Price and Risk Awareness for Data Offloading Decision-Making in Edge Computing Systems

Giorgos Mitsis, Eirini Eleni Tsiropoulou, *Senior Member, IEEE*, and Symeon Papavassiliou, *Senior Member, IEEE*

***Abstract*—The proliferation of multiaccess edge computing (MEC) paradigm has created a challenging multiuser–multiserver–multiaccess edge computing competitive environment, which brings the problem of data offloading decision-making to the forefront of research. In this article, we address this issue while jointly studying the impact of the user behavioral characteristics and the MEC servers pricing policies on determining the optimal user data offloading strategies. Prospect theory is exploited to reflect the user satisfaction and subjectivity from the data offloading, while the MEC servers' probability of failure owing to the potential overexploitation by the users, is modeled via the theory of tragedy of the commons. A multileader multifollower Stackelberg game is formulated among the MEC servers (leaders) and the users (followers), to determine the servers' optimal pricing policies and the users' optimal data offloading strategies. The users' data offloading decision-making is formulated as a noncooperative game among them and a Nash equilibrium is determined, while the MEC servers' optimal computing service prices are obtained either through a semiautonomous game-theoretic approach, or through a fully autonomous reinforcement learning-based approach. The performance evaluation and demonstration of the superiority of the proposed framework against other benchmarking alternatives is achieved via modeling and simulation.**

***Index Terms*—Edge computing, game theory, network economics, prospect theory, reinforcement learning (RL).**

I. Introduction

THE proliferation of mobile devices, such as smart phones, wearable devices, and Internet of Things (IoT) sensor nodes, renders the explosive growth of data and induces the emergence of computation-intensive and latency-critical applications, such as virtual and augmented reality, online gaming, surveillance, and others. Toward supporting those resource-hungry applications, multiaccess edge computing (MEC) is envisioned as a promising computing paradigm [1]. In such a setting, the users are able to offload their computation-intensive tasks to resource-rich infrastructures, i.e., MEC servers, which are usually colocated with macro base station (MBS) or unmanned aerial vehicles (UAVs). Thus, the development of optimal data offloading policies have recently received significant attention from industry and academia [2].

In parallel, the behavioral and economic modeling of the users' data offloading schemes, while accounting for the users behavioral decision-making characteristics and the MEC servers' computing service pricing policy, though of high practical importance, is still at an infant stage of study due to the inherent complexity and multidimensional nature of the problem [3]. In this article, we aim at exactly addressing this issue, by jointly studying the interplay of the users behavioral characteristics and the MEC servers pricing policies, as well as their impact on determining the optimal users data offloading strategies. The key objective is to simultaneously maximize the users' perceived service satisfaction and the MEC servers' profit. The introduced novel behavioral and economic modeling is performed based on the principles of prospect theory and network economics, while the users' and MEC servers' distributed decision-making is facilitated by game-theoretic (GT) and reinforcement learning (RL)-based approaches.

A. Related Work

Significant research efforts have been lately devoted to the investigation of the problem of multiuser and multiserver data offloading in MEC environments, under various settings. In [4], the authors introduced a multivariable centralized minimization problem of the users' energy cost and experienced latency by jointly determining the optimal users' data offloading strategies, users' scheduling, and resource allocation. In [5], the authors focused their study on small cell networks, where each small cell's access point is equipped with an MEC server. In particular, the authors determined the users' optimal data offloading strategies in a distributed manner via a GT approach based on the theory of potential games, while also addressing the minimization problem of the users' energy consumption and service delay. The data offloading problem in vehicular networks is studied in [6], where the MEC servers reside at the road side units. A combination of convex optimization and a GT approach is introduced to optimize the system wide profit of both the vehicles and the network operator via determining the optimal communication channel allocation, data offloading, and task scheduling at the MEC servers. A similar approach is introduced in [7] enabling the patients' medical nodes to offload data to MEC servers.

Manuscript received 24 July 2021; revised 10 January 2022 and 14 June 2022; accepted 3 July 2022. Date of publication 19 July 2022; date of current version 9 December 2022. This work was supported in part by the Hellenic Foundation for Research and Innovation (H.F.R.I.) under the "1st Call for H.F.R.I. Research Projects to support Faculty members and Researchers and the procurement of high-cost research equipment grant" (Project Number: HFRI-FM17-2436). The work of Eirini Eleni Tsiropoulou was supported by the NSF under Grant CRII-1849739. *(Corresponding author: Symeon Papavassiliou.)*

Giorgos Mitsis and Symeon Papavassiliou are with the School of Electrical and Computer Engineering, National Technical University of Athens, 15780 Athens, Greece (e-mail: gmitsis@netmode.ntua.gr; papavass@mail.ntua.gr).

Eirini Eleni Tsiropoulou is with the Department of Electrical and Computer Engineering, University of New Mexico, Albuquerque, NM 87131 USA (e-mail: eirini@unm.edu).

This article has supplementary downloadable material available at https://doi.org/10.1109/JSYST.2022.3188997, provided by the authors.

Digital Object Identifier 10.1109/JSYST.2022.3188997

Apart from the GT approaches, RL-based techniques have also been devised in literature to address the data offloading problem [8]. In [9], a budged-limited multiarmed bandit problem is formulated in order to enable the users to select the MEC server that minimizes their latency and energy consumption, as well as the corresponding amount of offloaded data. A similar problem formulation is introduced in [10] with application on vehicular networks. Specifically, the authors consider the vehicles' mobility, the MEC servers' heterogeneous computation resources, and the vehicles diverse computation demand in the designed multiarmed bandit learning algorithm. Moreover, an ϵ-greedy nonstationary multiarmed bandit-based scheme for online data offloading is introduced in [11] targeting at the minimization of the users' energy consumption and latency, and the MEC servers' computation resource usage optimization. Also, a fog-enabled federated learning framework is introduced in [12] to enable the distributed learning for supporting delay-sensitive applications in resource-constrained IoT systems. Moreover, Stackelberg games have been widely used to model the economic-based interaction between two entities in a hierarchical architecture [13], [14]. In [15], a Stackelberg game is formulated among the MEC server and the wireless body area network users in order to derive a joint cost and energy efficient task offloading mechanism. A novel layered optimization approach is proposed in [16] to minimize the users' overall delay by jointly optimizing the users' offloaded tasks and their transmission time.

