic utility is  $u_{i,s} = -k_i \cdot [b_{i,s}(\frac{1}{R_{i,s} \cdot t_i} + \frac{p_{i,s}}{R_{i,s} \cdot e_i})]^{\alpha_i}$ . For notational convenience, we set  $\epsilon_i = (\frac{1}{R_{i,s} \cdot t_i} + \frac{p_{i,s}}{R_{i,s} \cdot e_i})^{\alpha_i}$  and  $g_{i,s} = (\frac{\phi}{t_i \cdot l_i^c} + \frac{l_i^e \cdot \phi}{e_i} - \frac{1}{t_i \cdot R_{i,s}} - \frac{\phi}{t_i \cdot f_s^U} - \frac{p_{i,s}}{e_i \cdot R_{i,s}})^{\alpha_i}$ . Thus, the user's prospect-theoretic utility can be re-written as follows.

$$u_{i,s} = \begin{cases} b_{i,s}^{\alpha_i} \cdot g_{i,s}(\bar{b}_s) & , \text{ with prob. } (1 - p_s(\bar{b}_s)) \\ -k_i \cdot \epsilon_i \cdot b_{i,s}^{\alpha_i} & , \text{ with prob. } p_s(\bar{b}_s) \end{cases} \quad (9)$$

Therefore, each user's expected prospect-theoretic utility by offloading  $b_{i,s}$  data to a UAV-mounted MEC server is formulated as follows.

$$\mathbb{E}(u_{i,s}) = b_{i,s}^{\alpha_i} \cdot h_{i,s}(\bar{b}_s) \quad (10)$$

where  $h_{i,s}(\bar{b}_s) = g_{i,s}(1 - p_s(\bar{b}_s)) - k_i \epsilon_i p_s(\bar{b}_s)$ .

#### IV. OPTIMIZING USERS' SATISFACTION: A GAME THEORETIC APPROACH

##### A. Problem Formulation

The goal of each user is to maximize its overall expected prospect-theoretic utility  $\sum_{s \in \mathbb{F}} \mathbb{E}(u_{i,s})$  that obtains from the UAV-mounted MEC servers, while at the same time to minimize its overall local overhead  $O_i|_l$  and its overall actual overhead  $O_i|_{gr}$  by offloading part of its data to the ground MEC servers. Thus, we introduce each user's satisfaction utility, which is formulated as:

$$s_i(\mathbf{b}_i, \mathbf{b}_{-i}) = \sum_{s \in \mathbb{F}} \mathbb{E}(u_{i,s}) - O_i|_l - O_i|_{gr} \quad (11)$$

where  $\mathbf{b}_{-i} = [\mathbf{b}_1, \dots, \mathbf{b}_{i-1}, \mathbf{b}_{i+1}, \dots, \mathbf{b}_U]$  is the users' offloading strategy vector except of user  $i$ . The physical meaning of the user's satisfaction utility is the user's overall perceived satisfaction by processing its data in the UAV-assisted MEC system by jointly considering the local computation, as well as the computation at the ground MEC servers and the UAV-mounted MEC servers. Based on Eq. 1, 3, 10, the user's satisfaction utility is written as follows.

$$s_i(\mathbf{b}_i, \mathbf{b}_{-i}) = \sum_{s \in \mathbb{F}} b_{i,s}^{\alpha_i} \cdot h_{i,s}(\bar{b}_s) - L_i \phi \left( \frac{1}{t_i \cdot l_i^c} + \frac{l_i^e}{e_i} \right) - \sum_{s \in \mathbb{G}} b_{i,s} \left( \frac{1}{R_{i,s} \cdot t_i} + \frac{\phi}{f_s^G \cdot t_i} + \frac{p_{i,s}}{R_{i,s} \cdot e_i} \right) \quad (12)$$

where, as mentioned earlier,  $L_i = B_i - \sum_{s \in \mathbb{S}} b_{i,s}$  are the user's  $i$  data that remain to be processed locally.

Each user aims to autonomously determine its optimal data offloading  $\mathbf{b}_i^*$  by maximizing its satisfaction utility  $s_i$ , while at the same time it perceives a non-negative expected prospect-theoretic utility  $\mathbb{E}(u_{i,s})$  by each UAV-mounted MEC server, since a negative value of the latter implies a high probability of failure for the UAV-mounted MEC server. Furthermore, each user's optimal offloading strategy  $\mathbf{b}_i^*$  should satisfy its latency and energy requirements, i.e.,  $\mathbb{E}(O_i)|_t \leq t_i$ ,  $\mathbb{E}(O_i)|_e \leq e_i$ , where  $\mathbb{E}(O_i)|_t$  and  $\mathbb{E}(O_i)|_e$  are the expected overall time and energy overheads, as formulated in Eq. 14, 15, respectively. It is noted that the user's overall time overhead (Eq. 14) considers its aggregated transmission time that is required to sequentially transmit its offloaded data to the MEC servers. If we had considered that each user's device supports a multi-communication interface, i.e., transmission to more than one MEC server at the same time through multiple channels, instead of the single-communication interface assumed here, then the user's overall corresponding transmission time would be replaced by the maximum required transmission time. However, even in this case the provided mathematical analysis would follow the same line of thread.



Thus, each user's satisfaction utility maximization problem can be formulated as follows.

$$\begin{aligned} & \underset{\mathbf{b}_i \in \Gamma_i}{\text{maximize}} && s_i(\mathbf{b}_i, \mathbf{b}_{-i}) \\ & \text{subject to} && \left. \begin{aligned} & \sum_{s \in \mathbb{S}} b_{i,s} \leq B_i, \\ & \mathbb{E}(u_{i,s}) \geq 0, \forall s \in \mathbb{F}, \\ & \mathbb{E}(O_i)|_t \leq t_i, \\ & \mathbb{E}(O_i)|_e \leq e_i \end{aligned} \right\} (C_i) \end{aligned} \quad (13)$$

where  $\Gamma_i = \overbrace{[0, B_i] \times \dots \times [0, B_i]}^{S \text{ - times}}$ , and  $(C_i)$  are the constraints that each user's optimal offloading strategy  $\mathbf{b}_i^*$  must satisfy.

The above maximization problem (Eq. 13) can be confronted as a non-cooperative game among the users who aim to determine their optimal data offloading strategy in a distributed manner. Let  $G = [\mathbb{U}, \{\Gamma_i\}_{i \in \mathbb{U}}, \{s_i\}_{i \in \mathbb{U}}]$  denote the non-cooperative game, where  $\mathbb{U}$  is the users' set,  $\Gamma_i$  is each user's strategy space, and  $s_i$  is its satisfaction utility. The solution of the above maximization problem is captured by the Pure Nash Equilibrium (PNE), which is the users' offloading vector  $\mathbf{b}^* = [\mathbf{b}_1^*, \dots, \mathbf{b}_i^*, \dots, \mathbf{b}_U^*]$ , where no user has the incentive to change its offloading strategy  $\mathbf{b}_i^*$ , given the strategies of the rest users  $\mathbf{b}_{-i}^* = [\mathbf{b}_1^*, \dots, \mathbf{b}_{i-1}^*, \mathbf{b}_{i+1}^*, \dots, \mathbf{b}_U^*]$ .

**Definition 1:** The vector  $\mathbf{b}^* = [\mathbf{b}_1^*, \dots, \mathbf{b}_i^*, \dots, \mathbf{b}_U^*] \in \Gamma$ ,  $\Gamma = \Gamma_1 \times \dots \times \Gamma_U$ , is a Pure Nash Equilibrium (PNE) of the non-cooperative game  $G$ , if  $\forall i \in \mathbb{U}$  it holds true that  $s_i(\mathbf{b}_i^*, \mathbf{b}_{-i}^*) \geq s_i(\mathbf{b}_i, \mathbf{b}_{-i}^*)$ ,  $\forall \mathbf{b}_i \in \Gamma_i$ .

