Risk-Aware Data Offloading in Multi-Server Multi-Access Edge Computing Environment

Pavlos Athanasios Apostolopoulos, *Student Member, IEEE*, Eirini Eleni Tsiropoulou, *Member, IEEE*, and Symeon Papavassiliou, *Senior Member, IEEE*

***Abstract*— Multi-access Edge Computing (MEC) has emerged as a flexible and cost-effective paradigm, enabling resource constrained mobile devices to offload, either partially or completely, computationally intensive tasks to a set of servers at the edge of the network. Given that the shared nature of the servers' resources introduces high computation and communication uncertainty, in this paper we consider users' risk-seeking or loss-aversion behavior in their final decisions regarding the portion of their computing tasks to be offloaded at each server in a multi-MEC server environment, while executing the rest locally. This is achieved by capitalizing on the power and principles of Prospect Theory and Tragedy of the Commons, treating each MEC server as a Common Pool of Resources available to all the users, while being rivarlous and subtractable, thus may potentially fail if over-exploited by the users. The goal of each user becomes to maximize its perceived satisfaction, as expressed through a properly formulated prospect-theoretic utility function, by offloading portion of its computing tasks to the different MEC servers. To address this problem and conclude to the optimal allocation strategy, a non-cooperative game among the users is formulated and the corresponding Pure Nash Equilibrium (PNE), i.e., optimal data offloading, is determined, while a distributed low-complexity algorithm that converges to the PNE is introduced. The performance and key principles of the proposed framework are demonstrated through modeling and simulation, while useful insights about the users' data offloading decisions under realistic conditions and behaviors are presented.**

***Index Terms*— Data offloading, Multi-access Edge Computing, computation and communication overhead, risk-based behavior, probabilistic uncertainty, utility functions, convex optimization.**

I. Introduction

THE rise of 5G networks alongside the Internet of Things (IoT) evolution, have skyrocketed the number of connected objects, which was around a few dozen billions in 2015 and is expected to experience a many-fold increase by 2020 [1], [2]. Within this setting, multiple heterogeneous devices, with a wide range of computational capabilities, are expected to execute various applications with different constraints and requirements. Despite the recent hardware advances in the smart devices, several of them are not yet capable of efficiently supporting computationally-intensive applications, as their local computation and energy resources appear still insufficient.

The concept of MEC was motivated by the unprecedented growth of mobile traffic, especially by the smart phones, and the emergence of enhanced multimedia services, which are characterized by high computing demands. The edge computing - representing the practice of processing data near the edge of the network - currently comes either as an alternative or complementary to the cloud computing paradigm, which suffers from latency issues due to the connection to remote servers in the cloud through public Internet. Thus, the MEC solution reduces the network congestion at the backhaul of the network, as the users' computing applications are offloaded at the edge servers via the access network, and therefore facilitates the execution of various types of computing hungry applications (either due to delay sensitivity requirements or due to high throughput demands). For the same reason, by adopting the MEC approach the propagation delay becomes negligible, while in addition large and unpredictable queuing and transmission delays are avoided, which would instead occur if data were offloaded to the central cloud, due to the congestion at the backhaul of the network. Also, the MEC reduces the privacy and security concerns compared to the central cloud that has a single point of failure. Consequently, the MEC environment, as considered in this paper, appears as the appropriate candidate to support both latency-sensitive applications (e.g., smart home, smart city, autonomous vehicles, virtual reality, industrial Internet of Things applications) and in general computational-intensive applications.

A. Related Work

Single-MEC server and *multi-MEC servers* approaches have been proposed in the literature to consider the computation and communication limitations in the MEC environment. Regarding the *single-MEC server* setup, Mao *et al.* [3] have assumed that the computation task requests from the mobile users arrive in a stochastic manner, and they formulated a power consumption minimization problem with task buffer stability constraints to examine the tradeoff between the mobile users' power consumption and the execution delay of the computation tasks. The decision regarding the local execution and computation offloading is based on Lyapunov optimization, while the communication resources, i.e., transmission power and bandwidth, are allocated following the Gauss-Seidel method.

Manuscript received September 24, 2019; revised January 24, 2020; accepted March 14, 2020; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor K. Tang. Date of publication April 23, 2020; date of current version June 18, 2020. The work of Eirini Eleni Tsiropoulou was supported in part by the NSF CRII Award under Grant 1849739. *(Corresponding author: Eirini Eleni Tsiropoulou.)*

Pavlos Athanasios Apostolopoulos and Eirini Eleni Tsiropoulou are with the Department of Electrical and Computer Engineering, The University of New Mexico, Albuquerque, NM 87131 USA (e-mail: pavlosapost@unm.edu; eirini@unm.edu).

Symeon Papavassiliou is with the School of Electrical and Computer Engineering, National Technical University of Athens, 15780 Zografou, Greece (e-mail: papavass@mail.ntua.gr).

This article has supplementary downloadable material available at http://ieeexplore.ieee.org, provided by the authors.

Digital Object Identifier 10.1109/TNET.2020.2983119

A similar problem is considered in [4] under the consideration of a multi-channel wireless interference environment. The authors propose a distributed approach to determine the users' computation offloading decisions based on game theory.

On the other hand, a centralized approach is introduced in [5], [6], targeting at the energy-efficient data offloading via jointly optimizing the computation offloading and the radio resource allocation for all the users in the network, in order to obtain the minimal energy consumption under the latency constraints in a single-MEC server environment. The same problem is studied in [7] under the assumption of mobile users' personalized delay requirements, which introduces additional constraints (as many as the number of users) in the corresponding optimization problem. This problem has been also extended in a MIMO multicell system [8], where multiple users offload their data to a single-MEC server. The formulated optimization problem is non-convex, thus the authors propose an iterative algorithm following the successive convex approximation technique to determine a local optimal data offloading and radio resource allocation.

In [9], the authors study the workload balancing problem in a fog network to minimize the latency of data flows in the communications and processing procedures by associating mobile devices to suitable base stations. A hierarchical computing infrastracture is proposed in [10] consisting of shallow and deep cloudlets and the authors study the problem of users' data offloading to reduce the latency and improve the quality of service based on a queuing theory analysis. In [11], the authors examine the joint optimization problem of minimizing the system cost in terms of leasing virtual machines for computing purposes, while guaranteeing QoS requirements, and they address it as a mixed integer nonlinear programming problem. In [12], the author provides a techno-economic analysis via proposing a coalitional game-based pricing scheme to study the users' data offloading problem.

The centralized partial data offloading problem, while the mobile users can harvest energy from the environment, is studied in [13] and [14] based on linear programming and Lyapunov optimization, respectively, towards determining the optimal policies of offloading decision, clock frequency control, power splitting ratio and transmission power allocation. In [15], the authors focus their study on the communication collisions at the shared network when multiple users offload their data to a single-MEC server. The authors aim to minimize the average application completion time following a mixed integer programming approach.

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

The problem of pure data offloading is studied in [16], where the authors aim to determine the amount of offloaded data to each MEC server (without considering the radio resource allocation) via formulating a multiple knapsack problem. Similarly, two separate problems are formulated in [17] regarding the mobile users' energy consumption minimization and the minimization of application's execution latency. Both problems are non-convex ones and the authors transform the first problem to a convex one based on the variable substitution technique, while they propose a locally optimal algorithm with the univariate search technique to address the second one. The joint data offloading and radio resource allocation problem has been recently studied in [18] considering that the mobile users can harvest energy from the surrounding environment.

It is noted that, all the aforementioned approaches, whether considering a *single-MEC server* or *multi-MEC servers*, assume that the users have a risk-neutral behavior, acting as neutral maximizers that aim to maximize their payoff from the allocation of the communication and computation resources. However, in real life the individuals tend to exhibit risk-seeking or loss-aversion behavior under the presence of uncertainty (in terms of both computation and communication), which is a key property of the MEC environment.

B. Contributions & Outline

In this paper, towards exactly filling the aforementioned research gap, we exploit Prospect Theory [19] to account for users' risk-seeking and loss-aversion behavior [20], in their data offloading decisions. This comes in contrast to the majority of the relevant literature that considers risk-neutral users and classical utility maximizers [21]. In particular, in our work we address the data offloading problem under the uncertainties of a realistic *multi-MEC servers system*, consisting of servers with certain capabilities, e.g., storage, computing. This is achieved through a risk-based distributed framework that properly captures users' behavior under losses and gains, in a formal but still pragmatic manner.

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

1. A heterogeneous *multi-users* and *multi-MEC servers* environment is introduced, where each user can offload arbitrarily parts of its application to multiple MEC servers for remote execution. This goes beyond the majority of current literature, that primarily addresses the problem of binary offloading, i.e., each user may offload its whole application to one MEC server.

2. The users determine, under risk, the computation load to be offloaded at each MEC server, taking into consideration both the computation uncertainty (limited computation capability) and the communication uncertainty (interference) at each MEC server, due to its shared nature among the users. Each MEC server is treated as a common pool of resources (CPR) following the principles of the Tragedy of the Commons [22].

3. Each user's perceived prospect-theoretic utility by a MEC server, is properly formulated by considering its actual edge computing overhead that the user experiences via offloading part of its application to the MEC server, and the corresponding overhead that would be involved if instead processed the same amount locally. It is noted that the prospect-theoretic utility is of probabilistic nature, as the user's perceived satisfaction depends on the communication and computing congestion at the MEC servers, which are introduced as fragile computing resources that can fail to serve the users' computation demands. A corresponding probability of failure function is defined characterizing each MEC server and representing its probability to fail serving the end-users' computing requests

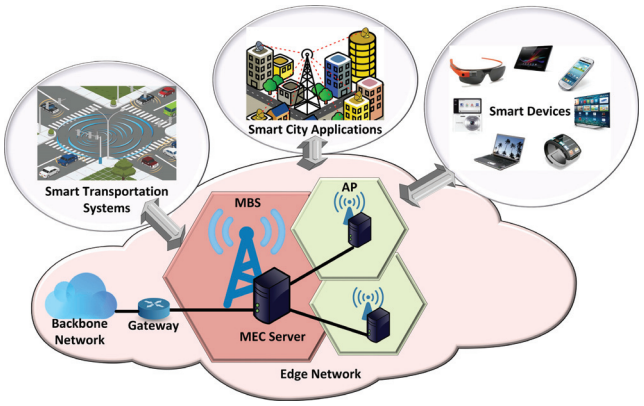


Fig. 1. Multi-MEC servers environment.

due to the over-exploitation of its computing capabilities. Also, the effective rate of return function is defined to quantify the user's perceived quality of service by a MEC server, while the return function expresses the user's assigned computational capability from a MEC server. Each user's satisfaction utility function is defined as the summation of the expected prospect-theoretic utility outcomes from the MEC servers, and the corresponding local computing overhead for the rest part of the user's application that is not offloaded.