On the other hand, rather limited research effort has been devoted to the problem of optimal computing service pricing from the MEC servers' side. In [17], several types of pricing policies, such as multidimensional pricing, penalty pricing, and discount pricing, have been proposed to study the different number of virtual machines that a cloudlet can accommodate. Aiming at minimizing the users' cost, while jointly maximizing the edge cloud's profit, a two-side game is introduced in [18] and [19] to determine the optimal MEC servers' price and the users' data offloading strategies. In [20], a static pricing-based approach is proposed to guide the users' cooperation with the MEC servers to conclude to a stable operational point. A dynamic pricing mechanism is devised in [21] to minimize the overall MEC system's cost, while guaranteeing the satisfaction of the users' quality of service (QoS).

It should be noted that all the aforementioned research works consider the users as rational decision-makers aiming at maximizing their perceived utility, while interacting with the MEC servers. However, in a realistic edge computing environment, the users typically demonstrate a risk-aware decision-making behavior, where the risk primarily stems from the scarcity due to the potential overexploitation of the computation resources available to the MEC servers. Prospect theory has been traditionally used in literature to capture the users' risk-aware behavior as compared to the expected utility theory [22]. Toward capturing the users' risk-aware decision-making, prospect theory has also been recently adopted in MEC environments [23]. However, these research attempts have been realized under the assumption that all the users weigh the MEC servers' probability of failure to serve their computing requests in exactly the same manner. This problem has been studied in ground-based MEC systems [3] or UAV-assisted MEC systems [24], while accounting for different imposed communication constraints or static pricing models [25]. Nevertheless, in these research works the joint consideration of the users to MEC servers optimal association and the MEC servers' optimal price decision regarding their offered computing services to the users is not treated. The latter problem, along with the adoption of a weighted probability that captures the distorted perception of the probability spectrum proposed by the prospect theory within the MEC environment, is part of the novelty of our work.

Prospect theory has been also combined with the theory of the tragedy of the commons [26] to capture the failure of the common pool resources (CPR), e.g., MEC servers, to serve the users due to their overexploitation [27]. In general, the principles of prospect theory and the tragedy of the commons have already been applied in several other research fields, such as dynamic spectrum management [28], [29], load balancing in smart grid systems [30], antijamming communications in cognitive systems [31], fog computing security [32], and network security [33]. In [34], prospect theory is combined with blockchain to determine the users' optimal data offloading toward jointly maximizing the utilities of both the miner devices and the MEC server providers. Also, in [24], a UAV-assisted MEC system is examined, and the users' optimal data offloading is determined based on a GT approach, while accounting for the risk of the MEC servers' failure due to overexploitation.

B. Contributions and Outline

In this article, we introduce a novel dynamic behavior and price-aware edge computing model to determine the users' optimal data offloading strategies and the MEC servers' optimal computing service pricing. One key novelty of our introduced model and approach is that the aforementioned objective is achieved while accounting for the users' risk-aware decision-making due to the potential MEC servers overexploitation. Even though prospect theory has been recently used to capture the usage risk-aware behavior in MEC systems, very little effort has been devoted to the problem of quantifying the sources and levels of risks in users' decision-making. In this article, toward achieving the latter goal, we jointly examine the principles of tragedy of the commons along with the users' probability weighting phenomenon to provide a more realistic and holistic approach regarding the users' risk-aware decision-making process. The key contributions and novelties that differentiate our article from the rest of the literature, are summarized as follows.

1) A multiuser–multiserver–multiaccess edge computing environment is considered. The MEC servers' probability of failure due to overexploitation by the users is captured via the theory of the tragedy of the commons. In contrast to the existing literature, in this work, we account for the probability weighting phenomenon, where the users tend to overestimate the likelihood of events with low probability of failure and underweight outcomes with high probability of failure.
2) To account for the users' risk-aware decision-making and behavior, their satisfaction from the data offloading and processing is captured by appropriately designed prospect-theoretic utility functions. Moreover the MEC servers' profit by serving the users' computation demand is designed as a function of the MEC servers' computing service pricing.
3) The association problem between users and MEC servers is jointly treated with the data offloading problem, which comes in contrast to existing alternative approaches in literature, where the server selection and data offloading processes are performed in a disjoint and uncorrelated manner. To achieve this, a multileader–multifollower

Stackelberg game is formulated among the MEC servers (leaders) and the users (followers) to determine the MEC servers' optimal computing service pricing policies and the users' optimal data offloading strategies. The goal of the users is to maximize their expected prospect-theoretic utility, while accounting for the servers' pricing policies and their probability to fail while serving the users due to potential overexploitation of their computing resources. The users' data offloading decision-making is formulated as a noncooperative game among them and a Nash equilibrium (NE) point is determined.

4) The MEC servers' optimal computing service pricing policies are obtained following two alternative decision-making mechanisms, that present different benefits and tradeoffs. The first one introduces a semiautonomous GT approach, while the second one provides for a fully autonomous RL-based approach in order to tackle the common problem of utility-specificity in GT approaches.
5) A detailed numerical analysis and evaluation is realized via modeling and simulation, to quantify the performance of the proposed edge computing framework under both decision-making alternatives and models (i.e., GT and RL-based ones), in terms of convergence and/or operation efficiency. Furthermore, a comparative evaluation of the proposed framework against other alternative data offloading benchmarking strategies is presented and discussed.

The rest of the article is organized as follows. Section II presents the overall system model and an overview of the operation of the proposed framework. In Section III, the users' data offloading problem is formulated and solved based on a GT approach. In Section IV, the MEC servers' optimal computing service pricing policies are determined based on the GT and RL-based alternatives. In Section V, simulation and comparative numerical results are illustrated and analyzed. Finally, Section VI concludes this article.