It is noted that in principle, finding the PNE of a non-cooperative game could be essentially considered as a complex combinatorial problem among the users, whose computation complexity makes it intractable [40]. To treat this issue, in this work we focus on investigating a distributed solution that overcomes the aforementioned limitations and inefficiencies. In particular, the existence and uniqueness of a PNE point of the non-cooperative game  $G$  is proven (Section IV-B). Moreover, capitalizing on the continuous Best Response (BR) dynamics methodology and properties, the convergence of a distributed-based method to the unique PNE is proven [41]. Specifically, following the BR principles, each time a user is selected to determine its optimal offloading data strategy by solving a convex optimization problem (Section V).

### B. Existence, Uniqueness and Convergence of PNE

We denote as  $A_i$ , each user's set of strategies that satisfy the group of constraints  $(C_i)$ , thus  $A_i = \{\mathbf{b}_i \in \Gamma_i : \mathbf{b}_i \text{ satisfies } (C_i)\}$ ,  $A_i \subseteq \Gamma_i$ . Let us introduce the transformed non-cooperative game  $G' = [\mathbb{U}, \{A_i\}_{i \in \mathbb{U}}, \{s_i\}_{i \in \mathbb{U}}]$ .

**Theorem 1:** The non-cooperative game  $G'$  among the users is an  $n$ -person concave game, where  $n = U$ .

In order to prove the above theorem, we first state the following Lemmas 1-4.

**Lemma 1:** For each user  $i$  and each UAV-mounted MEC server  $s, s \in \mathbb{F} = \mathbb{S} - \mathbb{G}$ , there exists a threshold value  $\tilde{b}_{i,s} \geq 0$ , such that  $h_{i,s}(\tilde{b}_{i,s}) = 0$ , and  $\mathbb{E}(u_{i,s}) \geq 0$ ,  $\forall b_{i,s} \leq \tilde{b}_{i,s}$ , while  $\mathbb{E}(u_{i,s}) < 0$ ,  $\forall b_{i,s} > \tilde{b}_{i,s}$ .

*Proof:* See Appendix A.

Consequently, based on Lemma 1 the maximization problem in Eq. 13 can be rewritten as follows:

$$\begin{aligned} & \underset{\mathbf{b}_i \in \Gamma_i}{\text{maximize}} && s_i(\mathbf{b}_i, \mathbf{b}_{-i}) \\ & \text{subject to} && \left. \begin{aligned} & \sum_{s \in \mathbb{S}} b_{i,s} \leq B_i, \\ & 0 \leq b_{i,s} \leq \tilde{b}_{i,s}, \forall s \in \mathbb{F}, \\ & \mathbb{E}(O_i)|_t \leq t_i, \\ & \mathbb{E}(O_i)|_e \leq e_i \end{aligned} \right\} (C_i) \end{aligned} \quad (16)$$

where the second constraint  $(C_i)$  was replaced by the inequality  $0 \leq b_{i,s} \leq \tilde{b}_{i,s}$ .

**Lemma 2:** For each user  $i$  and each UAV-mounted MEC server  $s, s \in \mathbb{F}$ , the expected prospect-theoretic utility  $\mathbb{E}(u_{i,s})$  (Eq. 10) is a strictly concave function  $\forall b_{i,s} \in (0, \tilde{b}_{i,s})$ , where  $\tilde{b}_{i,s}$  is the threshold value that was defined in Lemma 1.

*Proof:* See Appendix B.

**Lemma 3:** Each user's group of constraints  $(C_i)$  is a set of convex functions.

*Proof:* See Appendix C.

Based on Lemma 3, each user's set  $A_i$  is the intersection of the level sets of the convex functions in Eq. C.1, thus  $A_i = (\bigcap_{n_1 \in \{1,4,5\}} \text{Lev}(\mu_i^{(n_1)}, 0)) \cap (\bigcap_{n_2 \in \{2,3\}} \text{Lev}(\mu_{i,s}^{(n_2)}, 0))$ ,  $\forall s \in \mathbb{F}$ , which are necessarily convex sets (see Section 3.1.6 of [42]). Therefore, each user's set of strategies  $A_i$  is a convex set as an intersection of convex sets.

**Lemma 4:** Each user's satisfaction utility  $s_i$  is a concave function over the strategy space  $A_i$ .

*Proof:* See Appendix D.

Based on Lemmas 1 - 4, each user's strategy space  $A_i$  is a convex set, and each user's  $i$  satisfaction utility  $s_i(\mathbf{b}_i, \mathbf{b}_{-i})$  is a concave function over the set  $A_i$ . Therefore, the non-cooperative game  $G'$  is an  $n$ -person concave game, where  $n = U$ , and the proof of Theorem 1 is completed. An  $n$ -person concave game has at least one PNE point [43], thus, the existence of at least one PNE point for the non-cooperative game  $G'$  is guaranteed. Finally, based on Theorem 1, Lemma 4, and [43], the following Theorem proves the convergence of the users' strategies to the PNE.

**Theorem 2:** Considering the user  $i$  and an  $S \times S$  matrix function  $\mathbb{X}_i$ ,  $(\mathbb{X}_i)_{ss'} = \lambda_i \frac{\partial^2 s_i}{\partial b_{i,s} \partial b_{i,s'}}$ ,  $\forall s, s' \in \mathbb{S}$ , and the positive constant choices  $\lambda_i > 0$ , then the PNE of the game  $G'$  is unique if  $\mathbb{X}_i + \mathbb{X}_i^T$  is strictly negative definite. Also, starting from any initial offloading strategy vector  $\mathbf{b} = (\mathbf{b}_1, \dots, \mathbf{b}_U)$ ,  $\mathbf{b} \in A = A_1 \times \dots \times A_i \times \dots \times A_U$ , the continuous Best Response (BR) dynamics converge to the unique PNE [41].

*Proof:* See Appendix E.

It is noted that, given that the user's satisfaction utility  $s_i(\mathbf{b}_i, \mathbf{b}_{-i})$  is a concave function over its convex strategy space  $A_i$  (Lemma 4), it has a global maximum point. In the case that the global maximum point is beyond the user's feasibility region, i.e., strategy space  $A_i$ , then the user converges to its maximum data offloading strategy (see DCP algorithm's line 13 in Section V-B), in order to maximize its satisfaction utility.



$$\mathbb{E}(O_i)|_t = \mathbb{E}(O_i|_{fl})|_t + O_i|_{gr}|_t + O_i|_l|_t \quad \frac{p_s = \bar{b}_s / \bar{B}_s, \bar{b}_s = b_{i,s} + \sum_{i' \in U_s - \{i\}} b_{i',s}}{f_s^U = (1 - \frac{\bar{b}_s}{\bar{B}_s}) F_s^U, \forall s \in \mathbb{F}} \sum_{s \in \mathbb{F}} b_{i,s} \left( \frac{1}{R_{i,s}} + \frac{\phi}{f_s^U} \right) + \sum_{s \in \mathbb{F}} \frac{\phi}{l_i^e} \left( \frac{b_{i,s}^2}{\bar{B}_s} + b_{i,s} \frac{\sum_{i' \in U_s - \{i\}} b_{i',s}}{\bar{B}_s} \right) \quad (14)$$

$$+ \sum_{s \in \mathbb{G}} b_{i,s} \left( \frac{1}{R_{i,s}} + \frac{\phi}{f_s^G} \right) + \frac{\phi}{l_i^e} (B_i - \sum_{s \in \mathbb{S}} b_{i,s})$$

$$\mathbb{E}(O_i)|_e = \mathbb{E}(O_i|_{fl})|_e + O_i|_{gr}|_e + O_i|_l|_e \quad \frac{p_s = \bar{b}_s / \bar{B}_s, \bar{b}_s = b_{i,s} + \sum_{i' \in U_s - \{i\}} b_{i',s}}{\sum_{s \in \mathbb{S}} b_{i,s} \frac{p_{i,s}}{R_{i,s}} + \sum_{s \in \mathbb{F}} b_{i,s} \phi l_i^e \frac{b_{i,s} + \sum_{i' \in U_s - \{i\}} b_{i',s}}{\bar{B}_s}} \quad (15)$$

$$+ \phi l_i^e (B_i - \sum_{s \in \mathbb{S}} b_{i,s})$$

## V. DETERMINING THE EQUILIBRIUM

### A. A Convex Optimization Approach

Each user's best response offloading strategy  $\mathbf{b}_i^*(\mathbf{b}_{-i}) : A_{-i} \Rightarrow A_i$  can be formulated as follows.