4. The problem of each user determining in an autonomous manner the portion of its computation task that will be performed at each MEC server, has been formulated as a convex optimization problem of each user's satisfaction utility, and is treated as a non-cooperative game among the users. The respective non-cooperative game is solved in a distributed manner, and the existence and uniqueness of a Pure Nash Equilibrium (PNE) is proven. A distributed low-complexity algorithm that converges to the PNE is introduced.

5. A series of experiments are performed to evaluate the performance of the distributed framework, in terms of users' satisfaction regarding the inherent attributes of the proposed prospect-theoretic formulation and system's scalability. Furthermore, a detailed comparative evaluation with alternative decision-making scheme demonstrates our framework's superiority and benefits, in terms of devices' overhead and proper system's operation regarding the environment's uncertainty.

The remainder of this research paper is organized as follows. In Section II, the overall system model is described, while in Section III, the user's risk-aware behavior is captured and its prospect-theoretic utility function is formulated. In Section IV, the problem of risk-aware data offloading is formulated and treated as a non-cooperative game among the users, while the existence and uniqueness of a Pure Nash Equilibrium (PNE) is shown. In Section V, a low-complexity algorithm is introduced towards determining the PNE, following a convex optimization approach. Finally, a detailed performance evaluation of our approach is presented in Section VI via modeling and simulation, while Section VII concludes the paper.

II. System Model

A multi-access edge computing (MEC) system with *multi-MEC servers*, as shown in Fig. 1, is considered, where the users have the ability to offload part of their application to the MEC system through a 5G heterogeneous network. The MEC servers can be small data centers at the edge of the network and possibly managed by different Wireless Internet Service Providers (WISPs). Following the existing literature in the field of multi-access edge computing, the MEC servers reside at the Macro Base Stations (MBSs) of the macrocells or at the Access Points (APs) of the small cells, e.g., femtocells [5], [8]. Considering that typically the small cells and the macrocell are overlapping, each user is assumed capable of potentially offloading part of its data to all the MEC servers in the examined scenario. By offloading a portion of the user application data to the MEC system, the computing performance and energy consumption that the users observe, are significantly improved and reduced respectively, thus overall enhancing user experience. Data offloading decision is a dynamic process that depends not only on the the communication and computing environment, but also on the type of users' requested services. These services may impose different time and energy constraints in the data offloading problem based on their real and non-real time nature. Accordingly, the users may request either elastic or inelastic services, such as executing a machine learning algorithm for data analytics purpose or online gaming, respectively.

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

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

In a realistic multi-MEC servers system, the end-users sense their environment and available options of the MEC servers. The user devices are sufficiently intelligent and make optimal data offloading decisions in an autonomous and distributed manner, while expressing and considering the users' risk-aware behavioral characteristics. On the other hand, having a centralized load balancer to control the users' optimal data offloading, introduces several drawbacks in the system design and efficiency, which our proposed solution bypasses. *First*, the load balancer is a centralized decision-making entity, which is prone to be a single point of failure that can be attacked, e.g., distributed denial of service (DDoS) attacks, and the system can misoperate. *Second*, it is assumed that all the service operators owning the various MEC servers will

TABLE I
SUMMARY OF KEY NOTATIONS

Notation	Description
$\mathbb{U}$	Set of users
$\mathbb{S}$	Set of MEC servers
A_i	User's i computing application
B_i	Total input bits of user i
C_i	Number of CPU cycles required by user's computing application A_i
ϕ_i	User's i application's level of intensity [CPU-cycles/bit]
t_i	Time constraint of user i [sec]
e_i	Energy constraint of user i [J]
$b_{i,s}$	Offloaded data of user i to server s [bits]
$R_{i,s}$	Uplink data rate of user i to server s [bps]
W	System's bandwidth
$p_{i,s}$	Transmission power of user i to server s [W]
$g_{i,s}$	Channel gain between user i and server s
N_s	Set of users offloading to server s
σ_0^2	Variance of the Additive White Gaussian Noise
$O_{i,s}^{m,t}$	Overhead of the transmission time of the data [sec]
$O_{i,s}^{m,e}$	Overhead of the energy consumption due to transmission [J]
$\tilde{b}_s$	Total amount of data that server s can process [bits]
F_s	Total computing capability of server s [CPU-cycles/sec]
$F_{i,s}(\bar{b}_s)$	Comp. capability assigned to user i by server s [CPU-cycles/sec]
$\bar{b}_s$	Total amount of offloaded data to server s [bits]
f_s	Server's s production function
$O_{i,s}^{m,t}\|_{total}$	Total time overhead [sec]
$O_{i,s}^{m}$	Relative MEC overhead for user i offloading to server s
O_i^m	Overall MEC overhead for user i
L_i	Amount of data executed locally at the user's device [bits]
lc_i	User's i local computing capability [CPU-cycles/sec]
le_i	User's i local energy consumption [J/CPU-cycles]
$O_i^{l,t}$	Overhead of local computing execution time [sec]
$O_i^{l,e}$	Overhead of local energy consumption to process the data [J]
O_i^l	Relative overhead regarding the local computing approach
O_i	User's i total overhead
$p_s(\tilde{b}_s)$	Probability of failure of server s
α_i, γ_i	Sensitivity to the gains and losses of user i, respectively
k_i	Loss aversion parameter of user i
$u_{i,s}$	Prospect-theoretic utility of user i offloading to server s
s_i	User's i satisfaction utility
Γ_i	User's i strategy space

accept and trust the centralized load-balancer to control the data offloading to them. *Third*, even in the simple case of considering risk-neutral rational users, the users are burdened by signaling overhead in order to report their characteristics to the centralized load-balancer. *Fourth*, in the case of risk-aware users, as considered in the proposed framework, the centralized load-balancer has no feasible way to know the user's behavioral characteristics and the users are reluctant to reveal them due to privacy concerns. Based on the above description, we evangelize that a distributed risk-aware data offloading in multi-MEC server environments is a more realistic framework compared to a centralized approach.

A. Communication Model

Each AP/MBS operates and receives data over a dedicated communication link, i.e., frequency band, thus, each user, while transmitting part of its data to a MEC server, senses the interference from the rest of the users transmitting only to the same MEC server, i.e., $\sum_{j\in N_s, j\neq i} p_{j,s}\cdot g_{j,s}$, where the communication channel gain between the user j and the MEC server s is denoted by $g_{j,s}$, $N_s = \{j \in \mathbb{U} : b_{j,s} \neq 0\}$ is the set of users that offload part of their application to server s, and $p_{i,s}$ is the user's i transmission power to offload part of the data to server s. The signal-to-interference-plus-noise-ratio (SINR) measured at the receiver side, i.e., MEC servers, with respect to the transmission of user i is $\gamma_{i,s} = \frac{p_{i,s}\cdot g_{i,s}}{\sigma_0^2+\sum_{j\in N_s, j\neq i} p_{j,s}\cdot g_{j,s}}$, and given that the bandwidth allocated to each communication link is W, the corresponding user's achievable data rate, while communicating with server s, is [23]:

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

where σ_0^2 indicates the variance of the Additive White Gaussian Noise (AWGN) of the server s.

The user i, by offloading $b_{i,s}$ amount of data to the MEC server s, experiences an overhead consisting of: a) the transmission time [sec] of the data

$$O_{i,s}^{m,t} = \frac{b_{i,s}}{R_{i,s}} \quad (2)$$

and b) the transmission energy consumption [Joules]

$$O_{i,s}^{m,e} = \frac{b_{i,s}\cdot p_{i,s}}{R_{i,s}} \quad (3)$$

B. Computing Model

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

$$F_{i,s}(\bar{b}_s) = \frac{\phi_i}{\sum_{j\in N_s}\phi_j}\cdot f_s(\bar{b}_s) \quad (4)$$

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

$$f_s(\bar{b}_s) = \begin{cases} (1 - \frac{\bar{b}_s}{\tilde{b}_s})\cdot F_s, & \text{if } \bar{b}_s \leq \tilde{b}_s \\ 0, & \text{otherwise} \end{cases} \quad (5)$$

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

Proposition 1: Each MEC server's s, $s \in \mathbb{S}$, production function $f_s(\bar{b}_s)$, and each user's i, $i \in \mathbb{U}$ return function $F_{i,s}(\bar{b}_s)$, are strictly decreasing with respect to the total offloaded data $\bar{b}_s$ at the MEC server s.

The return function of user i (Eq. 4) is personalized based on the computing demand ϕ_i of its application, and due to Eq. 5 decreases as the total computing offloading $\bar{b}_s$ increases.

User i can execute remotely its offloaded data by receiving a computing capability $F_{i,s}$ from the server, and the corresponding execution time is $\frac{\phi_i\cdot b_{i,s}}{F_{i,s}}$. As a result, based on Eq. 2 user's i total time overhead is calculated as follows:

$$O_{i,s}^{m,t}|_{total} = \frac{b_{i,s}}{R_{i,s}} + \frac{\phi_i\cdot b_{i,s}}{F_{i,s}} \quad (6)$$

Based on Eq. 3 and Eq. 6 the relative MEC overhead that user i experiences by deciding to offload part of its application to the server s, considering both the user's application time constraints and the user's energy availability, is formulated as follows:

$$O_{i,s}^{m}(b_{i,s}) = \frac{\frac{b_{i,s}}{R_{i,s}} + \frac{\phi_i \cdot b_{i,s}}{F_{i,s}}}{t_i} + \frac{\frac{b_{i,s} \cdot p_{i,s}}{R_{i,s}}}{e_i} \quad (7)$$

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

$$O_i^m(\mathbf{b_i}) = \sum_{s \in \mathbb{S}} b_{i,s} \cdot \left(\frac{1}{R_{i,s} \cdot t_i} + \frac{\phi_i}{F_{i,s} \cdot t_i} + \frac{p_{i,s}}{R_{i,s} \cdot e_i}\right) \quad (8)$$

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

$$O_i^{l,t} = \frac{\phi_i \cdot L_i}{lc_i} \quad (9)$$

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

$$O_i^{l,e} = \phi_i \cdot L_i \cdot le_i \quad (10)$$

Based on Eq. 9 and Eq. 10, the user's relative overhead regarding the local computing approach considering both the computing time and the energy consumption overhead, is given as follows:

$$O_i^l(L_i) = \frac{O_i^{l,t}}{t_i} + \frac{O_i^{l,e}}{e_i} = \phi_i \cdot L_i \cdot \left(\frac{1}{t_i \cdot lc_i} + \frac{le_i}{e_i}\right) \quad (11)$$

C. Actual Total Overhead

Based on Eq. 8 and Eq. 11, user's i total overhead is given as follows:

$$O_i = \sum_{s \in \mathbb{S}} b_{i,s} \cdot \left(\frac{1}{R_{i,s} \cdot t_i} + \frac{\phi_i}{F_{i,s} \cdot t_i} + \frac{p_{i,s}}{R_{i,s} \cdot e_i}\right) + \phi_i \cdot L_i \cdot \left(\frac{1}{t_i \cdot lc_i} + \frac{le_i}{e_i}\right) \quad (12)$$

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

III. The Prospect of Data Offloading

A. Risk-Aware Behavior: The Tragedy of the Commons

In the *multi-MEC servers'* environment, each MEC server constitutes a Common Pool of Resources (CPR), since all the users are able to arbitrarily offload part of their applications to the MEC servers for remote execution. Due to Eq. 4 and Eq. 5 each MEC server s is rivarlous and subtractable, as the MEC server's computing capability is a shared resource among the users. Specifically, Eq. 5 denotes that the $\tilde{b}_s$ is the received bytes threshold value for each MEC server s, thus if $\bar{b}_s \geq \tilde{b}_s$ the MEC server s is considered unable to execute the receiving amount of applications at the same time, so it "fails". This phenomenon is well known in the literature as the Tragedy of the Commons [22]. As a result, in the case of the CPR's failure, it is more beneficial for the user i either to offload the $b_{i,s}$ amount of data to another MEC server, or process the data locally.