II. Multiaccess Edge Computing

A. Behavior and Price-Aware Modeling

We consider a multiuser–multiserver–multiaccess edge computing environment, consisting of a set of users $N = \{1, \ldots, n, \ldots, |N|\}$ and a set of MEC servers $S = \{1, \ldots, s, \ldots, |S|\}$. Each user requests a service that is characterized by a computation task $J_n = (b_n, i_n)$, where b_n [bits] denotes the input bits that need to be processed and i_n [CPU cycles] the computation demand of the user's service, expressing the number of necessary CPU cycles to process the b_n bits. Each user can select one server to offload $b_{n,s}^{\text{MEC}}$ [bits] amount of data, while the rest of the data, i.e., $b_n - b_{n,s}^{\text{MEC}}$, are processed locally on the user's device. The user's device computation capability is denoted as f_n [CPU cycles/sec] and the consumed energy per CPU cycle to locally process the user's data is γ_n [J/CPU cycles]. The total processing time for each user's computation task, if it is fully processed locally, is $t_n = \frac{i_n}{f_n}$ [s] and the corresponding consumed energy is $e_n = \gamma_n i_n$ [J]. Each MEC server charges p_s [\$/bit] monetary units per bit of processed data to perform the computing.

The computing capabilities of the MEC servers are assumed to be shared among the users, thus, they are treated as a CPR. Given that the CPR is excludable, rivalrous, and can be commonly accessible to all users, the phenomenon of the tragedy of the commons may arise [26]. Thus, the MEC servers may fail to serve the users due to potential overexploitation, and no user will enjoy the computing capabilities of the server that failed. The users may experience risks in their decision-making process, i.e., to which server to offload part of their data, which may stem from either the complete failure or the depletion of the computing resources, caused by the potential (over)exploitation of the CPR, i.e., fragility of the shared resources. In our proposed framework, each user reacts in a personalized risk-aware manner based on its perception of the MEC servers' computing resources' usage. The majority of the existing literature applies centralized admission control mechanisms to allow the users to access the MEC servers' computing resources. However, it is well known that a centralized admission control approach suffers from several drawbacks, e.g., single point of failure, control and communication overhead, and privacy concerns. Moreover, it is highlighted that in emerging complex MEC systems, due to the fact that different MEC servers may be owned by different service providers, the solution of a centralized entity performing admission control and task scheduling would not be realistic, or even feasible is several cases.

Based on the general principles of prospect theory, the users present different behavior (i.e., utility values), expressed as satisfaction or dissatisfaction, based on the gains or losses they experience from a service. Specifically, based on the *loss aversion property*, the users experience greater dissatisfaction in the case of losses compared to the perceived satisfaction from gains of the same magnitude. The aforementioned gains and losses are determined with respect to a predefined reference point $U_{n,0}$, which in our case is defined as $U_{n,0} = \frac{b_n}{t_n e_n}$, reflecting the user's satisfaction from processing its computation tasks on its device. The latter captures the user's perceived utility if it processed the whole amount of its data locally.

Therefore, the user's prospect-theoretic utility by offloading $b_{n,s}^{\text{MEC}}$ data to an MEC server is defined formally as follows:

$$P_{n,s}(U_{n,s}) = \begin{cases} (U_{n,s} - U_{n,0})^{\alpha_n}, & \text{if } U_{n,s} \geq U_{n,0} \\ -k_n(U_{n,0} - U_{n,s})^{\beta_n}, & \text{otherwise} \end{cases} \quad (1)$$

where $\alpha_n, \beta_n \in [0, 1]$, and $k_n \in R^+$. The risk-aware parameters α_n, β_n reflect the users' risk-averse behavior in gains, and risk-seeking behavior in losses, respectively. Also, the loss aversion parameter k_n captures the way that the user weighs the losses and gains. Specifically, the user weighs the gains more than ($k_n < 1$) or equal to ($k_n = 1$) the losses, while the opposite holds true if $k_n > 1$. In the following analysis, without loss of generality, we consider that the users' risk-aware parameters are equal, i.e., $\alpha_n = \beta_n \forall n \in N$. The user's actual utility function $U_{n,s}(b_{n,s}^{\text{MEC}})$ captures the user's actual satisfaction from: either 1) processing all its data locally on its device [first branch of (2)], or 2) offloading part of its data to an MEC server, while the latter one survives [second branch of (2)], or 3) offloading part of its data to an MEC server, while the latter one fails [third branch of (2)]. The user's actual utility function is defined as follows:

$$U_{n,s}(\mathbf{b}_s^{\text{MEC}}) =$$

$$\begin{cases} \dfrac{b_n}{t_n e_n}, & \text{if } b_{n,s}^{\text{MEC}} = 0 \\ \frac{b_n - b_{n,s}^{\text{MEC}}}{t_n e_n} + b_{n,s}^{\text{MEC}} R(D_s) - c_s(b_{n,s}^{\text{MEC}}), & \text{if } b_{n,s}^{\text{MEC}} \neq 0 \ \& \\ & s \text{ survives} \\ \dfrac{b_n - b_{n,s}^{\text{MEC}}}{t_n e_n} - c_s(b_{n,s}^{\text{MEC}}), & \text{if } b_{n,s}^{\text{MEC}} \neq 0 \ \& \ s \text{ fails} \end{cases} \quad (2)$$

where $\mathbf{b}_s^{\text{MEC}}$ denotes the data offloading vector of all the users, and $c_s(b_{n,s}^{\text{MEC}})$ denotes the user's cost by processing its data to the MEC server s. The latter is obtained based on the announced price p_s [\$] by the MEC server s and the corresponding normalized amount of its offloaded data. Therefore, the user's cost can be formally defined as follows:

$$c_s\left(b_{n,s}^{\text{MEC}}\right) = p_s i_n \frac{b_{n,s}^{\text{MEC}}}{b_n}. \tag{3}$$