$$\mathbf{b}_i^*(\mathbf{b}_{-i}) = \arg \max_{\mathbf{b}_i \in A_i} (s_i(\mathbf{b}_i, \mathbf{b}_{-i})), \mathbf{b}_{-i} \in A_{-i} \quad (17)$$

where  $A_{-i} = A_1 \times \dots \times A_{i-1} \times A_{i+1} \times \dots \times A_U$  and equivalently it can be written as:

$$\mathbf{b}_i^*(\mathbf{b}_{-i}) = \arg \min_{\mathbf{b}_i \in A_i} (s'_i(\mathbf{b}_i, \mathbf{b}_{-i})), \mathbf{b}_{-i} \in A_{-i} \quad (18)$$

Therefore, each user should solve the following optimization problem to determine its optimal data offloading strategy.

$$\begin{aligned} & \underset{\mathbf{b}_i \in A_i}{\text{minimize}} && s'_i(\mathbf{b}_i, \mathbf{b}_{-i}) \\ & \text{subject to} && \left. \begin{aligned} & \sum_{s \in \mathbb{S}} b_{i,s} \leq B_i, \\ & 0 \leq b_{i,s} \leq \bar{b}_{i,s}, \forall s \in \mathbb{F}, \\ & \mathbb{E}(O_i)|_t \leq t_i, \\ & \mathbb{E}(O_i)|_e \leq e_i \end{aligned} \right\} (C_i) \end{aligned} \quad (19)$$

It is clarified that the non-offloading strategy (i.e.,  $\mathbf{b}_i = \mathbf{0}$ ) corresponds to the worst case decision, as in that case the users would execute their tasks locally by using their own devices limited resources, which would conclude to lower perceived satisfaction, compared to the case where part of their tasks are offloaded to the MEC environment. Thus, under the assumption that the non-offloading strategy, i.e.,  $\mathbf{b}_i = \mathbf{0}$ , constitutes a worst feasible solution of the optimization problem in Eq. 19 (that is  $\mathbf{b}_i = \mathbf{0} \in A_i$ ), the proposed distributed algorithm (Section V-B) will examine and eventually converge to any alternative offloading strategy (if it exists) that satisfies the constraints in Eq. 19 and leads to a higher perceived satisfaction utility. As a result, the optimization problem in Eq. 19 is a non-linear feasible convex optimization problem, thus  $A_i \neq \emptyset$ .

### B. Algorithm & Complexity Analysis

In this section, the Distributed algorithm for Convergence to the PNE (DCP Algorithm) of the non-cooperative game  $G'$  is presented. Firstly, each UAV-mounted MEC server evaluates its threshold data value  $\bar{B}_s$ , and the latter is shared with the users via a broadcasted signal, at the beginning of the users' offloading decision-making process. As discussed in Section II-C, each UAV-mounted MEC server's threshold data value decreases over time, thus in practice it may deviate from the

### Algorithm 1 DCP Algorithm

---

```

1: Input/Initialization:  $\mathbb{F}, \mathbb{G}, \mathbb{U}, T_i, \mathbf{b}_i \in \Gamma_i, \forall i \in \mathbb{U}, \bar{B}_s, \bar{b}_s,$ 
    $\sum_{j \in U_s, j \neq i} p_{j,s} g_{j,s}, \forall s \in \mathbb{S}, ite = 0$ 
2: Output: PNE strategy  $\mathbf{b}^* = (\mathbf{b}_1^*, \dots, \mathbf{b}_U^*)$ 
3: while Convergence == 0 do
4:    $ite = ite + 1$ 
5:    $flag = 0$ 
6:   for  $i = 1$  to  $U$  do
7:     for  $s = 1$  to  $S$  do
8:       user  $i$  calculates the transmission uplink rate  $R_{i,s}$ 
9:       if ( $s \in \mathbb{F}$ ) then
10:         $r_{i,s} = \text{BinarySearch}([0, \bar{B}_s], \epsilon);$ 
11:       end if
12:       if ( $s \in \mathbb{F}$ ) then
13:         $\bar{b}_{i,s} = \min(r_{i,s}, B_i);$ 
14:       end if
15:     end for
16:      $\mathbf{b}_i^* = fmincon();$ 
17:     if ( $|\bar{b}_{i,s} - b_{i,s}| \leq \epsilon, \forall s \in \mathbb{S}$ ) then
18:        $flag = flag + 1;$ 
19:     end if
20:      $\mathbf{b}_i = \mathbf{b}_i^*$ 
21:     user  $i$  updates  $\bar{b}_s, \sum_{j \in U_s, j \neq i} p_{j,s} g_{j,s}, \forall s \in \mathbb{S}$ 
22:     user  $i$  broadcasts the new values intra-channel
23:   end for
24:   if ( $flag == U$ ) then
25:     Convergence = 1,  $Ite = ite;$ 
26:   end if
27: end while

```

---

received threshold data value  $\bar{B}_s$  by the users at the beginning of their offloading decision-making. The latter uncertainty is captured through the UAV-mounted MEC server's probability of failure function (Section II-C). Following the principles of continues BR dynamics, at each round a user is selected to determine its optimal offloading strategy. Each user receives the  $\bar{b}_s, \forall s \in \mathbb{F}$ , and the factor  $\sum_{j \in U_s, j \neq i} p_{j,s} g_{j,s}, \forall s \in \mathbb{S}$  via intra-channel broadcasted signals [34] from the user that was selected on the previous round to determine its offloading strategy, thus avoiding any need for each user to receiving individual information about the rest of the users, both in terms of individual user offloading strategies as well as communication information (i.e., channel gains). Moreover, based on Lemma 1 the root  $r_{i,s}$  of the equation  $h_{i,s} = 0$  exists, and since the  $h_{i,s}$  is a strictly decreasing function, the root  $r_{i,s}$  is found via Binary Search in  $[0, \bar{B}_s]$ , while  $\bar{b}_{i,s}$  is obtained as:  $\bar{b}_{i,s} = \min(r_{i,s}, B_i)$ . Moreover, in order to solve the non-linear convex optimization problem in Eq. 19, a variety of known methods can be applied [44]. In this paper, the method of the sequential quadratic programming (SQP) [45] is adopted by using the function *fmincon()* in the MATLAB Optimization Toolbox [46]. Finally, after the user  $i$  determines its offloading

decision  $\mathbf{b}_i$ , then it appropriately updates and broadcasts the received  $\bar{b}_s$  and the factor  $\sum_{j \in U, j \neq i} p_{j,s} g_{j,s}, \forall s \in \mathcal{S}$ .

Regarding the DCP algorithm's complexity, each user applies a Binary Search routine in each interval  $[0, \bar{B}_s]$ , so as to determine the  $r_{i,s}$  and  $\tilde{b}_{i,s}, \forall s \in \mathcal{F}$ . Therefore, each user finds the  $\tilde{b}_{i,s}, \forall s \in \mathcal{F}$ , with a complexity  $\mathcal{O}(F \cdot \log_2(\max_{s \in \mathcal{F}}(\bar{B}_s)))$ . By denoting as  $\mathcal{O}(\Delta)$  the complexity of the  $fmincon()$  function, and since the rest operations involve only algebraic calculations, each user's complexity to allocate its best response offloading strategy  $\mathbf{b}_i^*$  at each iteration  $ite$  of the Best Response (BR)-dynamics is  $\mathcal{O}(\Delta + F \cdot \log_2(\max_{s \in \mathcal{F}}(\bar{B}_s)))$ . Considering that the DCP algorithm is executed by  $U$  users, and denoting as  $Ite$  the required iterations for convergence to the PNE, the overall complexity of the DCP algorithm is  $\mathcal{O}(U \cdot Ite \cdot (\Delta + F \cdot \log_2(\max_{s \in \mathcal{F}}(\bar{B}_s))))$ . Finally, since the complexity of the optimization problem  $\mathcal{O}(\Delta)$  can be considered significantly greater than the complexity  $\mathcal{O}(F \cdot \log_2(\max_{s \in \mathcal{F}}(\bar{B}_s)))$ , then, the overall complexity of the DCP algorithm is  $\mathcal{O}(U \cdot Ite \cdot \Delta)$ .