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

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

In this paper, we consider a linear probability of failure function for each MEC server s, thus $p_s(\bar{b}_s) = \frac{\bar{b}_s}{\tilde{b}_s}$, $\forall \bar{b}_s < \tilde{b}_s$, while $p_s(\bar{b}_s) = 1$, $\forall \bar{b}_s \geq \tilde{b}_s$ in order to represent a smooth potential failure of the MEC server, if its computing capabilities become over-exploited by the users. It should be noted however, that the provided mathematical analysis would follow exactly the same philosophy and pattern for any convex form of the probability of failure function, e.g., logarithmic or exponential (which essentially respects Assumption 1). The physical meaning of the logarithmic or exponential function compared to the linear probability of failure function is that the MEC server would be more sensitive to failure for the same total amount of offloaded data. Each user i with strategy $\mathbf{b_i}$, by offloading $b_{i,s}$ amount of data to the MEC server s, should consider the server's probability of failure p_s, since in the case that the server s "fails" to execute user's i amount of offloaded data, the user has to process the data locally. Given that the MEC server's s probability of failure is p_s, then the probability that the server survives and executes successfully the total received amount of offloaded data is accordingly $(1 - p_s)$. Based on Eq. 7, Eq. 11, and given p_s, user's i expected MEC overhead from the server s, is formulated as follows:

$$\begin{aligned}\mathbb{E}(O_{i,s}^m(b_{i,s})) &= (1 - p_s(\bar{b}_s)) \cdot O_{i,s}^m(b_{i,s}) \\ &\quad + p_s(\bar{b}_s) \cdot \left(O_i^l(b_{i,s}) + \frac{b_{i,s}}{R_{i,s} \cdot t_i} + \frac{b_{i,s} \cdot p_{i,s}}{R_{i,s} \cdot e_i}\right) \quad (13)\end{aligned}$$

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

Following the same reasoning as in Eq. 8, and applying the operation of expectation, the overall expected multi-access edge computing overhead that user i experiences is $\mathbb{E}(O_i^m(\mathbf{b_i})) = \sum_{s \in \mathbb{S}} \mathbb{E}(O_{i,s}^m(b_{i,s}))$. Consequently, taking the expectation of the overall MEC overhead, where the first term of Eq. 12 becomes $\mathbb{E}(O_i^m(\mathbf{b_i}))$, the user's i overall perceived

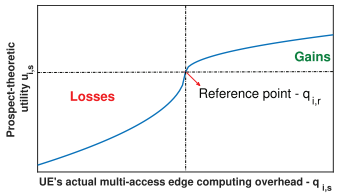


Fig. 2. Prospect-theoretic utility.

expected overhead is given as follows:

$$\mathbb{E}(O_i) = \sum_{s\in\mathbb{S}} \mathbb{E}(O^m_{i,s}(b_{i,s})) + \phi_i \cdot L_i \cdot (\frac{1}{t_i \cdot lc_i} + \frac{le_i}{e_i}) \quad (14)$$

where the overall local computing overhead remains the same (i.e., second term of both Eq. 12 and Eq. 14).

B. Offloading Decision Under Prospect Theory

To address the users' subjectivity in decision-making under uncertainty, as they tend to exhibit different decisions under losses or gains with respect to their actual satisfaction, Prospect Theory has been adopted. Prospect Theory was introduced by Kahneman and Tversky [19] and proposes a behavioral model, where humans make actions autonomously under risk and uncertainty. Following this behavioral model and applying it to the users' decision-making mechanism regarding the allocation of the amount of the offloading data $b_{i,s}$ to each MEC server s, users' relative MEC overhead as expressed in Eq. 7, is evaluated with respect to a reference point (*reference dependence property*). In our case, the reference point for each user is the guaranteed overhead $O^l_i(b_{i,s})$ (Eq. 11) that the user i can obtain by processing the $b_{i,s}$ amount of data locally instead of offloading them to the server s.

As it is shown in Fig. 2, the users' prospect-theoretic utility is a concave function with respect to the users' actual MEC overhead above the reference point, i.e., gains curve, while it is a convex function bellow it, i.e., losses curve (loss aversion property). Furthermore, the different slope in losses compared to the gains, depicts the fact that the users, weigh more the case where they experience a higher MEC overhead compared to the corresponding local computing overhead that they would have perceived if they had processed the data locally (*diminishing sensitivity property*).

C. Risk-Aware Utility Function

User's i prospect-theoretic utility, when offloading $b_{i,s}$ amount of data to the MEC server s is formulated as follows:

$$u_{i,s}(q_{i,s}) = \begin{cases} (q_{i,r} - q_{i,s})^{\alpha_i}, & \text{if } q_{i,s} \le q_{i,r} \\ -k_i \cdot (q_{i,s} - q_{i,r})^{\gamma_i}, & \text{if } q_{i,s} > q_{i,r} \end{cases} \quad (15)$$

where $q_{i,s} = O^m_{i,s}(b_{i,s})$ is the user's i actual perceived MEC overhead by offloading $b_{i,s}$ amount of data to the MEC server s, as defined in Eq. 7, and $q_{i,r} = O^l_i(b_{i,s})$ denotes the reference point of the user's i prospect-theoretic utility. It is noted that each user aims at maximizing its prospect-theoretic utility function. Regarding the first branch of Eq. 15, the maximization of the $u_{i,s}$ directly implies the minimization of the MEC overhead. On the other hand, at the second branch of Eq. 15 the MEC fails to serve the users' computing demands which are performed locally, thus, the maximization of the user's prospect-theoretic utility concludes to minimizing the additional overhead imposed to the user to offload its data.

The real parameters α_i, γ_i depict user's i sensitivity to the gains and losses of its actual perceived MEC overhead $q_{i,s}$, respectively. Small values of α_i capture user's i risk-seeking in losses and risk-averse in gains behavior, while small values of γ_i reflect a higher decrease in the user's prospect-theoretic utility, when its MEC overhead $q_{i,s}$ is close to its corresponding local computing overhead $q_{i,r}$. In this paper, without loss of generality, we consider similar behavior for all the users in losses and gains, i.e., $\alpha_i = \gamma_i$, $\forall i \in \mathbb{U}$. Furthermore, through the $k_i, k_i \in [0, +\infty)$ parameter, user i expresses if its losses weigh more than its gains. If $k_i > 1$, the prospect-theoretic utility $u_{i,s}$ has a greater slope of decrease in the case of losses compared to the slope of increase in the gains' part. The exact opposite holds true if $k_i \le 1$.

Considering the case that the MEC server s does not fail due to the users' offloaded amount of data, the user's i MEC overhead $q_{i,s}$ is calculated by Eq. 7, and in this case, it is lower than the corresponding local computing overhead $q_{i,r}$ (reference point), thus $q_{i,s} \le q_{i,r}$. As a result, based on user's i first branch of its prospect-theoretic utility (Eq. 15), via subtracting $q_{i,s}$ from the reference point $q_{i,r}$, we have $u_{i,s} = [b_{i,s}(\frac{\phi_i}{t_i lc_i} + \frac{le_i\phi_i}{e_i} - \frac{1}{t_i R_{i,s}} - \frac{\phi_i}{t_i F_{i,s}} - \frac{p_{i,s}}{e_i R_{i,s}})]^{\alpha_i}$. On the other hand, if the MEC server s fails to execute the received amount of offloaded data due to the fact that it is overloaded, then the user i has to process the $b_{i,s}$ amount of data locally, thus its experienced overhead is given by Eq. 11, while it has an extra communication overhead, as user i at first had to offload the $b_{i,s}$ amount of data to the MEC server s. As a result, user's i actual experienced overhead in the case of the MEC server's s failure consists of the local computing overhead $q_{i,r}$ (reference point) and the extra communication overhead, thus $q_{i,s} = q_{i,r} + \frac{b_{i,s}}{R_{i,s}\cdot t_i} + \frac{b_{i,s}\cdot p_{i,s}}{R_{i,s}\cdot e_i}$, and is greater than the reference point $q_{i,r}$. Therefore, based on the second branch of Eq. 15, by subtracting the reference point $q_{i,r}$ from user's i actual multi-access edge computing overhead $q_{i,s}$, its prospect-theoretic utility becomes $u_{i,s} = -k_i \cdot [b_{i,s} \cdot (\frac{1}{R_{i,s}\cdot t_i} + \frac{p_{i,s}}{R_{i,s}\cdot e_i})]^{a_i}$.

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

$$u_{i,s} = \begin{cases} b^{\alpha_i}_{i,s} \cdot h_{i,s}(\bar{b}_s), & \text{with probabil. } 1 - p_s(\bar{b}_s) \\ -k_i \cdot \epsilon_i \cdot b^{\alpha_i}_{i,s}, & \text{with probabil. } p_s(\bar{b}_s) \end{cases} \quad (16)$$

Based on Eq. 16 each user's i expected prospect-theoretic utility regarding MEC server s is formulated as follows:

$$\begin{aligned} \mathbb{E}(u_{i,s}) &= b^{\alpha_i}_{i,s} \cdot h_{i,s}(\bar{b}_s)(1 - p_s(\bar{b}_s)) - k_i\epsilon_i b^{\alpha_i}_{i,s} p_s(\bar{b}_s) \\ &\triangleq b^{\alpha_i}_{i,s} \cdot ert_{i,s}(\bar{b}_s) \end{aligned} \quad (17)$$

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

IV. Prospect-Theoretic Partial Offloading: A Game-Theoretic Approach

Each user i has to sophisticatedly and selfishly determine its best offloading strategy in order to maximize its overall perceived expected prospect-theoretic utility, i.e., $\sum_{s\in\mathbb{S}} \mathbb{E}(u_{i,s})$.