The physical meaning of (3) is that, as expected, a user experiences a higher cost from the MEC server either due to a high computing service price or if it requests a large amount of data to be processed following the principles of proportional fairness or if the data are characterized by high computation demand to be processed at the MEC server. For fairness purposes among the users, we consider that the MEC server announces the same price p_s for all the users that offload their computation tasks to it for further processing. Nevertheless different prices are announced by the different servers to the users in order to promote competition. Furthermore, the second branch of (2) is formulated based on the satisfaction that a user experiences from offloading part of its data to the MEC server (first term), while considering the cost that is charged with to process its data at the server (third term) and the rate of return $R(D_s)$ that it experiences by having its data $b_{n,s}^{\text{MEC}}$ processed at the edge (second term). The rate of return implicitly reflects the value that the user gains from to the remote execution of its task. In particular, the rate of return function $R(D_s)$ is assumed to be continuous and monotonically decreasing with respect to the users' normalized effective demand D_s from the server (formally defined below). Thus, if the users' normalized effective demand D_s is high, meaning that the MEC server's computing capabilities are overexploited, the satisfaction that the users experience by processing their data to the server is decreased due to an increased data processing delay. For demonstration purposes, in the following analysis, the MEC servers' rate of return function is formulated as follows:

$$R(D_s) = 2 - e^{D_s - 1}. \tag{4}$$

The users' normalized effective demand D_s from the MEC server s is a sigmoidal function that maps the users' actual computing demand $d_s = \sum_{n=1}^{|N|} i_n \frac{b_{n,s}^{\text{MEC}}}{b_n}$ from the MEC server s to the interval [0,1] and is a continuous and strictly increasing function with respect to d_s, defined as follows:

$$D_s(d_s) = -1 + \frac{2}{1 + e^{-\theta_s d_s}}. \tag{5}$$

The parameter $\theta_s > 0$ is a positive constant which is used to calibrate the sigmoidal curve to appropriately capture the MEC servers' computing capabilities. In a practical implementation, the value of the normalized effective demand D_s is broadcasted by the server to the users, providing an indication of how overexploited an MEC server is, in order to further facilitate the users' distributed decision-making process. Given the CPR nature of the MEC server's computing capability, due to the joint exploitation from multiple users that offload their data to the same server, the latter one is characterized by a probability of failure $Pr_s(D_s)$ depending on the users' normalized effective demand D_s. The MEC server's probability of failure is a continuous and strictly increasing function with respect to the users' demand D_s and can be indicatively defined as $Pr_s(D_s) = D_s^2$. It is noted that the latter function is adopted only for demonstration purposes, while any continuous and strictly increasing probability of failure function can be adopted without limiting the applicability of the rest of the analysis.

Based on the previous analysis and discussion, and for simplicity in the presentation, let us denote as $U_{n,s}^{\text{surv.}}$ and $U_{n,s}^{\text{fail}}$ the second and third branch of (2), respectively. Then the user's prospect-theoretic utility function, as expressed in (1), can be rewritten as follows:

$$P_{n,s}(b_{n,s}^{\text{MEC}}, \mathbf{b}_{-n,s}^{\text{MEC}}) = \begin{cases} P_{n,s}^{\text{surv.}} = \left(U_{n,s}^{\text{surv.}} - \frac{b_n}{t_n e_n}\right)^{\alpha_n}, & \text{if } U_{n,s}^{\text{surv.}} \geq U_{n,0} \\ P_{n,s}^{\text{fail}} = -k_n \left(\frac{b_n}{t_n e_n} - U_{n,s}^{\text{fail}}\right)^{\alpha_n}, & \text{otherwise} \end{cases} \tag{6}$$

where $\mathbf{b}_{-n,s}^{\text{MEC}}$ denotes the data offloading vector of all the users except for user n to the MEC server s. One of the key principles and findings of prospect theory, states that the users tend to overestimate the likelihood of events with low probability of failure and underweight outcomes with high probability of failure, i.e., $\pi(Pr_s) > Pr_s$ for small Pr_s values and $\pi(Pr_s) < Pr_s$ for large Pr_s values. This latter observation of how humans behave under risk-aware decision-making processes is defined as the *probability weighting phenomenon*. The prospect-theoretic probability weighting function $\pi(Pr_s)$ of outcomes with different likelihood to occur is defined as follows [35]:

$$\pi(Pr_s) = e^{-(-\ln(Pr))^{\gamma}} \tag{7}$$

where $\gamma \in \mathbb{R}^+$ denotes the psychological distortion parameter.

Considering the aforementioned probabilities, the user's expected prospect-theoretic utility function from offloading part of its data to a selected MEC server is defined as follows:

$$\mathbb{E}(P_{n,s}(b_{n,s}^{\text{MEC}}, \mathbf{b}_{-n,s}^{\text{MEC}})) = P_{n,s}^{\text{surv.}}(1 - \pi(Pr_s)) + P_{n,s}^{\text{fail}}\pi(Pr_s). \tag{8}$$

Focusing on the MEC servers' side, each MEC server announces the price p_s for serving the user's computing requests, while bearing an operational cost κ_s [\$] to process the data and support its operation. Each MEC server's reward from participating in the MEC environment is defined as

$$\mathcal{R}(p_s) = B_s(p_s - \kappa_s) \tag{9}$$

where $B_s = \sum_{n=1}^{|N|} b_{n,s}^{\text{MEC}}$ is the total amount of offloaded data to the MEC server s. Focusing on the network economics-based operation of an MEC server, we make the following observations. An MEC server naturally tends to increase its announced price p_s if: 1) its operational cost is high, in order to sustain some profit 2) it processes a large amount of data, reaching its maximum capacity B^{MAX} [bits] in terms of data that can simultaneously process, in order to avoid its overexploitation and even failure in the worst case scenario, and 3) the rest of the MEC servers increase their price $\mathbf{p}_{-s} = [p_1, \ldots, p_{s-1}, p_{s+1}, \ldots, p_{|S|}]$, in order to remain competitive in the edge computing market. Based on these observations and interdependencies, we define the MEC server's payoff function that captures the aforementioned aspects, as follows:

$$W(p_s) = -\left(p_s - \frac{B_s}{B^{\text{MAX}}}\kappa_s \frac{\sum_{\forall j \neq s} p_j}{p_s}\right)^2. \tag{10}$$