## VI. NUMERICAL RESULTS

In this section, a detailed numerical evaluation is presented to study the performance and the inherent attributes of the proposed framework in the UAV-assisted network. Initially, we assume users exhibiting common risk averse behavior, in order to gain some insight about the process of optimal data offloading in each computing environment, as well as the corresponding utility obtained (Section VI-A), while subsequently, the impact of user heterogeneity on the data offloading process is investigated (Section VI-B). A comparative evaluation of our approach against alternative data offloading strategies is provided in Section VI-C, while in Section VI-D the proposed framework's performance is studied for different topologies with respect to the number of users and their position distribution. Finally, Section VI-E summarizes the main observations derived, by providing meaningful insights about the overall operation and key features of the framework. The proposed framework's evaluation was conducted in a MacBook Pro Laptop, 2.5GHz Intel Core i7, with 16GB LPDDR3 available RAM.

We consider a UAV-assisted network servicing  $U = 200$  users, via a set of  $S = 10$  MEC servers, i.e.,  $G = 7$  ground MEC servers and  $F = 3$  UAV-mounted MEC servers with each UAV having a coverage area of radius  $R_s = 100m$ . Unless otherwise explicitly stated, the users are randomly and uniformly distributed in two-dimension grid  $1000m \times 1000m$ . Each user's channel gain is modeled as  $g_{i,s} = \frac{1}{d_{i,s}^\theta}$ , where  $d_{i,s}$  is the user's  $i$  distance from the MEC server  $s$  and  $\theta = 3$  is the distance loss exponent. In this research work, in line with the corresponding models adopted in the majority of the related literature [47], [48], we consider the free-space path loss model regarding the users' channel gain in communication with the MEC servers (ground or UAV-mounted), as the line-of-sight links are much more dominant than other channel impairments such as shadowing or small-scale fading [49]. However, it is noted that the adopted channel model does not have an impact on the foundations and validity of the proposed

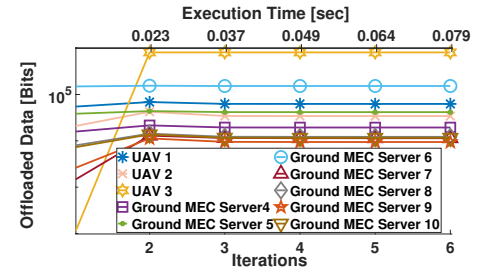


Fig. 2: Convergence of an indicative user's offloaded data to the ground and UAV-mounted MEC servers

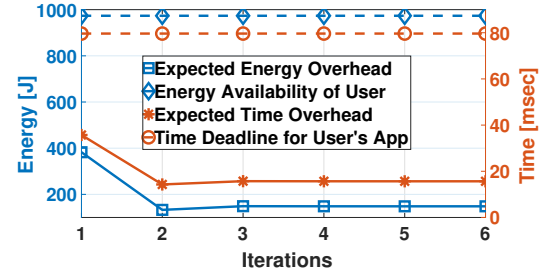


Fig. 3: Convergence of an indicative user's energy and time overhead by offloading data to all the MEC servers

distributed data offloading framework, which can be directly applied by adopting other channel models as well. Each MEC server's channel bandwidth is  $W_s = 5MHz$ , and each user's transmission power to the MEC server  $s$  is  $p_{i,s} = \frac{d_{i,s}^2}{R_s^2}$ , so it is normalized and proportional to its distance from the respective MEC server. Also, we set  $l_i^c \in [0.1, 1] \cdot 10^9 \frac{CPU-Cycles}{sec}$ ,  $l_i^e = 10^{-9} \frac{J}{CPU-Cycle} \forall i \in U$ ,  $F_s^U \in [4, 10] \cdot 10^9 \frac{CPU-Cycles}{sec}$ ,  $E_s \in [100, 200]KJ$ ,  $\tilde{b}_s \in [30, 70]\% \cdot \sum_{i=1}^{200} B_i$ ,  $B_i \in [1000, 5000]KB$  and  $\phi = 10^3 [\frac{CPU-Cycles}{bit}]$  [50], [51]. Unless otherwise explicitly stated, we assume a homogeneous population with common risk preferences, i.e.,  $\alpha_i = 0.2$  and  $k_i = 5$ ,  $\forall i \in U$ .

### A. Pure Operation of the Algorithm

In the following, we present the operational characteristics and performance of the proposed user-centric prospect-theoretic data offloading approach in a UAV-assisted network consisting of 3 UAV-mounted MEC servers and 7 ground MEC servers. Fig. 2 illustrates the evolution of a representative user's data offloading  $b_{i,s}$  at each MEC server (either ground or UAV-mounted), as a function of the DCP algorithm's iterations required for convergence to the PNE. It is clearly shown that the convergence is achieved in a few iterations (i.e., less than 4), starting from any feasible initial random value, while the corresponding average time that the user needs to determine its optimal offloading strategy till convergence is achieved, is relatively low as well, as demonstrated on the upper horizontal axis of Fig. 2 (for practical purposes less than 0.05 sec). Similarly, Fig. 3, presents the corresponding experienced energy and time overhead of a representative user, where we observe that the corresponding values at the PNE satisfy the user's computation task's latency and energy requirements. Fig. 4 presents the average satisfaction utility

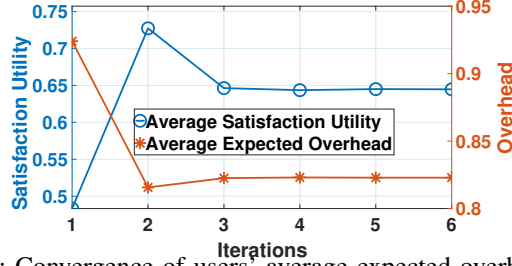


Fig. 4: Convergence of users' average expected overhead and satisfaction utility by offloading data to all the MEC servers

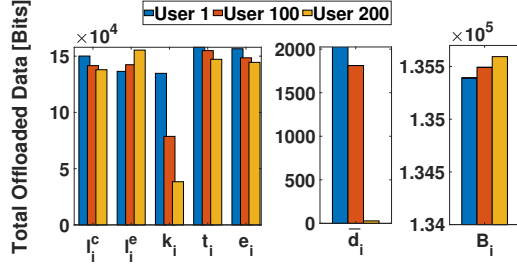


Fig. 5: Three indicative users' total offloaded data as a function of their personal parameters

and corresponding expected overhead of all the users by offloading data to all the MEC servers. The results illustrate that after the convergence to the optimal data offloading point, the users experience high levels of satisfaction and low levels of expected overhead.