In this process there is a natural tradeoff between user's i overall MEC overhead and its overall local computing overhead. In order to capture this tradeoff, we introduce each user's i satisfaction utility, which is expressed by its overall expected prospect-theoretic utility subtracting its overall local computing overhead as follows.

$$s_i(\mathbf{b_i}, \mathbf{b_{-i}}) = \sum_{s\in\mathbb{S}} \mathbb{E}(u_{i,s}) - O_i^l(L_i) \tag{18}$$

where $\mathbf{b_{-i}} = [\mathbf{b_1} \ldots, \mathbf{b_{i-1}}, \mathbf{b_{i+1}}, \ldots, \mathbf{b_U}]$ is the users' offloading strategies' vector except for the user i, $\sum_{s\in\mathbb{S}} \mathbb{E}(u_{i,s})$ is the overall expected prospect-theoretic utility that user i obtains, and $O_i^l(L_i)$ is given by Eq. 11, where $L_i = B_i - \sum_{s\in\mathbb{S}} b_{i,s}$ is the amount of locally processed data.

Therefore, the ultimate goal of each user i is to maximize its perceived satisfaction utility s_i by determining its data offloading strategy $\mathbf{b_i}$. This problem can be formulated as a maximization problem of user's i satisfaction utility, and based on Eq. 11 and Eq. 17 can be expressed as follows.

$$\max_{\mathbf{b_i}\in\Gamma_i} s_i(\mathbf{b_i}, \mathbf{b_{-i}}) = \sum_{s\in\mathbb{S}} b_{i,s}^{\alpha_i} \cdot ert_{i,s}(\bar{b}_s) - \phi_i L_i (\frac{1}{t_i \cdot lc_i} + \frac{le_i}{e_i}) \tag{19}$$

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

Due to the non-cooperative and distributed nature of the above maximization problem, it can be treated as a non-cooperative game among the users who act as players making the optimal decisions about themselves in a selfish and distributed manner. Let $G = [\mathbb{U}, \{\Gamma_i\}_{i\in\mathbb{U}}, \{s_i\}_{i\in\mathbb{U}}]$ denote the non-cooperative game among the users which set is $\mathbb{U}$, where each user's strategy space is Γ_i, and its payoff is the satisfaction utility s_i (Eq. 18). Towards solving the non-cooperative game G, the concept of Nash equilibrium is adopted. The Nash equilibrium (NE) of the non-cooperative game G is the strategy vector which consists of users' offloading vectors, $\mathbf{b}^* = [\mathbf{b_1^*}, \ldots, \mathbf{b_i^*}, \ldots, \mathbf{b_U^*}]$, where no user has the incentive to change its own strategy (i.e., at least the amount of offloading data at one MEC server s) given the strategies of the rest of the users. Let $\mathbf{b_{-i}^*} = [\mathbf{b_1^*}, \ldots, \mathbf{b_{i-1}^*}, \mathbf{b_{i+1}^*}, \ldots, \mathbf{b_D^*}]$ denote the users' offloading strategies vector except for user i at the NE point.

Definition 1: The vector $\mathbf{b}^* = [\mathbf{b_1^*}, \ldots, \mathbf{b_i^*}, \ldots, \mathbf{b_U^*}] \in \Gamma$, $\Gamma = \Gamma_1 \times \cdots \times \Gamma_U$, is a Pure Nash Equilibrium (PNE) point of the non-cooperative game G, if $\forall i \in \mathbb{U}$ it holds true that $s_i(\mathbf{b_i^*}, \mathbf{b_{-i}^*}) \geq s_i(\mathbf{b_i}, \mathbf{b_{-i}^*})$, $\forall \mathbf{b_i} \in \Gamma_i$.

A. Problem Formulation

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

Additionally, each user aims at satisfying its time t_i and energy e_i constraints, as follows: $\mathbb{E}(O_i)|_t \leq t_i$ and $\mathbb{E}(O_i)|_e \leq e_i$, where $\mathbb{E}(O_i)|_t$ and $\mathbb{E}(O_i)|_e$ are given by Eq. 20 and Eq. 21, as shown at the bottom of this page, respectively. Therefore, the maximization problem of user's i satisfaction utility (Eq. 19-21) can be formulated as follows:

$$\begin{aligned} &\underset{\mathbf{b_i}\in\Gamma_i}{\text{maximize}}\, s_i(\mathbf{b_i}, \mathbf{b_{-i}}) \\ &\text{subject to } \left.\begin{aligned} &\sum_{s\in\mathbb{S}} b_{i,s} \leq B_i, \\ &\mathbb{E}(u_{i,s}) \geq 0, \quad \forall s \in \mathbb{S}, \\ &\mathbb{E}(O_i)|_t \leq t_i, \\ &\mathbb{E}(O_i)|_e \leq e_i \end{aligned}\right\} (C_i) \end{aligned} \tag{22}$$

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

B. Existence, Uniqueness and Convergence of PNE

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

Theorem 1: The non-cooperative game $\mathbb{G}$ among the users is an n-person concave game, where $n = U$.

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

Lemma 1: For each user i and each MEC server s there exists a value $b_{i,s}^{th} \geq 0$ such that $ert_{i,s}(b_{i,s}^{th}) = 0$ and $\mathbb{E}(u_{i,s}) \geq 0$, $\forall b_{i,s} \leq b_{i,s}^{th}$, while $\mathbb{E}(u_{i,s}) < 0$, $\forall b_{i,s} > b_{i,s}^{th}$.

Proof: See Appendix A in the Supplementary Material.

$$\begin{aligned} \mathbb{E}(O_i)|_t &= \sum_{s\in\mathbb{S}} \mathbb{E}(O_{i,s}^m)|_t + O_i^l|_t = \sum_{s\in\mathbb{S}} [\underbrace{b_{i,s}(\frac{1}{R_{i,s}} + \frac{\phi_i}{F_{i,s}})(1 - p_s(\bar{b}_s))}_{\text{multi-access edge computing time overhead}} + \underbrace{b_{i,s}(\frac{1}{R_{i,s}} + \frac{\phi_i}{lc_i})p_s(\bar{b}_s)}_{\text{local computing and transmission time overhead for } b_{i,s}}] + \frac{\phi_i L_i}{lc_i} \\ &\overset{(\text{Eq. 4 \& 5}) F_{i,s} = \frac{\phi_i}{\sum_{j\in N_s} \phi_j}(1 - \frac{\bar{b}_s}{\tilde{b}_s}) F_s}{\underset{p_s = \bar{b}_s/\tilde{b}_s, \bar{b}_s = b_{i,s} + b_{-i,s}}{=\!=\!=}} \sum_{s\in\mathbb{S}} [b_{i,s}(\frac{1}{R_{i,s}} + \frac{\sum_{j\in N_s}\phi_j}{F_s} + \frac{\phi_i}{lc_i}(\frac{b_{i,s} + b_{-i,s}}{\tilde{b}_s}))] + \frac{\phi_i L_i}{lc_i} \end{aligned} \tag{20}$$

$$\begin{aligned} \mathbb{E}(O_i)|_e &= \sum_{s\in\mathbb{S}} \mathbb{E}(O_{i,s}^m)|_e + O_i^l|_e = \sum_{s\in\mathbb{S}} [\underbrace{b_{i,s}\frac{p_{i,s}}{R_{i,s}}(1 - p_s(\bar{b}_s))}_{\text{multi-access edge computing energy overhead}} + \underbrace{b_{i,s}(\frac{p_{i,s}}{R_{i,s}} + \phi_i le_i)p_s(\bar{b}_s)}_{\text{local computing and transmission energy overhead for } b_{i,s}}] + \phi_i L_i le_i \\ &\overset{(\text{Eq. 4 \& 5}) F_{i,s} = \frac{\phi_i}{\sum_{j\in N_s} \phi_j}(1 - \frac{\bar{b}_s}{\tilde{b}_s}) F_s}{\underset{p_s = \bar{b}_s/\tilde{b}_s, \bar{b}_s = b_{i,s} + b_{-i,s}}{=\!=\!=}} \sum_{s\in\mathbb{S}} [b_{i,s}(\frac{p_{i,s}}{R_{i,s}} + \phi_i le_i(\frac{b_{i,s} + b_{-i,s}}{\tilde{b}_s})] + \phi_i L_i le_i \end{aligned} \tag{21}$$

Based on Lemma 1, the maximization problem in Eq. 22 can be rewritten as follows.

$$\begin{aligned} &\underset{\mathbf{b_i}\in A_i}{\text{maximize}} s_i(\mathbf{b_i},\mathbf{b_{-i}}) \\ &\text{subject to } \left.\begin{array}{l} \sum_{s\in\mathbb{S}} b_{i,s} \le B_i, \\ 0 \le b_{i,s} \le b_{i,s}^{th}, \quad \forall s \in \mathbb{S}, \\ \mathbb{E}(O_i)|_t \le t_i, \\ \mathbb{E}(O_i)|_e \le e_i \end{array}\right\} (C_i) \end{aligned} \tag{23}$$

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

Lemma 2: For each user i and each MEC server s, the expected prospect-theoretic utility $\mathbb{E}(u_{i,s})$ is strictly concave $\forall b_{i,s} \in (0, b_{i,s}^{th})$.

Proof: See Appendix B in the Supplementary Material.

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

Lemma 3: For each user i, its group of constraints (C_i) is a set of convex functions.

$$\begin{aligned} g_i^{(1)} &= \sum_{s\in\mathbb{S}} b_{i,s} - B_i \\ g_{i,s}^{(2)} &= b_{i,s} - b_{i,s}^{th}, \quad \forall s \in \mathbb{S} \\ g_{i,s}^{(3)} &= -b_{i,s}, \quad \forall s \in \mathbb{S} \\ g_i^{(4)} &= \mathbb{E}(O_i)|_t - t_i \\ g_i^{(5)} &= \mathbb{E}(O_i)|_e - e_i \end{aligned} \tag{24}$$

Proof: See Appendix C in the Supplementary Material.