B. Edge Computing Operation

In this section, we provide an overview of the operation of the proposed behavior and price-aware multiuser–multiserver–multiaccess edge computing system. We formulate its operation as a multileader–multifollower Stackelberg game, where the

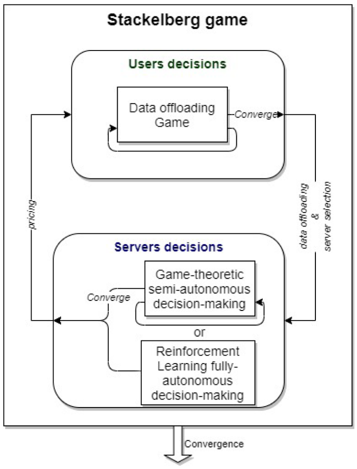

Stackelberg game
Users decisions
Data offloading Game
Converge
pricing
data offloading & server selection
Servers decisions
Game-theoretic semi-autonomous decision-making
Converge
or
Reinforcement Learning fully-autonomous decision-making
Convergence

Fig. 1. Overview of the proposed framework.

users act as followers, determining their optimal amount of offloaded data, and the MEC servers behave as leaders, announcing their optimal price to provide their computing services to the users. An overview of the proposed framework's operation is presented in Fig. 1.

Initially, the MEC servers select the prices to impose to the users (e.g., randomly) without any knowledge on the amount of data that each user is willing to offload. Given the MEC servers' prices, the users participate in a noncooperative game among them, in order to determine the server with whom they want to associate with, as well as the optimal amount of offloaded data. This is done based on the criterion of each user maximizing its perceived expected prospect-theoretic utility function, as defined in (8). The latter outcome in turn acts as input to the MEC servers, who determine the optimal announced prices to offer their computing services to the users. It is noted that the optimal prices of the MEC server are determined with two different alternatives based on the information availability among the MEC servers, as well as the methodological learning philosophy adopted to conclude to the optimal solutions. Specifically, a semiautonomous GT model and a fully autonomous RL-based model are introduced and their drawbacks and benefits are discussed and demonstrated in a comparative manner. The interaction among the users and MEC servers is repeated iteratively until the overall system converges to a Stackeblerg equilibrium, where the users' data offloading strategies and the MEC servers' prices have converged to the optimal values.

III. Optimal Data Offloading

A. Problem Formulation

In this section, the problem of determining the MEC servers' selection by the users and the optimal data offloading strategies is formulated as a distributed optimization problem. Each user aims at selecting the MEC server that will eventually maximize the user's expected prospect-theoretic utility, while in parallel determining the optimal data offloading strategy. Thus, in our proposed framework, the association problem between users and MEC servers is jointly treated with the data offloading problem. The corresponding optimization problem is formulated as follows:

$$\begin{aligned}&\max_{\forall s\in S}\left\{\max_{b_{n,s}^{\mathrm{MEC}}}\mathbb{E}\left(P_{n,s}\left(b_{n,s}^{\mathrm{MEC}},\mathbf{b}_{-n,s}^{\mathrm{MEC}}\right)\right)\right\},\\&\text{s.t. } 0\le b_{n,s}^{\mathrm{MEC}}\le b_n.\end{aligned}\tag{11}$$

Thus, a user selects the MEC server that maximizes the maximum potential expected prospect-theoretic utility. Toward determining the latter value, the nested optimization problem should be addressed as follows:

$$\max_{b_{n,s}^{\mathrm{MEC}}\in[0,b_n]}\mathbb{E}\left(P_{n,s}\left(b_{n,s}^{\mathrm{MEC}},\mathbf{b}_{-n,s}^{\mathrm{MEC}}\right)\right).\tag{12}$$

The optimization problem in (12) can be addressed as a noncooperative game among the users, who compete among each other about the MEC server's computing resources. The noncooperative game is defined as $G=[N,\{\mathcal{B}_n\}_{\forall n\in N},\{\mathbb{E}(P_{n,s})_{\forall n\in N}\}]$, where N is the set of users, $\mathcal{B}_n=[0,b_n]$ is each user's strategy set, and $\mathbb{E}(P_{n,s})$ is the user's expected prospect-theoretic utility function. Our goal is to determine a nNE point, where the users have converged to their optimal data offloading strategies.

Definition 1 (Nash Equilibrium): A data offloading vector $\mathbf{b}^*=(b_{1,s}^{\mathrm{MEC}*},\ldots,b_{n,s}^{\mathrm{MEC}*},\ldots,b_{|N|,s}^{\mathrm{MEC}*})\forall s\in S$ is an NE if the following condition holds true for every user $n\in N\forall s\in S,\forall b_{n,s}^{\mathrm{MEC}}\in\mathcal{B}_n$:

$$\mathbb{E}\left(P_{n,s}\left(b_{n,s}^{\mathrm{MEC}*},\mathbf{b}_{-n,s}^{\mathrm{MEC}*}\right)\right)\ge\mathbb{E}\left(P_{n,s}\left(b_{n,s}^{\mathrm{MEC}},\mathbf{b}_{-n,s}^{\mathrm{MEC}*}\right)\right).\tag{13}$$

B. Problem Solution

Toward determining the existence of an NE of the noncooperative game G, we show that the game is submodular.

Definition 2 (Submodular Games): The noncooperative game $G=[N,\{\mathcal{B}_n\}_{\forall n\in N},\{\mathbb{E}(P_{n,s})_{\forall n\in N}\}]$ is submodular if the following conditions hold true for all users.