In Fig. 5 and Fig. 6, we present the total offloaded bits and the satisfaction utility respectively, of three representative users by examining the effect of seven different personal parameters, i.e., overall average distance from all the MEC servers  $\bar{d}_i$ , local computing capability  $l_i^c$  and energy consumption  $l_i^e$ , total amount of bits  $B_i$ , loss aversion parameter  $k_i$  and the latency  $t_i$  and energy  $e_i$  requirements. It is noted that every parameter's value under examination is assigned in an ascending order to the users with ID 1, 100, and 200, while when we examine the impact of each one of these parameters, all other parameters' values remain the same for all three users. The results reveal that the less distant is the user from the UAVs, the more data will offload to them, as less power is needed for its transmission resulting in lower energy overhead. For this reason, in Fig. 6 it is observed that the user with ID 200, who is the most distant from the MEC servers, experiences the lowest satisfaction utility, as it offloads the smallest amount of data and processes the majority of its data locally. Regarding the impact of the local computing capability user 1, who has the lowest  $l_i^c$ , tends to offload the greatest amount of data compared to the other users, resulting to a greater satisfaction utility. The exact opposite impact is observed for the local energy consumption  $l_i^e$ . With reference to the loss aversion parameter  $k_i$ , the greater its value is, the more loss-averse the users appear, thus user 200, who has the greatest  $k_i$  value, offloads the smallest amount of data and experiences a lower satisfaction utility. In addition, the more data  $B_i$  a user needs to process, the more data it will offload to the MEC servers and process locally, thus, it receives low satisfaction utility. Finally, if the user's latency and energy requirements are relaxed, then the user will prefer to offload less data to the MEC servers,

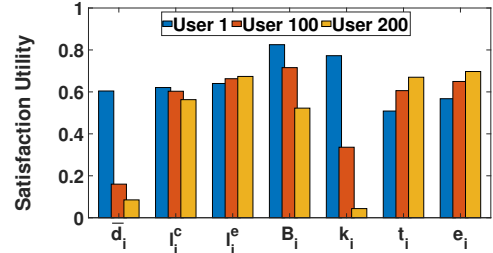


Fig. 6: Three indicative users' satisfaction utility as a function of their personal parameters

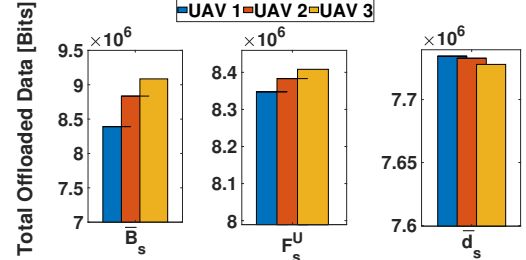


Fig. 7: UAV-mounted MEC servers' total received offloaded data by all the users as a function of the system's parameters

resulting to high levels of satisfaction utility, as the total local overhead is low and satisfies the users.

A study from the system's perspective is also presented in Fig. 7-9 considering the threshold data value  $\bar{B}_s$ , the UAV-mounted MEC server's computation capability  $F_s^U$ , and the average distance  $\bar{d}_s$  of the UAV-mounted MEC server  $s$  from the users. It is noted that every examined parameter's value is assigned in an ascending order to the UAV-mounted MEC servers with ID 1, 2, and 3, while when we examine the impact of each one of these parameters, all other parameters' values remain the same for all three UAV-mounted MEC servers. In particular, it is observed that the greater the UAV-mounted MEC servers' computational capability  $F_s^U$  is, the more data it collects from the users (Fig. 7), as it appears as a more appealing choice, however its probability of failure increases (Fig. 8). Also, the greater is the UAV-mounted MEC server's average distance  $\bar{d}_s$  from the users, the less data it collects, as the users must consume more energy to send their data. Moreover, for larger values of the UAV-mounted MEC server's operational threshold  $\bar{B}_s$ , the UAV appears more robust in terms of the amount of data that it can process, thus, its probability of failure is lower (Fig. 8). Also, as expected the energy that each UAV-mounted MEC server consumes to process the users' offloaded data increases with respect to the total amount of data (Fig. 9).

Finally, Fig. 10 presents the total offloaded bits that each ground MEC server received by studying the impact of its computational capability  $f_s^G$  and its average distance  $\bar{d}_s$  from the users (the values  $f_s^G$  and  $\bar{d}_s$  increase with respect to the ascending ID of the ground MEC server). A similar trend with the UAV-mounted MEC servers is observed, i.e., the greater  $f_s^G$  a ground MEC server has or the less distant is from the users, the more data it receives.

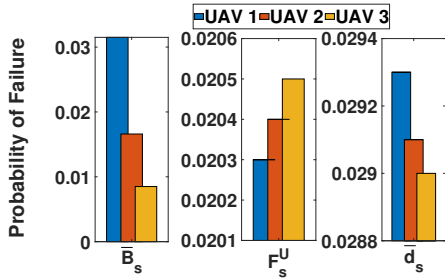


Fig. 8: UAV-mounted MEC servers' probability of failure as a function of the system's parameters

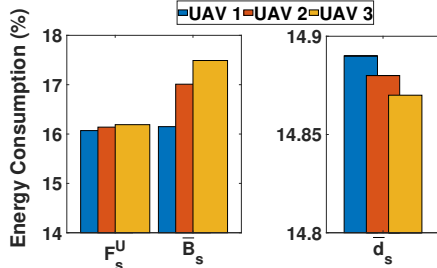


Fig. 9: UAV-mounted MEC servers' energy consumption as a function of the system's parameters

### B. Heterogeneous Users - Loss Aversion

In this section, the impact of the users' heterogeneous loss aversion behavior on their data offloading decisions and achieved satisfaction utility is evaluated. Specifically, a heterogeneous scenario, where the users are associated with different loss aversion parameters  $k_i$ , is compared against a homogeneous scenario, where all the users have the same exactly loss aversion parameter (equal to the average value of the corresponding  $k_i$  parameters in the heterogeneous scenario). It is reminded that the more loss averse is the user's behavior, the greater is the loss aversion parameter  $k_i$ . Thus, those users offload less amount of data to the UAV-mounted MEC servers (Fig. 11), their satisfaction utility is lower and their expected overhead from the UAV-mounted MEC servers is higher (Fig. 12). Regarding the risk seeking users, they tend to offload more data to the UAV-mounted MEC servers resulting in high probability of failure (Fig. 11), thus making the overall system unstable and prone to failure.

Furthermore, in Fig. 11, it is observed that the heterogeneous population led to higher levels of UAV-mounted MEC servers' congestion compared to the homogeneous population, as both the average amount of offloaded data to the UAV-mounted MEC servers and the corresponding average probability of failure of the latter ones increase. In Fig. 12, it is shown that the heterogeneous users, by offloading more data to the UAV-mounted MEC servers, they experience a greater satisfaction utility and a lower expected overhead.

### C. Comparative Analysis

In this section, a detailed comparative evaluation of the proposed framework is performed against five other alternative data offloading strategies: (i) Non prospect-theoretic (NonProsTheor) - users minimize their expected overhead by the UAV-mounted MEC servers via determining their best response strategy  $\mathbf{b}_i^*$ , (ii) Full Game-theoretic Offloading

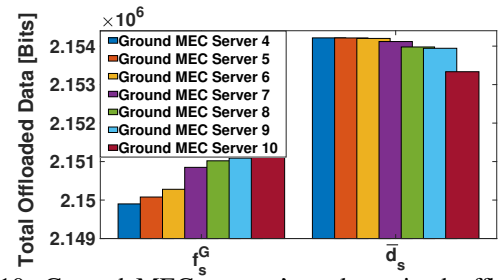


Fig. 10: Ground MEC servers' total received offloaded data as a function of their computation capability  $F_s^U$  and average distance  $\bar{d}_s$  from the users

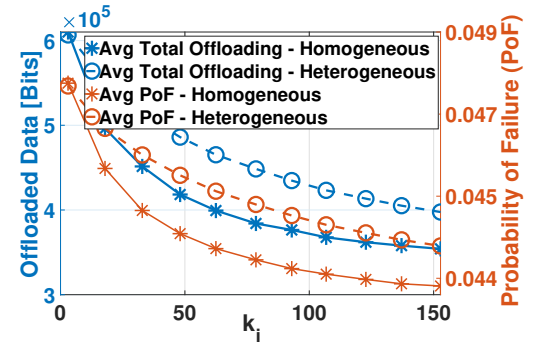


Fig. 11: Users' average offloaded data and UAV-mounted MEC servers' probability of failure as a function of their loss aversion parameter  $k_i$

(FullGameOff) - each user offloads the whole amount of its data to one UAV-mounted MEC server through a formulation of a non-cooperative game in order to minimize its expected overhead, (iii) Single UAV-mounted MEC servers environment (SingleUAV) - characterized by the average capabilities of all the UAV-mounted MEC servers, (iv) Each user processes all its data locally (LocalExec), (v) Each user determines randomly its data offloading strategy (Random).