Based on Lemma 3, for each user i, the set $\mathbb{C}_i = \Gamma_i \cap (\bigcap_{n_1\in\{1,4,5\}} Lev(g_i^{(n_1)}, 0)) \cap (\bigcap_{n_2\in\{2,3\}} Lev(g_{i,s}^{(n_2)}, 0)), \forall s \in \mathbb{S}$ is a convex set as an intersection of a convex set Γ_i and level sets of convex functions, which are necessarily convex sets (see Section 3.1.6 of [24]). Therefore, each user's i strategy space $\mathbb{A}_i = \Gamma_i \cap \mathbb{C}_i$ in the non-cooperative game $\mathbb{G}$, is a convex set as an intersection of convex sets.

Lemma 4: Each user's i, $i \in \mathbb{U}$ satisfaction utility s_i, is a concave function over the strategy space $\mathbb{A}_i$.

Proof: See Appendix D in the Supplementary Material.

Based on Lemmas 1-4, each user's i strategy space $\mathbb{A}_i$ is a convex set, and its satisfaction utility $s_i(\mathbf{b_i}, \mathbf{b_{-i}})$ is concave over the set $\mathbb{A}_i$. Thus, the non-cooperative game $\mathbb{G}$ is an n-person concave game, where $n = U$, so the Theorem 1 holds true. An n-person concave game has at least one Pure Nash Equilibrium (PNE) [25], thus the existence of at least one PNE point for the non-cooperative game $\mathbb{G}$ holds true.

Finally, based on Theorem 1, Lemma 4 and [25], the following Theorem proves the convergence of the users' strategies to the PNE.

Theorem 2: Consider the user i and an $S \times S$ matrix function $\mathbb{X}_i$ in which $(\mathbb{X}_i)_{ss'} = \lambda_i \frac{\partial^2 s_i}{\partial s \partial s'}$, $\forall s, s' \in \mathbb{S}$, and the constant choices $\lambda_i > 0$. Then, if $\mathbb{X}_i + \mathbb{X}_i^T$ is strictly negative definite, then the PNE of the game $\mathbb{G}$ is unique. Starting from any initial offloading strategy $\mathbf{b_0} = [\mathbf{b_{1,0}}, \cdots, \mathbf{b_{U,0}}]$, $\mathbf{b_{i,0}} \in \mathbb{A}_i$, the continuous Best Response (BR) dynamics converge to the unique PNE.

Proof: See Appendix E in the Supplementary Material.

V. Towards Determining the Equilibrium

A. A Convex Optimization Approach

Each user's i satisfaction utility s_i is a concave function over $\mathbb{A}_i$ (Lemma 4), thus the function $z_i(\mathbf{b_i}, \mathbf{b_{-i}}) = -s_i(\mathbf{b_i}, \mathbf{b_{-i}})$ is a convex function over the same space. Let us denote each user's i best response strategy $\mathbf{b_i^*}(\mathbf{b_{-i}}) : \mathbb{A}_{-i} \rightrightarrows \mathbb{A}_i$ considering the other users' strategies, as follows.

$$\mathbf{b_i^*}(\mathbf{b_{-i}}) = \underset{\mathbf{b_i}\in\mathbb{A}_i}{\arg\max}(s_i(\mathbf{b_i}, \mathbf{b_{-i}})), \mathbf{b_{-i}} \in \mathbb{A}_{-i} \tag{25}$$

where $\mathbf{b_{-i}}$ is the vector of the offloading strategies of all the users except user i, as it was defined in Section IV, and $\mathbb{A}_{-i} = \mathbb{A}_1 \times \cdots \times \mathbb{A}_{i-1} \times \mathbb{A}_{i+1} \times \cdots \times \mathbb{A}_U$ is the corresponding strategy space, thus $\forall i \in \mathbb{U}$, $\mathbf{b_{-i}} \in \mathbb{A}_{-i}$. Each user's i best response strategy $\mathbf{b_i} \in \mathbb{A}_i$ should satisfy the group of constraints (C_i) (Eq. 23). Furthermore, considering that the function z_i is a convex function over the convex set $\mathbb{A}_i$, each user's i best response strategy can be formulated as follows:

$$\mathbf{b_i^*}(\mathbf{b_{-i}}) = \underset{\mathbf{b_i}\in\mathbb{A}_i}{\arg\min}(z_i(\mathbf{b_i}, \mathbf{b_{-i}})), \mathbf{b_{-i}} \in \mathbb{A}_{-i} \tag{26}$$

Each user i in order to maximize its satisfaction utility s_i (Eq. 18), should equivalently minimize the convex function z_i over the convex set $\mathbb{A}_i$. Thus, each user i during the continues BR dynamics solves the following optimization problem to determine its best response strategy $\mathbf{b_i^*}$.

$$\begin{aligned} &\underset{\mathbf{b_i}\in\Gamma_i}{\text{minimize}} z_i(\mathbf{b_i},\mathbf{b_{-i}}) \\ &\text{subject to } \left.\begin{array}{l} \sum_{s\in\mathbb{S}} b_{i,s} \le B_i, \\ 0 \le b_{i,s} \le b_{i,s}^{th}, \ \forall s \in \mathbb{S}, \\ \mathbb{E}(O_i)|_t \le t_i, \\ \mathbb{E}(O_i)|_e \le e_i \end{array}\right\} (C_i) \end{aligned} \tag{27}$$

Moreover, assuming that each user i is able to satisfy its time and energy constraint in (C_i) by executing its whole application locally, i.e., $\mathbf{b_i} = \mathbf{0} \in \mathbb{A}_i$, the set $\mathbb{A}_i$ is non-empty, and the above minimization problem is a nonlinear convex optimization problem, where the function $z_i(\mathbf{b_i}, \mathbf{b_{-i}})$ is the objective function, and the $g_i^{(n_1)}$, $n_1 \in \{1, 4, 5\}$, $g_{i,s}^{(n_2)}$, $n_2 \in \{2, 3\}$ (Eq. 24) are the inequality constraints.

B. Algorithm and Complexity Analysis

In this section, the Distributed Algorithm for Convergence to the PNE (DACP) of the non-cooperative game $\mathbb{G}$ is presented. The DACP algorithm is a decision-making tool that runs at the beginning of the data offloading process and after it converges to $\mathbf{b^*} = [\mathbf{b_1^*}, \ldots, \mathbf{b_i^*}, \ldots, \mathbf{b_U^*}]$ the users know the data that should be offloaded to each MEC server and the ones that should be processed locally. The DACP algorithm is an iterative distributed sequential algorithm, where at each iteration only one randomly selected user plays an action. At the first iteration ($ite = 0$), each user selects randomly a feasible data offloading vector $\mathbf{b_i^*}, \forall i \in U$. Then, this is reported to the MEC servers by a user's broadcasting signal and each MEC server calculates the $\bar{b_s}, \forall s \in S$, which then it is broadcasted to all the users. At the next iteration of the DACP algorithm, one user is randomly selected to make an action $\mathbf{b_i^*}$ given the values $\bar{b_s}, \forall s \in S$. The user makes an action and broadcasts its decision to all the MEC servers in order the latter to recalculate the new values $\bar{b_s}, \forall s \in S$.

The same procedure is followed iteratively until the DACP algorithm converges (Line 15 of DACP algorithm). After the DACP algorithm converges, then each user has decided its data offloading vector $\mathbf{b_i^*}$ and performs the data offloading.

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

Each user has to solve the nonlinear optimization problem given in Eq. 27 in order to determine its best response strategy. Since, as we have already proven, the problem in Eq. 27 is a convex optimization problem, the constrained local minimum is also a constrained global minimum. Thus, each user may apply any of well known existing methods for solving constrained nonlinear optimization problems [26], and conclude to the global minimum of $z_i(\mathbf{b_i}, \mathbf{b_{-i}})$ (Eq. 27), while determining its best response strategy $\mathbf{b_i^*}$. For demonstration purposes, we consider the sequential quadratic programming (SQP) [27] method, that is also provided by the function *fmincon()* in the MATLAB Optimization Toolbox [28].

Regarding the Algorithm's DACP complexity, each user i, is required to determine the $b_{i,s}^{th}$, $\forall s\in\mathbb{S}$. Given that the complexity of the Binary Search into the interval $[0,\tilde{b}_s]$, $s\in\mathbb{S}$, is $\mathcal{O}(\log_2\tilde{b}_s)$ [29], the complexity of the user i to determine all the $b_{i,s}^{th}$ is $\mathcal{O}(S\cdot\log_2(\underset{s\in\mathbb{S}}{max}(\tilde{b}_s))$. Also, by denoting as $\mathcal{O}(\Delta)$ the complexity of the function *fmincon()*, and since the rest operations involve only algebraic calculations, the complexity of each user i to determine its best response $\mathbf{b_i^*}$ at each iteration *ite* is $\mathcal{O}(\Delta + S\cdot\log_2(\underset{s\in\mathbb{S}}{max}(\tilde{b}_s))$. Considering that U users execute the Algorithm DACP and given that *Ite* iterations are needed for convergence to the PNE, the total complexity of the distributed Algorithm DACP for all the users is $\mathcal{O}(U\cdot \textit{Ite}\cdot(\Delta + S\cdot\log_2(\underset{s\in\mathbb{S}}{max}(\tilde{b}_s))))$. Finally, the complexity of the optimization problem $\mathcal{O}(\Delta)$ can be considered significantly greater than the complexity $\mathcal{O}(S\cdot\log_2(\underset{s\in\mathbb{S}}{max}(\tilde{b}_s)))$, therefore the complexity of the Algorithm DACP is $\mathcal{O}(U\cdot \textit{Ite}\cdot\Delta)$.

In a nutshell, the DACP algorithm is a decision-making tool enabling the users to determine their optimal data offloading satisfying their personal constraints (C_i), as presented in Eq. 27, before they actually perform it. Also, as it is presented in Section VI.A, the DACP algorithm needs only few iterations (i.e., less than five iterations) in order to converge, thus the signaling overhead added to the end-users is rather limited, and in most cases practically insignificant.