1) $\mathcal{B}_n\forall n\in N$ is a compact subset of an Euclidean space.
2) $\mathbb{E}(P_{n,s})\forall n\in N\forall s\in S$ is smooth, submodular in $b_{n,s}^{\mathrm{MEC}}$ and has nonincreasing differences in $(b_{n,s}^{\mathrm{MEC}},\mathbf{b}_{-n,s}^{\mathrm{MEC}})$, i.e., $\frac{\partial^2\mathbb{E}(P_{n,s})}{\partial b_{n,s}^{\mathrm{MEC}}\partial b_{n,'s}^{\mathrm{MEC}}}\le 0$.

Theorem 1: The noncooperative game $G=[N,\{\mathcal{B}_n\}_{\forall n\in N},\{\mathbb{E}(P_{n,s})_{\forall n\in N}\}]$ is submodular and has at least one NE point.

Proof: The proof can be concluded following similar reasoning and steps, as in [25]. For additional theoretical details see [36] and [37]. ■

Based on Theorem 1, the existence of at least one NE point is shown. Thus, each user can determine its optimal amount of offloaded data $b_{n,s}^{\mathrm{MEC}*}$ to an MEC server s and select the MEC server s that maximizes its maximum expected prospect-theoretic utility, as expressed in (11). The NE point can be practically determined by following a best response dynamics algorithm [25].

IV. Computing Service Pricing

In this section, our goal is to determine the optimal announced prices by the MEC servers given the users' optimal data offloading strategies $b_{n,s}^{\mathrm{MEC}*}\forall n\in N,s\in S$. Please recall that these prices are utilized by the process described in Section III to determine the users' optimal data offloading, in an overall iterative manner. As defined in (10), each MEC server aims at maximizing its payoff, and therefore, the optimization problem can be defined accordingly as follows:

$$\max_{\{p_s\}_{\forall s \in S}} W(p_s) = -\left(p_s - \frac{B_s}{B^{\text{MAX}}}\kappa_s \frac{\sum_{\forall j \neq s} p_j}{p_s}\right)^2. \quad (14)$$

The above optimization problem can be treated and solved in principle based on standard convex optimization techniques, given that the payoff function $W(p_s)$ is concave with respect to the price p_s. However, such an approach would not be realistic in a real-life implementation, as a centralized entity should perform the optimization and inform the MEC servers about their optimal prices. Several reasons, however, would render such an approach either infeasible or prohibitive in practice. Indicatively we refer to the fact that MEC servers may be owned by different providers, the centralized entity making the decisions is a single point of failure, while significant signaling overhead would be imposed to the MEC servers to interact with the centralized entity. Thus, the need of devising an autonomous decision-making approach for the MEC servers arises. In the following subsections, we particularly focus on this problem and present two strategies to determine each MEC server's optimal announced price: a semiautonomous GT approach which has the objective of directly treating the problem in (14), and a fully autonomous alternative approach of concluding to the optimal price, based on RL.

A. Game-Theoretic Approach-Semiautonomous Decision-Making

The optimization problem in (14) can be formulated as a noncooperative game $\mathbb{G} = [S, \{P_s\}_{\forall s \in S}, \{W(p_s)\}_{\forall s \in S}]$ among the servers, where $P_s = [p_{\min}, p_{\max}]$ denotes their strategy set and $W(p_s)$ their payoff function. In a realistic computing market, the minimum $p_{\min}$ and maximum $p_{\max}$ prices could be set by the market regulations and homogeneously applied to all the computing service providers. Toward showing the existence and uniqueness of an NE point, and accordingly determining their optimal prices $p_s^* \forall s \in S$, we follow the theory of n-person concave games, where $n = |S|$.

Theorem 2 (Existence and Uniqueness of Nash Equilibrium): The noncooperative game $\mathbb{G} = [S, \{P_s\}_{\forall s \in S}, \{W(p_s)\}_{\forall s \in S}]$ is an n-person concave game and admits a unique NE point, if the following conditions hold true [38].

1) The strategy sets $P_1, \ldots, P_{|S|}$ are nonempty, compact, convex subsets of finite dimensional Euclidean spaces.
2) All payoff functions $W(p_1), \ldots, W(p_{|S|})$ are continuous on $P = P_1 \times \cdots \times P_{|S|}$.
3) Every payoff function is concave with respect to p_s, if all other strategies are held fixed.

Proof: By definition, the strategy sets $P_1, \ldots, P_{|S|}$ are nonempty, compact, and convex, and the payoff function $W(p_s)$ of each server is continuous on p_s. Also, it holds true that $\frac{\partial^2 W(p_s)}{\partial p_s^2} = -2 - 6\frac{\left(\frac{B_s}{B^{MAX}}\sum_{\forall j \neq s} p_j\right)^2}{p_s^4} < 0$, thus, the payoff function of each MEC server is concave with respect to p_s. Therefore, the noncooperative game $\mathbb{G}$ is an n-person concave game and admits a unique NE point

$$p_s^* = \sqrt{\frac{B_s}{B^{\text{MAX}}}\kappa_s \sum\nolimits_{\forall j \neq s} p_j}. \quad (15)$$

■

The NE point in (15) can be determined by implementing a best response dynamics algorithm. Based on (15), it is observed that each MEC server needs to be aware of the summation of the prices of all the rest of the MEC servers existing in the examined edge computing environment. In practice, the overall summation of the MEC servers' prices can be broadcasted by a market regulatory entity, which monitors the proper operation of the computing market, to all the MEC servers/edge computing providers. However, the final decision of the optimal price is performed by each MEC server in a distributed manner. Thus, the proposed GT approach to determine the servers' optimal prices is characterized as semiautonomous. Based on Theorems 1 and 2, the Stackelberg equilibrium can be derived. The complexity of proposed GT approach to determine the Stackelberg equilibrium can be readily obtained as $O(|N| * |S| * \text{ite}_1 * A)$, where ite_1 is the number of iterations needed until convergence of the whole Stackelberg game, and A the complexity of the algorithm solving the maximization problem of (12). Detailed results showing the convergence and complexity to determine the Stackelberg equilibrium are presented in Section V-A. Toward realizing a fully autonomous decision-making approach for the MEC servers' optimal announced prices and alleviating the need for designing specific utility functions as required in the GT approaches, a RL model is introduced in the following.