Fig. 13 and Fig. 14 illustrate the user's average expected overhead and the UAV-mounted MEC servers' average probability of failure, respectively, for each of the aforementioned approaches. It is evident that our proposed data offloading approach achieves the best results while the SingleUAV, LocalExec and Random demonstrate the worst performance. Specifically, in the LocalExec approach, the users experience the highest expected overhead, as they process their computation task locally. In the Random approach, the users offload partially their data to randomly selected MEC servers (UAV-mounted or ground MEC servers), thus, even if the users experience a lower expected overhead than the LocalExec approach, the probability of the UAV-mounted MEC servers' failure remains high. Regarding the SingleUAV approach, the users offload their data to the single UAV-mounted MEC server and share its computational capabilities. Thus, they experience a higher expected overhead and a greater probability of failure (Fig. 13, 14) compared to the non prospect-theoretic and the full game-theoretic data offloading approaches.

The Non prospect-theoretic approach achieves the second best performance after our proposed framework, as the users partially offload their data to the UAV-mounted MEC servers



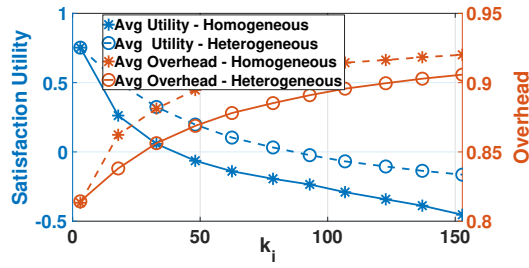


Fig. 12: Users' average satisfaction utility and overhead as a function of their loss aversion parameter  $k_i$

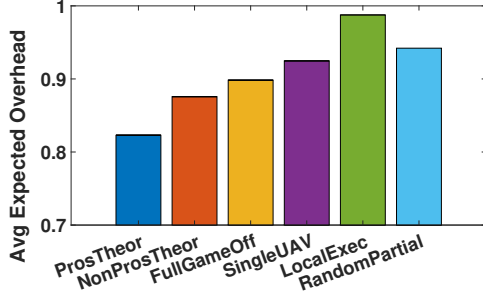


Fig. 13: Users' average expected overhead for different comparative scenarios

and they aim to minimize their expected overhead. However, they do that in an agnostic manner with respect to the guaranteed performance that they could get if they execute their applications in the safe resources, i.e., in the ground MEC servers and in their mobile devices. On the contrary, our prospect-theoretic framework results in lower average probability of failure and average expected overhead, by taking these aspects into consideration during the decision-making process. Finally, in the Full Game-theoretic Offloading, the users select a UAV-mounted MEC server to offload their whole computation task, without taking advantage of the partial offloading to multiple UAV-mounted MEC servers, thus concluding to a higher probability of failure compared to the Non prospect-theoretic approach.

#### D. Performance Analysis for different User Topologies

In this section, we further examine the performance of the proposed framework for different and varying topological characteristics, and in particular with reference to the increasing number of users, as well as to their position distribution within the examined environment. Specifically, Fig. 15 shows the users' average expected overhead and the corresponding actual execution time of the DCP algorithm as a function of the number of users in the examined system. The results reveal that for a five-fold increase in the number of users (i.e. from 200 to 1000 users), the corresponding average expected overhead that the users experience, increases by approximately 13%. This slight increase is owed to the fact that the ground and the UAV-mounted MEC servers are required to process more computation tasks (offloaded by the users), thus, they become more congested in terms of computation processing. Based on these results, we observe that the proposed framework achieves to serve the users in

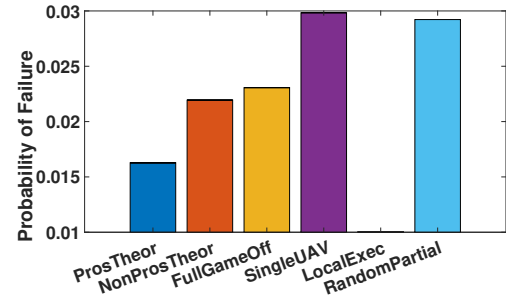


Fig. 14: UAVs' probability of failure for different comparative scenarios

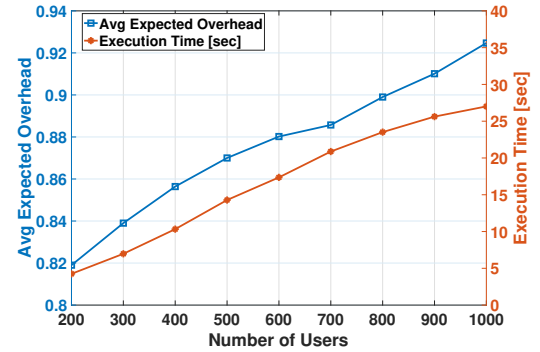


Fig. 15: Avg. expected overhead and execution time with respect to increasing number of users

a satisfactory manner, even when considering a large scale computing environment. Moreover, it is noted that this is achieved while noticing approximately a five-fold increase in the corresponding execution time of the DCP algorithm, essentially demonstrating an almost linear increase of the execution time with respect to the number of users.

Additionally, Fig. 16 illustrates the users' average expected overhead and the execution time of the proposed framework, for different topological characteristics. We focus on investigating our proposed framework's behavior with respect to different users' position distributions within the two-dimensional grid, while still maintaining the aforementioned base experimental setting, i.e.,  $U = 200, G = 7, F = 3$ . In particular, except from the users' random and uniform position distribution scenario, we also consider several Poisson distributions with different values of variance, i.e.,  $\lambda$  parameter. The corresponding results reveal that the DCP algorithm execution presents a stable behavior and performance, as indicated by the fact that the execution time is rather insensitive to the users' position distribution. Furthermore, it is observed that as the users are distributed more closely to each other, as reflected by lower values in the Poisson parameter  $\lambda$ , their average expected overhead increases. The latter phenomenon is due to the fact that the more closely among each other are the users distributed, they tend to have similar distances from the MEC servers, thus, making similar offloading decisions, and accordingly over-congesting the corresponding servers that are close to them. The opposite holds true for larger values of the Poisson parameter  $\lambda$ ,

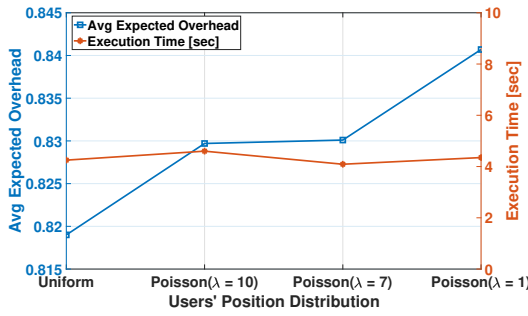


Fig. 16: Avg. expected overhead and execution time with respect to the users' position distribution

### E. Discussion and Guidelines

In this following, insights and guidelines regarding the operation and key features of the proposed framework are summarized, highlighting the user and system points of view.

1. (*Users' perspective*) The proposed framework enables the users to satisfy their energy and latency requirements, maximize their satisfaction utility, and converge to a stable data offloading equilibrium within few iterations. It is demonstrated that the users' physical and risk-aware characteristics have a significant impact on their data offloading decisions. Specifically, the users tend to offload more data to the UAV-mounted MEC servers, if they (i) are less distant from them; (ii) have stricter energy and latency requirements; (iii) present more risk seeking behavior; and (iv) have low local computing capability. The more data the users offload to the UAV-mounted MEC servers, the greater is their satisfaction utility, except for the cases of (i) having relaxed latency and energy requirements, where the local processing is more beneficial, and (ii) having a large amount of data to process, where inevitably a large portion of them will be processed locally resulting in low satisfaction utility.