Algorithm 1 DACP: Distributed Algorithm for Convergence to PNE

1: **Input:**
   $\Rightarrow$ *Set of users:* $\mathbb{U}=[1,\ldots,i,\ldots,U]$
   $\Rightarrow$ *Set of MEC Servers:* $\mathbb{S}=[1,\ldots,s,\ldots,S]$
2: **Output:**
   $\Rightarrow$*Profile Strategy at PNE:* $\mathbf{b^*}=[\mathbf{b_1^*},\ldots,\mathbf{b_i^*},\ldots,\mathbf{b_U^*}]$
3: **Initialization:**
   $\Rightarrow \mathbf{b_i}=[b_{i,1},\ldots,b_{i,s},\ldots,b_{i,S}]$
   $\Rightarrow ite=0$, $Convergence=0$
4: **Iterative Procedure:**
5: **while** *Convergence* $==0$ **do**
6: $ite=ite+1$;
7: $flag=0$;
8: **for** $i=1$ to U **do**
9: **for** $s=1$ to S **do**
10: *user i calculates the transmission uplink rate $R_{i,s}$*
11: $r_{i,s}^*$ = *BinarySearch(*$[0,\tilde{b}_s]$, ϵ*);*
12: $b_{i,s}^{th}=min(r_{i,s}^*, B_i)$;
13: **end for**
14: $\mathbf{b_i^*}$ = *fmincon();*
15: **if** $(|b_{i,s}^*-b_{i,s}|\leq\epsilon', \forall s\in\mathbb{S})$ **then**
16: $flag=flag+1$;
17: **end if**
18: $\mathbf{b_i}=\mathbf{b_i^*}$
19: **end for**
20: **if** $(flag==U)$ **then**
21: $Convergence=1$, $Ite=ite$;
22: **end if**
23: **end while**

VI. Numerical Results

In this section, we provide a detailed numerical performance evaluation of the proposed prospect-theoretic framework, through modeling and simulation, illustrating the operation, features and benefits of our approach. Specifically, in Section VI-A, we focus on the pure operational characteristics of our prospect-theoretic framework, in terms of efficiently controlling the users' offloaded data with respect to the heterogeneous multiple MEC server environment. In Section VI-B we provide a detailed study of our framework's operation under heterogeneous users regarding their loss-aversion characteristics. Furthermore, in Section VI-C a scalability and fragility evaluation study is shown with respect to an increasing number of users and MEC servers, while finally in Section VI-D, a comparative evaluation of our approach against alternative approaches and offloading strategies is provided.

In our study, we consider a set of $S=3$ heterogeneous MEC servers, with each MEC server s, $s\in\mathbb{S}$ having a coverage area of radius $R_s=100m$, and $U=50$ users in total. Each user's i, $i\in\mathbb{U}$ channel gain is modeled as $g_{i,s}=\frac{1}{d_{i,s}^{\theta}}$, where $d_{i,s}$ is the user's i distance from the MEC server s, i.e., $d_{i,s}\leq R_s$, and θ is the distance loss exponent, e.g., $\theta=2$. The system's transmission bandwidth is considered $W=5MHz$, while a representative value of the service uplink rate for video conference application is $R_{fix}=128$ kbps. Each user i transmits to the MEC server s with power $p_{i,s}=\frac{d_{i,s}^2}{R_s^2}$, thus each user's i transmission power is normalized and proportional to its distance from the corresponding MEC server. Moreover, for each user i we consider $lc_i\in[0.1,1]$ GHz and $le_i=10^{-9}\frac{Joules}{CPU-Cycle}$, $\forall i\in\mathbb{U}$ [30]. The considered application characteristics (e.g., face recognition application) are $B_i\in[1000,5000]$ KB and $C_i\in[1000,5000]$ Mega-Cycles. In the remaining of the

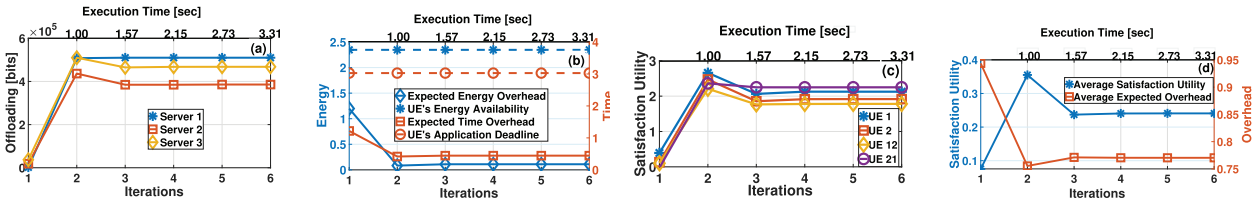


Fig. 3. Pure operation of the proposed framework - users' perspective.

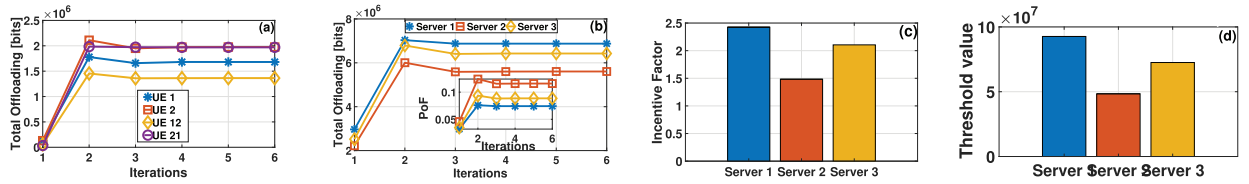


Fig. 4. Pure operation of the proposed framework - MEC servers' perspective.

paper, unless otherwise explicitly stated, we assume homogeneous users with prospect-theoretic parameters $\alpha_i = 0.2$ and $k_i = 5$, $\forall i \in \mathbb{U}$. Finally, for each MEC server s, $s \in \mathbb{S}$ we consider that $F_s \in [1,4] \cdot 10^3$ GHz and $\tilde{b}_s \in [30,70]\% \cdot \sum_{i=1}^{50} B_i$.

A. Pure Operation of the Algorithm

Fig. 3a presents the evolution of a specific user's offloading strategy ($b_{i,s}$, $\forall s \in \mathbb{S}$) at each MEC server s, as a function of the number of iterations and actual execution time needed for the Algorithm DACP to converge at the PNE point. Firstly, we observe that the user by starting from randomly selected feasible initial values, as the amounts of the data offloading at each MEC server, it converges in few iterations, i.e., less than five, at the unique PNE point. Indicatively we note that the DACP algorithm needs approximately $1.5sec$ to converge to the Pure Nash Equilibrium considering that 3 MEC servers and 50 users reside in the network, while significantly smaller times are observed if smaller-scale systems are considered or enhanced devices are utilized. Also, in our simulated scenarios, we have considered an indicative application with latency constraints $3sec$ (Fig. 3b) and based on the decision of the DACP algorithm a time overhead $\mathbb{E}(O_i)|_t = 50msec$ is achieved for the execution of the application.

As we also see in Fig. 4a, where four different users are considered, each user's total amount of offloaded data converges to a stable point, while the difference in the values of these points is due to the users' heterogeneous characteristics, e.g., users' application characteristics, users' location inside the system. Furthermore, as Fig. 3b illustrates, the examined user determines its best response strategy (in accordance to Eq. 26) by satisfying its energy and time constraint at every iteration, while at the same time its satisfaction utility converges to a stable point (Fig. 3c), as the user's data offloading strategy at each MEC server converges (Fig. 3a). Also, the propagation time is negligible in our presented numerical results, as the maximum distance of each user from each MEC server is $100m$. Moreover, the convergence of the users' average satisfaction utility and overhead are presented in Fig. 3d.

Fig. 4b presents the total amount of offloaded data that each MEC server collects by the users. MEC servers' heterogeneous characteristics, in terms of insetting the users to offload part of their applications to the MEC servers, are better captured by the incentive factor $\frac{\frac{\tilde{b}_s}{\sum_{j\in\mathbb{S}}\tilde{b}_j}+\frac{F_s}{\sum_{j\in\mathbb{S}}F_j}}{\frac{\sum_{i\in\mathbb{U}}d_{i,s}}{\sum_{j\in\mathbb{S}}\sum_{i\in\mathbb{U}}d_{i,j}}}$ of each MEC server, which is presented in Fig. 4c. The MEC server's incentive factor indicates that the higher the ability of a MEC server to process bigger amounts of data, i.e., $\tilde{b}_s$, or the higher the MEC server's computation capability, i.e., F_s, then the higher is the incentive of a user to offload part of its application to this MEC server in order to obtain an increased satisfaction utility. Consequently, the higher is the MEC server's incentive factor, the greater is the amount of data that it gathers by the users (Fig. 4b). Also note, that each MEC server's incentive factor is being influenced by the average distance of the users from the server, since for small distances the users will experience less communication overhead during their data offloading at this MEC server. Although MEC servers 1 and 3 collect higher amount of data compared to the server 2, the latter one concludes to a higher probability of failure (sub-figure within Fig. 4b). This phenomenon is observed since MEC server 1 and 3 are assumed to have a significantly higher threshold value $\tilde{b}_s$ than MEC server 2 (Fig. 4d), which enables them to process a higher amount of data.

In the following, we present some indicative results in order to study the tradeoffs in the users' offloading decisions with respect to the MEC servers' characteristics, i.e., threshold value $\tilde{b}_s$, computation capability F_s, and the average distance of the users from each MEC server. In particular, we assume a scenario where each user initially has the same distance from each MEC server, while the MEC server 3 has improved computation capabilities compared to the rest of the servers, in terms of its threshold value $\tilde{b}_s$ and its computation capability F_s. Then, the distance of the MEC server 3 from each user increases, thus, the users reduce their amount of data that they offload to the MEC server 3 (Fig. 5a). Specifically, the MEC server 3, due to its improved computation capability gathered a greater amount of data at the initial point, while then the amount of collected data decreases, as each user experiences a greater communication overhead due to the increase of its transmission power and time, which overturns the obtained computation benefit. Moreover, the users in order to reduce their additional local amount of data due to the decrease of their offloaded amount of data at the MEC server 3, they

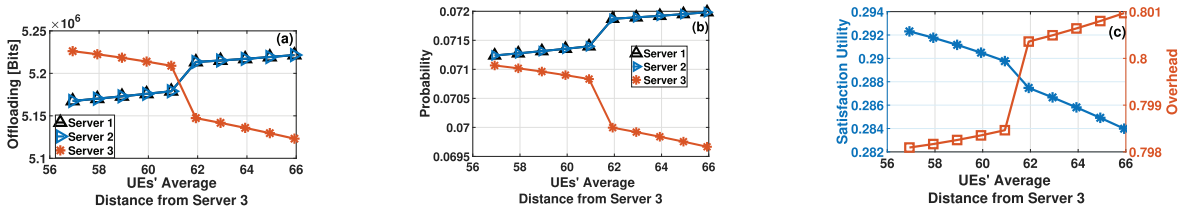


Fig. 5. Computation vs communication overhead.