B. Reinforcement Learning Approach-Fully Autonomous Decision-Making

The proposed RL-based model aims to generate the highest profit for the servers without requiring any knowledge on how their choice affects the amount of data offloaded to them or the pricing of the other servers; the decisions are achieved by simply observing the effects that each server's actions have on its own profit. Toward achieving this goal, we model the decision-making problem as a multiarmed bandit problem [39], that focuses on solving the exploration–exploitation dilemma of a learner (server s) willing to find the best action (price p_s) that maximizes his perceived reward $R(p_s)$. It should be noted that the multiarmed bandit is a special case of the Markov decision process (MDP) in which there is only one state, a set of actions, i.e., prices, and a reward gained by selecting an action (9). Thus, the multiarmed bandit is considered as stateless [40], or equivalently as an one-state MDP. In the multiarmed bandit problem, each action provides a random reward from a probability distribution specific to the action and the MEC server selects the action that generates the highest reward. During this process, a balance should be kept among exploiting the actions that have already been found to perform well and exploring new actions in order to gather more information on the expected reward of the rest of the actions.

Initially, we discretize the pricing strategy space $P_s = [p_{\min}, p_{\max}]$ in distinct actions within a range of a minimum and a maximum price, thus having a set A of M actions $A = \{a_1, \ldots, a_m, \ldots, a_M\}$ where $a_m \in [p_{\min}, p_{\max}]$. Each server chooses at each timeslot a pricing action from the action set A based on which the users play their data offloading game. Thus, at the end of the timeslot the servers observe the obtained reward according to (9), and can decide on the pricing action of the next timeslot.

In order to solve the multiarmed bandit problem, we adopt the upper confidence bound algorithm (UCB1) [41] that has been proven to have a bounded regret. The regret [42] measures the efficiency of the algorithm and corresponds to the difference between the cumulative reward of the proposed action and that of the best possible action. Apart from the regret guarantees, the proposed algorithm allows to fine-tune the range of the confidence interval and enables for better exploration by the MEC servers, which increases the probability of choosing less explored actions.

The main idea of the UCB1 algorithm is that the MEC server keeps a record of the average reward that it obtains, via selecting each action, as well as a confidence interval based on the total number of times the action was selected. Then, instead of choosing the action with the best average reward, it chooses the action with the best upper bound of the interval, meaning that it chooses the action with the best potential. Specifically, the MEC server chooses the action that maximizes the following score:

$$\text{score}_{a_m} = \bar{x}_{a_m} + \sqrt{\frac{2\ln(n_{a_m})}{t}} \tag{16}$$

where a_m is the action, $\bar{x}_{a_m}$ is the average reward experienced by the MEC server for the action a_m, n_{a_m} is the number of times that the action a_m has been chosen and t is the total number of iterations of the algorithm. Similarly as before, the complexity of our proposed approach is $O(|N| * |S| * \text{ite}_2 * A)$, where ite_2 is the number of iterations of just the users' game (see Section III-B). It should be finally clarified, that in general the multiagent RL is a very complex case and therefore, in our work we considered a heuristic approach where each agent separately tries to solve its own multiarmed bandit problem independently from the others. The choices of the rest of the agents are implicitly hidden within the randomness of the perceived reward.

V. Performance Evaluation

In this section, the performance evaluation of the proposed optimization and decision-making framework is realized via modeling and simulation. Initially, we demonstrate the performance of the proposed framework, considering the semiautonomous GT model to determine the MEC servers' optimal prices (Section V-A). Subsequently, in Section V-B, the evaluation is extended to demonstrate the operation and tradeoffs of the adoption of a fully autonomous decision-making approach in determining the MEC servers' prices. The impact of the users' behavioral characteristics on the offloading strategies and the system performance is studied in Section V-C. Finally, Section V-D presents a comparative evaluation of the proposed framework against baseline alternatives to demonstrate its operational superiority and efficiency.

The default system and users' parameters utilized in the following performance evaluation, unless otherwise explicitly stated, are as follows. The total number of users and servers in the examined multiaccess edge computing environment is set to $|N| = 50$ and $|S| = 4$, respectively. The users' amount of input bits b_n, the computation demand of the users' applications i_n, the computation capability of the users' devices f_n, and the users' local consumed energy per CPU cycle follow uniform distributions with mean 10^7 bits, $8 * 10^9$ CPU cycles, $6 * 10^9$ CPU cycles/s, and $4 * 10^{-9}$ Joule/CPU cycles, respectively. Furthermore, for demonstration only purposes, the MEC servers' operational cost is $\boldsymbol{\kappa} = [1, 3, 5, 3] * 10^{-3}$ \$/bit, while the users' behavioral characteristics are captured by the risk-aware parameter $a_n = 0.2$, the loss aversion parameter $k_n = 1.2$, and the distortion parameter $\gamma = 0.6$.