2. (*System's perspective*) The UAV-mounted and ground MEC servers receive more data, if they have high computation capability and small average distance from the users. Also, increased amount of data is received by the UAV-mounted MEC servers if their operational threshold (i.e., amount of data that they can concurrently process) is high, in which case they present high robustness to failure. The more data the UAV-mounted MEC servers receive, the higher is their probability of failure and the energy consumption to process them.

3. The more loss averse the users are, the more data they process locally, the less satisfaction utility they perceive, the more overall overhead they experience, and the less they contribute to the UAV-mounted MEC servers' failure, as they exhibit a conservative data offloading behavior.

4. The users' heterogeneity in their loss averse behavior increases the UAV-mounted MEC servers' probability of failure.

5. The combined consideration of the (i) users' physical and risk-aware characteristics, (ii) UAV-mounted and ground MEC servers characteristics, (iii) users' distributed and autonomous decision-making, and (iv) users' ability to partially offload their data to multiple MEC servers (while process part of them

locally on their devices), concludes to superior data offloading strategies, users' satisfaction, and sophisticated system's resources exploitation, compared to other alternative approaches.

## VII. CONCLUSIONS

In this paper, a novel approach towards determining the user optimal data offloading strategy within a complex MEC environment consisting of both ground MEC servers and UAV-mounted MEC servers is introduced. Given the inherent computing uncertainty introduced, the UAV-mounted MEC servers are treated as CPRs, and the users act as prospect theoretic decision-makers, aiming to maximize their perceived prospect theoretic utility, while at the same time minimize the time and energy overhead by the ground MEC servers and the local execution. Accordingly, the risk-aware data offloading problem is formulated as a non-cooperative game among the users and the existence and uniqueness of the corresponding Pure Nash Equilibrium point (PNE) is proven. A low complexity distributed algorithm converging to the PNE is introduced, while detailed numerical results that demonstrate our framework's operation and superiority are presented.

Our current and future work focuses on studying the task offloading computation problem under a variety of probability of failure functions, e.g., Poisson process of the arrival data from all users, that also capture the uncertainty stemming from the rapidly changing communication environment in a metropolitan area. Moreover, we are interested in investigating the combination of the aforementioned framework with the optimal placement of the UAV-mounted and ground MEC servers, by considering several factors and aspects, such as coverage area, overall energy availability of the UAV-mounted MEC servers, computation capabilities, UAVs mobility, etc.

## ACKNOWLEDGMENT

The research of Dr. Tsiropoulou and Mr. Fragkos was conducted as part of the NSF CRII-1849739. The research of Dr. Papavassiliou was supported by the Hellenic Foundation for Research and Innovation (H.F.R.I.) under the "1st 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] M. Mozaffari, W. Saad, M. Bennis, Y.-H. Nam, and M. Debbah, "A tutorial on uavs for wireless networks: Applications, challenges, and open problems," *IEEE Communications Surveys & Tutorials*, 2019.
- [2] S. Jeong, O. Simeone, and J. Kang, "Mobile edge computing via a uav-mounted cloudlet: Optimization of bit allocation and path planning," *IEEE Trans. on Vehic. Techn.*, vol. 67, no. 3, pp. 2049–2063, 2017.
- [3] ETSI, *Multi-access Edge Computing (MEC); Phase 2: Use Cases and Requirements*. RGS/MEC-0002v211TechReq, ETSI, 2018.
- [4] D. He, Y. Qiao, S. Chan, and N. Guizani, "Flight security and safety of drones in airborne fog computing systems," *IEEE Communications Magazine*, vol. 56, no. 5, pp. 66–71, 2018.
- [5] F. Luo, C. Jiang, S. Yu, J. Wang, Y. Li, and Y. Ren, "Stability of cloud-based uav systems supporting big data acquisition and processing," *IEEE Transactions on Cloud Computing*, vol. 7, no. 3, pp. 866–877, 2017.