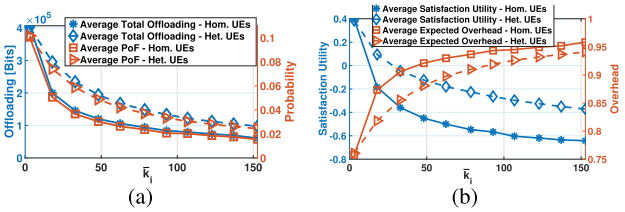


Fig. 6. Heterogeneous users - loss aversion impact study.

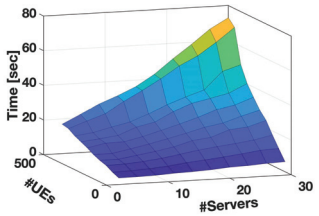


Fig. 7. Execution time vs no. of users and MEC servers.

increase their corresponding amounts to the rest MEC servers, and this lead the MEC servers 1 & 2 to receive greater amount of data compared to the MEC server 3 (Fig. 5a), after a specific point. As a result, the values of the probabilities of failure for the MEC servers 1 & 2 increase, while MEC server's 3 corresponding value decreases (Fig. 5b). Furthermore, due to the increase of MEC server's 3 distance from the users, each user's offloading is becoming less beneficial, and as a result its satisfaction utility decreases as it experiences a greater expected overhead (Fig. 5c).

B. Heterogeneous Users - Loss Aversion

In this section, initially the impact of the users' heterogeneous loss aversion prospect-theoretic behavior on the achievable performance is studied. Specifically, the results presented in Fig. 6a and Fig. 6b compare a scenario of homogeneous users (i.e., same loss aversion prospect-theoretic parameter $\bar{k}_i$ for all the users) against a heterogeneous scenario, where each user i, $i \in \mathbb{U}$ is associated with a different personalized loss aversion parameter k_i, while for fairness purposes the $\bar{k}_i$ loss aversion value for the homogeneous population is equal to the average among all the loss aversion values of the heterogeneous users. It is noted that the more loss averse is the users' behavior, the bigger is the loss aversion index k_i, $\forall i \in \mathbb{U}$, thus, the less amount of data the users offload to the MEC servers, and as a result the benefits that they obtain from their offloading, in terms of their experienced satisfaction utility and expected overhead, are lower. However, the opposite holds for the highly risk seeking users, which may lead the MEC servers to be overloaded, and thus, the users' perceived satisfaction utility will be decreased due to the high uncertainty of the system. In particular, as we observe in Fig. 6a the heterogeneous environment indeed led to higher congestion levels for the MEC servers, as both the average total amount of offloaded data by the users and the average MEC servers' probability of failure increase. However, Fig. 6b illustrates that the increase of the average amount of offloaded data, led the heterogeneous users to achieve a higher average satisfaction utility, and as a result a lower average expected overhead. On the other hand, the homogeneous users presenting a high loss averse behavior keep significantly higher amount of data for local execution, and as a result they experience a higher expected overhead and a lower satisfaction utility.

C. Scalability & Fragility Evaluation

Fig. 7 illustrates the necessary time for convergence to PNE, both for an increasing number of users and an increasing number of MEC servers. It is observed that our prospect-theoretic framework scales very well with respect to the increasing number of MEC servers, since the required execution time has a smaller increase rate when compared to the corresponding increase rate on the number of servers. Moreover, as the number of the users increases, the framework's execution time follows almost a linear increasing trend with respect to the number of users, and this indicates that the factor $\Delta \cdot Ite$ in our framework's complexity $\mathcal{O}(U \cdot Ite \cdot \Delta)$ (Section V.B) increases with a significantly lower rate compared to the increase of the factor U. It should be clarified that the DACP algorithm is a decision-making tool enabling the users to determine their optimal data offloading satisfying their personal constraints (C_i), as presented in Eq. 27, before they actually perform it.

Furthermore, in Fig. 8 our framework's performance in terms of the MEC servers' probability of failure, users' experienced satisfaction utility and expected overhead is studied. The results reveal that by keeping the number of users constant and increasing the number of MEC servers, the performance of the system improves, since the average amount of data that each MEC server receives from the users is reduced (Fig. 8a), as the users have more choices/MEC servers to offload their data. Thus, the average probability of the MEC servers decreases (Fig. 8b) (p_s is decreasing with respect to $\bar{b}_s$), and the users experience lower expected overhead (Fig. 8d), and greater satisfaction utility (Fig. 8c).

On the other hand, by keeping the number of MEC servers constant and increasing the number of users, the exact opposite phenomenon is observed. Specifically, the MEC servers become more congested as the servers' average received amount of data increases (Fig. 8a), and as a result the average probability of failure of the MEC servers also increases (Fig. 8b). Moreover, since each MEC server is overloaded, the computation capability portion that each user obtains ($F_{i,s}$, Eq. 4) from each MEC server decreases, while at the same time the communication overhead increases. As a result, each user experiences a greater expected overhead (Fig. 8d), thus, its satisfaction utility decreases (Fig. 8c).

In order to further study the effect of competition on the fragility of each MEC server (i.e., treated as CPR) between

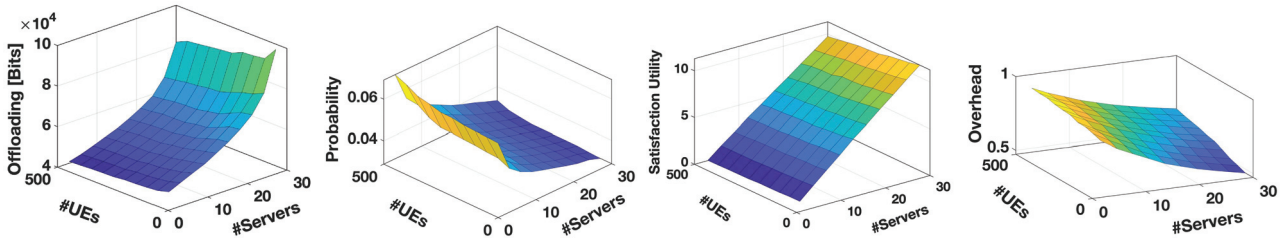


Fig. 8. Performance vs no. of users and MEC servers.

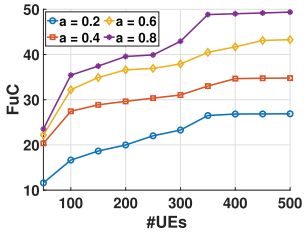


Fig. 9. Fragility under Competition vs no. of users.

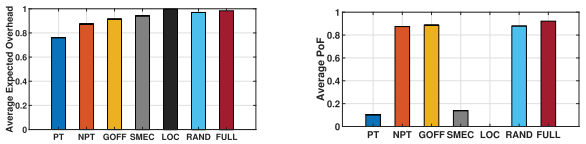


Fig. 10. Comparative evaluation.

a single and several self-interested users, we use the Fragility under Competition (FuC) metric, which is defined as the ratio of the fragility of a MEC server when there are several users to the fragility of the server when there is only one user [31]. The fragility of the MEC server s is expressed by the failure probability function, $p_s(\bar{b}_s)$ which steadily increases as users' total offloaded bits (i.e., investment) $\bar{b}_s$ increases. The fragility of MEC server s is expressed by the failure probability function, which steadily increases as users' total investment increases. Specifically, the Fragility under Competition for each MEC server s is given by $FuC_s = \frac{p_s(\bar{b}_s^*)}{p_s(b_{s,i}^*)}$, where the numerator $p_s(\bar{b}_s^*)$ is the probability of failure function when the total investment in the server s at the Pure Nash Equilibrium (PNE) point of N, $N \geq 2$ homogeneous visitors is $\bar{b}_s^*$, whereas the denominator $p_s(b_{i,s}^*)$ is the probability of failure function when considering a single user i (i.e., $N = 1$) who has the same risk preferences as the homogeneous group and its optimal investment in CPR is $b_{i,s}^*$.

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

D. Comparative Analysis

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

(a) non prospect-theoretic (NPT) users, but expected overhead minimizers instead. Each user i determines its best response $\mathbf{b_i^*}$ that minimizes its perceived expected overhead (Eq. 14),

(b) a full game-theoretic offloading approach (only non partial offloading is permitted) (GOFF) [4], where a non-cooperative game is formulated among the users. Each user i determines its best choice $ch_i^* \in \{0, 1, \ldots, S\}$, in terms of which MEC server to select to offload its whole application (data), ($ch_i = 0$, if the user i keeps its application for local execution), that minimizes its perceived expected overhead (Eq. 14),

(c) a single MEC server environment (SMEC), where a single only MEC server with the average capabilities of the three MEC servers of the basic setting is placed, instead of a *multi-MEC servers* environment,

(d) only local (LOC) computing users, who are risk-averse and keep the task's execution locally to obtain the guaranteed, though limited, performance provided by their device,

(e) full offloading users (FULL), who are risk seeking, and offload their whole task to the *multi-MEC servers* environment, by choosing randomly a MEC server s, thus $\mathbf{b_i^*} = B_i$,

(f) each user i determines its best response $\mathbf{b_i^*} = [b_{i,1}^*, \cdots, b_{i,S}^*]$ randomly (RAND), such that $\sum_{s \in \mathbb{S}} b_{i,s} \leq B_i$.

Specifically, Fig. 10a illustrates the users' average experienced overhead and Fig. 10b indicates the MEC servers' average probability of failure for each different approach. The results clearly reveal that our proposed approach achieves the best performance in terms of both experienced overhead and probability of failure, while the LOC, RAND and FULL alternatives achieve the worst performance. In particular, in the LOC approach the users perceive the highest expected overhead, since they keep the whole application for local execution, and thus they obtain the worst performance in terms of time and energy overhead due to the limited local computing characteristics of the devices. On the other hand, under the RAND and FULL approaches which offload either part (RAND) or the whole application (FULL), respectively,

the users experience lower overhead. However, under the FULL approach the MEC servers become overloaded, and thus the highest average probability of failure is observed.

Furthermore, as Fig. 10a presents, the NPT approach achieves the second best performance after our approach. This is due to the pure benefits stemming from the optimization of the partial offloading, while under the GOFF approach the users offload their whole application without taking advantage of the potential for partial offloading. This leads the MEC servers to higher levels of congestion, with a higher probability of failure (Fig. 10b), and as a result since the uncertainty of the MEC servers' successful operation increases, it is expected that the users will execute greater amounts of data locally, and the expected overhead (Eq. 14) increases accordingly.