- [6] N. Cheng, W. Xu, W. Shi, Y. Zhou, N. Lu, H. Zhou, and X. Shen, "Air-ground integrated mobile edge networks: Architecture, challenges, and opportunities," *IEEE Com. Mag.*, vol. 56, no. 8, pp. 26–32, 2018.
- [7] R. Valentino, W.-S. Jung, and Y.-B. Ko, "Opportunistic computational offloading system for clusters of drones," in *20th Intern. Conf. on Advanced Communication Technology*. IEEE, 2018, pp. 303–306.
- [8] Z. Yu, Y. Gong, S. Gong, and Y. Guo, "Joint task offloading and resource allocation in uav-enabled mobile edge computing," *IEEE Internet of Things Journal*, 2020.
- [9] R. Wang, Y. Cao, A. Noor, T. A. Alamoudi, and R. Nour, "Agent-enabled task offloading in uav-aided mobile edge computing," *Comp. Comm.*, vol. 149, pp. 324–331, 2020.
- [10] J. Xiong, H. Guo, and J. Liu, "Task offloading in uav-aided edge computing: Bit allocation and trajectory optimization," *IEEE Communications Letters*, vol. 23, no. 3, pp. 538–541, 2019.
- [11] S. Jeong, O. Simeone, and J. Kang, "Mobile cloud computing with an uav-mounted cloudlet: Optimal bit allocation for communication and computation," *IET Communications*, vol. 11, pp. 969–974, 2016.
- [12] G. Wu, Y. Miao, Y. Zhang, and A. Barnawi, "Energy efficient for uav-enabled mobile edge computing networks: Intelligent task prediction and offloading," *Computer Communications*, vol. 150, pp. 556–562, 2020.
- [13] M. Li, N. Cheng, J. Gao, Y. Wang, L. Zhao, and X. Shen, "Energy-efficient uav-assisted mobile edge computing: Resource allocation and trajectory optimization," *IEEE Trans. on Vehicular Technology*, vol. 69, no. 3, pp. 3424–3438, 2020.
- [14] F. Zhou, Y. Wu, H. Sun, and Z. Chu, "Uav-enabled mobile edge computing: Offloading optimization and trajectory design," in *IEEE Intern. Conference on Communications*. IEEE, 2018, pp. 1–6.
- [15] F. Zhou, Y. Wu, R. Q. Hu, and Y. Qian, "Computation rate maximization in uav-enabled wireless-powered mobile-edge computing systems," *IEEE JSAC*, vol. 36, no. 9, pp. 1927–1941, 2018.
- [16] P. Vamvakas, E. E. Tsiropoulou, and S. Papavassiliou, "Dynamic spectrum management in 5g wireless networks: A real-life modeling approach," in *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*. IEEE, 2019, pp. 2134–2142.
- [17] —, "On the prospect of uav-assisted communications paradigm in public safety networks," in *IEEE INFOCOM WKSHPS: WCNEE 2019: Wireless Comm. and Net. in Extreme Env.* IEEE, 2019, pp. 1–6.
- [18] D. Kahneman and A. Tversky, "Prospect theory: An analysis of decision under risk," in *Handbook of the fundamentals of financial decision making: Part I*. World Scientific, 2013, pp. 99–127.
- [19] M. Lewandowski, "Prospect theory versus expected utility theory: Assumptions, predictions, intuition and modelling of risk attitudes," *Central European Journal of Economic Modelling and Econometrics*, pp. 275–321, 2017.
- [20] M. Kwon, E. Biyik, A. Talati, K. Bhasin, D. P. Losey, and D. Sadigh, "When humans aren't optimal: Robots that collaborate with risk-aware humans," in *Proceedings of the 2020 ACM/IEEE International Conference on Human-Robot Interaction*, 2020, pp. 43–52.
- [21] C. Zhang, Y.-x. Tian, L.-n. Fan, and Y.-h. Li, "Customized ranking for products through online reviews: a method incorporating prospect theory with an improved vikor," *Applied Intelligence*, pp. 1–20, 2020.
- [22] L. Tang and S. He, "Multi-user computation offloading in mobile edge computing: A behavioral perspective," *IEEE Network*, vol. 32, no. 1, pp. 48–53, 2018.
- [23] P. A. Apostolopoulos, E. E. Tsiropoulou, and S. Papavassiliou, "Cognitive data offloading in mobile edge computing for internet of things," *IEEE Access*, pp. 55 736–55 749, 2020.
- [24] P. Vamvakas, E. E. Tsiropoulou, and S. Papavassiliou, "On controlling spectrum fragility via resource pricing in 5g wireless networks," *IEEE Networking Letters*, vol. 1, no. 3, pp. 111–115, 2019.
- [25] L. Xiao, J. Liu, Y. Li, N. B. Mandayam, and H. V. Poor, "Prospect theoretic analysis of anti-jamming communications in cognitive radio networks," in *2014 IEEE Global Comm. Conf.*, 2014, pp. 746–751.
- [26] P. Vamvakas, E. E. Tsiropoulou, and S. Papavassiliou, "Exploiting prospect theory and risk-awareness to protect uav-assisted network operation," *EURASIP Journal on Wireless Communications and Networking*, vol. 2019, no. 1, p. 286, 2019.
- [27] A. Thanou, E. E. Tsiropoulou, and S. Papavassiliou, "Quality of experience under a prospect theoretic perspective: A cultural heritage space use case," *IEEE Transactions on Computational Social Systems*, vol. 6, no. 1, pp. 135–148, 2019.
- [28] F. Zhou, R. Q. Hu, Z. Li, and Y. Wang, "Mobile edge computing in unmanned aerial vehicle networks," *IEEE Wireless Communications*, vol. 27, no. 1, pp. 140–146, 2020.
- [29] G. Hardin, "Extensions of the tragedy of the commons," *Science*, vol. 280, no. 5364, pp. 682–683, 1998.
- [30] J. Wang, D. Feng, S. Zhang, J. Tang, and T. Q. Quek, "Computation offloading for mobile edge computing enabled vehicular networks," *IEEE Access*, vol. 7, pp. 62 624–62 632, 2019.
- [31] Y. Jin and H. Lee, "On-demand computation offloading architecture in fog networks," *Electronics*, vol. 8, no. 10, p. 1076, 2019.
- [32] F. Guo, H. Zhang, H. Ji, X. Li, and V. C. Leung, "An efficient computation offloading management scheme in the densely deployed small cell networks with mobile edge computing," *IEEE/ACM Transactions on Networking*, vol. 26, no. 6, pp. 2651–2664, 2018.
- [33] X. Diao, J. Zheng, Y. Wu, and Y. Cai, "Joint computing resource, power, and channel allocations for d2d-assisted and noma-based mobile edge computing," *IEEE Access*, vol. 7, pp. 9243–9257, 2019.
- [34] 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, 2015.
- [35] X. Chen, "Decentralized computation offloading game for mobile cloud computing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 4, pp. 974–983, 2014.
- [36] P. Barham, B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, R. Neugebauer, I. Pratt, and A. Warfield, "Xen and the art of virtualization," *ACM SIGOPS operating systems review*, vol. 37, no. 5, pp. 164–177, 2003.
- [37] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "Mobile edge computing: Survey and research outlook," *arXiv preprint arXiv:1701.01090*, 2017.
- [38] A. Tversky and D. Kahneman, "Advances in prospect theory: Cumulative representation of uncertainty," *Journal of Risk and uncertainty*, vol. 5, no. 4, pp. 297–323, 1992.
- [39] P. P. Wakker and J. Yang, "A powerful tool for analyzing concave/convex utility and weighting functions," *Journal of Economic Theory*, vol. 181, pp. 143–159, 2019.
- [40] C. H. Papadimitriou, "The complexity of finding nash equilibria," *Algorithmic game theory*, vol. 2, p. 30, 2007.
- [41] P. Dubey, O. Haimanko, and A. Zapechelnyuk, "Strategic complements and substitutes, and potential games," *Games and Economic Behavior*, vol. 54, no. 1, pp. 77–94, 2006.
- [42] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.
- [43] J. B. Rosen, "Existence and uniqueness of equilibrium points for concave n-person games," 1964.
- [44] C. Kao, "Performance of several nonlinear programming software packages on microcomputers," *Computers & operations research*, vol. 25, no. 10, pp. 807–816, 1998.
- [45] P. T. Boggs and J. W. Tolle, "Sequential quadratic programming," *Acta numerica*, vol. 4, pp. 1–51, 1995.
- [46] A. Grace, *Optimization Toolbox: for Use with MATLAB: User's Guide, November 1990*. MathWorks Incorporated, 1990.
- [47] X. Hu, K. Wong, K. Yang, and Z. Zheng, "Uav-assisted relaying and edge computing: Scheduling and trajectory optimization," *IEEE Transactions on Wireless Communications*, vol. 18, no. 10, pp. 4738–4752, 2019.
- [48] M. Li, N. Cheng, J. Gao, Y. Wang, L. Zhao, and X. Shen, "Energy-efficient uav-assisted mobile edge computing: Resource allocation and trajectory optimization," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 3, pp. 3424–3438, 2020.
- [49] A. A. Khuwaja, Y. Chen, N. Zhao, M.-S. Alouini, and P. Dobbins, "A survey of channel modeling for uav communications," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 4, pp. 2804–2821, 2018.
- [50] M.-H. Chen, B. Liang, and M. Dong, "Joint offloading and resource allocation for computation and communication in mobile cloud with computing access point," in *IEEE INFOCOM*. IEEE, 2017, pp. 1–9.
- [51] A. P. Miettinen and J. K. Nurminen, "Energy efficiency of mobile clients in cloud computing," *HotCloud*, vol. 10, no. 4-4, p. 19, 2010.
- [52] K. Sydsaeter, *Mathematics for economic analysis*. Pearson Education India, 2013.





**Pavlos Athanasios Apostolopoulos** is a PhD candidate and research assistant in the Department of Electrical and Computer Engineering, University of New Mexico. He received his Diploma in Electrical and Computer Engineering from the National Technical University of Athens in 2017. His main research interests include machine learning, deep learning, game theory, behavioral decision theory and optimization.



**Georgios Fragkos** is a PhD candidate and research assistant in the Department of Electrical and Computer Engineering, University of New Mexico. He received his Diploma in Electrical and Computer Engineering from the National Technical University of Athens in 2018. His main research interests include deep reinforcement learning, game theory, optimization, contract theory, and blockchain.



**Eirini Eleni Tsiropoulou** is currently an Assistant Professor at the Department of Electrical and Computer Engineering, University of New Mexico. Her main research interests lie in the area of cyber-physical social systems and wireless heterogeneous networks, with emphasis on network modeling and optimization, resource orchestration in interdependent systems, reinforcement learning, game theory, network economics, and Internet of Things. Five of her papers received the Best Paper Award at IEEE WCNC in 2012, ADHOCNETS in 2015, IEEE

IFIP WMNC 2019, INFOCOM 2019 by the IEEE Communications Systems Integration and Modeling Technical Committee, and IEEE/ACM BRAINS 2020. She was selected by the IEEE Communication Society - N2Women - as one of the top ten Rising Stars of 2017 in the communications and networking field. She has received the Early Career Award by the IEEE ComSoc Internet Technical Committee 2019.



**Symeon Papavassiliou** is currently a Professor in the School of ECE at National Technical University of Athens. From 1995 to 1999, he was a senior technical staff member at AT&T Laboratories, New Jersey. In August 1999 he joined the ECE Department at the New Jersey Institute of Technology, USA, where he was an Associate Professor until 2004. He has an established record of publications in his field of expertise, with more than 300 technical journal and conference published papers. His main research interests lie in the area of computer

communication networks, with emphasis on the analysis, optimization, and performance evaluation of mobile and distributed systems, wireless networks, and complex systems. He received the Best Paper Award in IEEE INFOCOM 94, the AT&T Division Recognition and Achievement Award in 1997, the US National Science Foundation Career Award in 2003, the Best Paper Award in IEEE WCNC 2012, the Excellence in Research Grant in Greece in 2012 and the Best Paper Awards in ADHOCNETS 2015, ICT 2016 and IFIP WMNC 2019. He also served on the board of the Greek National Regulatory Authority on Telecommunications and Posts from 2006 to 2009.