On the other hand, under the NPT approach, the users make their offloading decisions in order to simply minimize their perceived expected overhead, without evaluating however their perceived overhead regarding the guaranteed performance that they would obtain if executed the offloading amount of data locally. Consequently, the MEC servers conclude to significantly higher probability of failure (Fig. 10b), while the users obtain worst performance compared to our prospect-theoretic approach, where the users' decisions offloading strategies are based on the tradeoff between the perceived performance and the one that they would experience from their local device (Fig. 10a). Finally, the SMEC prospect-theoretic strategy results in relatively good performance in terms of MEC servers' average probability of failure owing to the consideration of the risk-based behavior modeling, however, since there is only one single MEC server, the users enjoy limited computation capabilities, while the communication overhead increases and the MEC server's computation capability is shared among all the users. As a result the SMEC strategy results in relatively higher expected overhead compared to the NPT and GOFF.

VII. Conclusion

In this paper, a novel approach towards determining the optimal data offloading of each user within a multiple MEC servers environment is introduced, while considering users' risk-seeking or loss-aversion behavior due to the computation and communication uncertainty imposed by the multi-MEC system. The users are able to offload part of their computing tasks to the MEC servers and execute the rest locally. Each MEC server is considered as a Common Pool of Resources, serving the users' computing requests, and can potentially fail due to over-exploitation. The latter characteristic is captured by the theory of Tragedy of Commons. In this uncertain environment, the users demonstrate different risk-seeking data offloading behaviors, which are captured in a holistic prospect-theoretic utility function, following the principles of Prospect Theory. The goal of each user is to maximize its perceived satisfaction, as expressed by the prospect-theoretic utility function, by offloading its computing tasks to the MEC servers.

A non-cooperative game among the users is formulated and the corresponding Pure Nash Equilibrium [32]–[34], i.e., optimal data offloading, is determined, while a distributed low-complexity algorithm that converges to the PNE is also introduced. Detailed numerical results were presented highlighting the operation and superiority of the proposed framework, while at the same time providing useful insights about the users' data offloading decisions under realistic conditions and behaviors, within such a competitive multi-MEC environment.

Our current and future research work focuses on treating the overall key problem of data offloading in various cloud computing environments, such as fog computing, where a large number of computing devices imposes scalability and stability challenges. Finally, it should be noted that the application of Prospect Theory can be adopted in several diverse fields that involve decision-making under uncertainty, such as finance, crowd-sourcing, 5G systems, etc. With reference to the latter, spectrum fragility and resource pricing have been recently jointly investigated [21] under a common resource management umbrella, by utilizing the principles of Prospect Theory, as a means of preserving resource stability. It is indeed of high research and practical significance to investigate how pricing mechanisms can influence the prospect-theoretic users' behavior within a multi-access edge computing environment in terms of offloading their data to the CPR, and in particular treating the potential issue of CPR's fragility associated with the over-exploitation of this resource under uncertainty.

References

[1] C. V. Networking, "Cisco global cloud index: Forecast and methodology, 2015–2020," Cisco, San Jose, CA, USA, White Paper, 2017.

[2] N. Heuveldop *et al.*, "Ericsson mobility report," Ericsson, Stockholm, Sweden, Tech. Rep., 2017.

[3] Y. Mao, J. Zhang, S. H. Song, and K. B. Letaief, "Power-delay tradeoff in multi-user mobile-edge computing systems," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2016, pp. 1–6.

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

[5] K. Zhang *et al.*, "Energy-efficient offloading for mobile edge computing in 5G heterogeneous networks," *IEEE Access*, vol. 4, pp. 5896–5907, 2016.

[6] H. Yu, Q. Wang, and S. Guo, "Energy-efficient task offloading and resource scheduling for mobile edge computing," in *Proc. IEEE Int. Conf. Netw., Archit. Storage (NAS)*, Oct. 2018, pp. 1–4.

[7] T.-Y. Kan, Y. Chiang, and H.-Y. Wei, "Task offloading and resource allocation in mobile-edge computing system," in *Proc. 27th Wireless Opt. Commun. Conf. (WOCC)*, Apr. 2018, pp. 1–4.

[8] S. Sardellitti, G. Scutari, and S. Barbarossa, "Joint optimization of radio and computational resources for multicell mobile-edge computing," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 1, no. 2, pp. 89–103, Jun. 2015.

[9] Q. Fan and N. Ansari, "Towards workload balancing in fog computing empowered IoT," *IEEE Trans. Netw. Sci. Eng.*, vol. 7, no. 1, pp. 253–262, Jan. 2020.

[10] A. Kiani, N. Ansari, and A. Khreishah, "Hierarchical capacity provisioning for fog computing," *IEEE/ACM Trans. Netw.*, vol. 27, no. 3, pp. 962–971, Jun. 2019.

[11] J. Yao and N. Ansari, "QoS-aware fog resource provisioning and mobile device power control in IoT networks," *IEEE Trans. Netw. Service Manage.*, vol. 16, no. 1, pp. 167–175, Mar. 2019.

[12] T. Zhang, "Data offloading in mobile edge computing: A coalition and pricing based approach," *IEEE Access*, vol. 6, pp. 2760–2767, 2018.

[13] Y. Zhang, J. He, and S. Guo, "Energy-efficient dynamic task offloading for energy harvesting mobile cloud computing," in *Proc. IEEE Int. Conf. Netw., Archit. Storage (NAS)*, Oct. 2018, pp. 1–4.

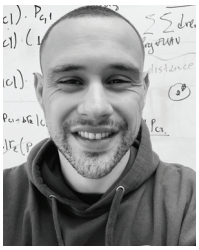
[14] Y. Mao, J. Zhang, and K. B. Letaief, "Dynamic computation offloading for mobile-edge computing with energy harvesting devices," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 12, pp. 3590–3605, Dec. 2016.

[15] K. Guo, M. Yang, Y. Zhang, and Y. Ji, "An efficient dynamic offloading approach based on optimization technique for mobile edge computing," in *Proc. 6th IEEE Int. Conf. Mobile Cloud Comput., Services, Eng. (MobileCloud)*, Mar. 2018, pp. 29–36.

[16] I. Ketyko, L. Kecskes, C. Nemes, and L. Farkas, "Multi-user computation offloading as multiple knapsack problem for 5G mobile edge computing," in *Proc. Eur. Conf. Netw. Commun. (EuCNC)*, Jun. 2016, pp. 225–229.

[17] Y. Wang, M. Sheng, X. Wang, L. Wang, and J. Li, "Mobile-edge computing: Partial computation offloading using dynamic voltage scaling," *IEEE Trans. Commun.*, vol. 64, no. 10, pp. 4268–4282, Oct. 2016.

[18] W. Chen, D. Wang, and K. Li, "Multi-user multi-task computation offloading in green mobile edge cloud computing," *IEEE Trans. Services Comput.*, vol. 12, no. 5, pp. 726–738, Sep. 2019.
[19] D. Kahneman and A. Tversky, "Prospect theory: An analysis of decision under risk," in *Handbook of the Fundamentals of Financial Decision Making: Part I*. Singapore: World Scientific, 2013, pp. 99–127.
[20] P. Vamvakas, E. E. Tsiropoulou, and S. Papavassiliou, "Dynamic spectrum management in 5G wireless networks: A real-life modeling approach," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, Apr. 2019, pp. 2134–2142.
[21] P. Vamvakas, E. E. Tsiropoulou, and S. Papavassiliou, "On controlling spectrum fragility via resource pricing in 5G wireless networks," *IEEE Netw. Lett.*, vol. 1, no. 3, pp. 111–115, Sep. 2019.
[22] G. Hardin, "Extensions of the tragedy of the commons," *Science*, vol. 280, no. 5364, pp. 682–683, 1998.
[23] C. U. Saraydar, N. B. Mandayam, and D. J. Goodman, "Efficient power control via pricing in wireless data networks," *IEEE Trans. Commun.*, vol. 50, no. 2, pp. 291–303, Feb. 2002.
[24] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.
[25] J. B. Rosen, "Existence and uniqueness of equilibrium points for concave N-person games," *Econometrica*, vol. 33, no. 3, pp. 520–534, Jul. 1965.
[26] C. Kao, "Performance of several nonlinear programming software packages on microcomputers," *Comput. Oper. Res.*, vol. 25, no. 10, pp. 807–816, Oct. 1998.
[27] P. T. Boggs and J. W. Tolle, "Sequential quadratic programming," *Acta Numer.*, vol. 4, pp. 1–51, Jan. 1995.
[28] A. Grace, *Optim. Toolbox: For Use With MATLAB: User's Guide*. Natick, MA, USA: MathWorks, Nov. 1990.
[29] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*. Cambridge, MA, USA: MIT Press, 2009.
[30] A. P. Miettinen and J. K. Nurminen, "Energy efficiency of mobile clients in cloud computing," *HotCloud*, vol. 10, p. 19, Jun. 2010.
[31] A. R. Hota, S. Garg, and S. Sundaram, "Fragility of the commons under prospect-theoretic risk attitudes," *Games Econ. Behav.*, vol. 98, pp. 135–164, Jul. 2016.
[32] S. B. Russ, "A translation of Bolzano's paper on the intermediate value theorem," *Historia Math.*, vol. 7, no. 2, pp. 156–185, May 1980.
[33] K. Sydsaeter, *Mathematics for Economics Analysis*. London, U.K.: Pearson, 2013.
[34] F. R. Gantmakher, *The Theory of Matrices*, vol. 131. New Providence, NJ, USA: American Mathematical Society, 2000.



Pavlos Athanasios Apostolopoulos (Student Member, IEEE) received the Diploma degree in ECE from the National Technical University of Athens in 2017. He is currently pursuing the Ph.D. degree with the Department of Electrical and Computer Engineering (ECE), University of New Mexico. He is also a Research Assistant with the Department of ECE, University of New Mexico. His main research interests include machine learning, deep learning, game theory, behavioral decision theory, and system optimization.

Eirini Eleni Tsiropoulou (Member, IEEE) is currently an Assistant Professor with the Department of Electrical and Computer Engineering, The University of New Mexico. Her main research interests lie in the area of cyber-physical social systems and wireless heterogeneous networks, with an emphasis on network modeling and optimization, resource orchestration in interdependent systems, reinforcement learning, game theory, network economics, and the Internet of Things. 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.

Symeon Papavassiliou (Senior Member, IEEE) was a Senior Technical Staff Member with AT&T Laboratories, NJ, USA, from 1995 to 1999. In August 1999, he joined the ECE Department, New Jersey Institute of Technology, USA, where he was an Associate Professor until 2004. He is currently a Professor with the School of ECE, National Technical University of Athens. He has an established record of publications in his field of expertise, with more than 300 technical journal articles and conference published papers. His main research interests lie in the area of computer communication networks, with an emphasis on the analysis, optimization, and performance evaluation of mobile and distributed systems, wireless networks, and complex systems.