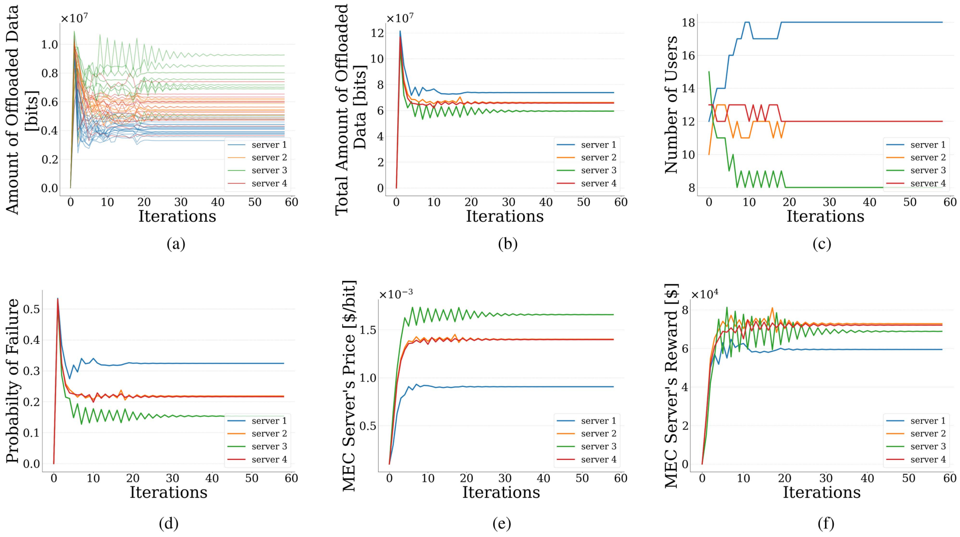
A. Pure Operation Performance Evaluation Under the Semiautonomous Game-Theoretic Decision-Making Model

In this section, we present the pure operation and performance of the proposed framework, considering the semiautonomous GT decision-making of the MEC servers' optimal prices. Initially, we present the evolution of several system parameters of interest as a function of the required iterations for convergence to a stable solution, including both the decision-making parameters under consideration here, namely, the average user offloaded data and the MEC server prices. In particular, Fig. 2(a)–(f) present each user's amount of offloaded data, the total amount of offloaded data per server, the total number of users associated with each server, the servers' probability of failure, the optimal announced prices, and the servers' reward (9), respectively, as a function of the Stackelberg game's iterations. First, we note that the results clearly demonstrate that the overall proposed behavior and price-aware edge computing framework converges quite fast to the Stackelberg equilibrium, i.e., users' optimal data offloading strategies [Fig. 2(a)] and MEC servers' optimal prices [Fig. 2(e)], as for practical purposes less than 40 iterations are needed (corresponding approximately to less than 5 s in simulation time). It is observed that the MEC servers with lower operational cost ($\kappa_1 < \kappa_2 = \kappa_4 < \kappa_3$), announce a lower price [Fig. 2(e)], thus attracting a larger number of users [Fig. 2(c)] which in turn offload an overall larger amount of data [Fig. 2(b)]. However, this strategic decision by some of the MEC servers results in a higher probability of failure [Fig. 2(d)], showing that these servers struggle to process the users' offloaded data. Those servers which are characterized by low operational cost and announce a low price to attract a large portion of the users' computing demand, result in experiencing low reward [Fig. 2(f)]. On the other hand, the servers, with intermediate operational cost announce a conservative price, enjoying a greater reward, even if they process a comparatively intermediate amount of data [Fig. 2(b)].

B. Fully Autonomous Decision-Making Reinforcement Learning Model

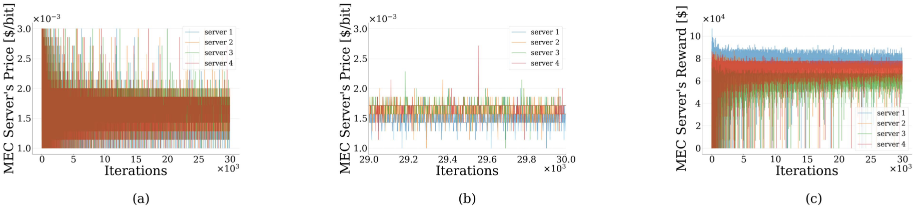
In this section, we extend our previous analysis and evaluation considering that the MEC servers decide their optimal prices without the need of explicitly receiving any external information. Instead they perform exploration and exploitation based on the RL model presented in Section IV-B, toward determining the optimal prices. Based on the insight we gained from the results obtained in Section V-B, for implementation and demonstration purposes, we bound the MEC servers' strategy space as $P_s = [10^{-3}, 3 * 10^{-3}]$ and we equally quantize it in 15 possible actions. Please note that in the following for better understanding and comprehending the operation and achieved system performance by the proposed RL model, the results are discussed, wherever possible, in comparison with the corresponding ones achieved by the semiautonomous GT model.

Specifically, in Fig. 3(a) we present the MEC servers' optimal announced prices for the overall execution period of the RL algorithm as a function of the corresponding iterations, while in Fig. 3(c) the corresponding MEC servers' reward is also presented. To gain some more insight about the algorithm operation and convergence, in Fig. 3(b) the evolution of the the MEC servers' optimal prices during the last 1000 iterations is highlighted. The results demonstrate that initially



(a) (b) (c) (d) (e) (f)

Fig. 2. Pure operation performance under the semiautonomous game-theoretic decision-making approach. (a) Data offloaded by each user to each server. (b) Total amount of offloaded data per server. (c) Number of users associated with each server. (d) Probability of failure of each server. (e) Optimal announced price by each server. (f) Reward experienced by each server.



(a) (b) (c)

Fig. 3. MEC servers prices and rewards based on a fully autonomous reinforcement learning model.

the MEC servers explore several prices to be announced to the users [Fig. 3(a)], as shown by the high price variations in consecutive iterations, but as the RL algorithm thoroughly explores the potential pricing strategies, it finally concludes and converges toward an optimal announced price with very limited exploration [Fig. 3(b)].

Also, it is observed that the fully autonomous decision-making model follows the same trend regarding the MEC servers' announced prices, i.e., $p_1 < p_2 = p_4 < p_3$ [Fig. 3(b)], as the semiautonomous GT model [Fig. 2(e)]. However, the servers with lower operational cost learn better the characteristics of the edge computing environment, and better account for the total amount of processed data [Fig. 2(b)], thus, they announce a higher price (i.e., p_1) in the fully autonomous RL decision-making model [Fig. 3(b) versus the corresponding prices obtained in the semiautonomous GT model Fig. 2(e)]. Thus, the MEC servers with lower operational cost eventually achieve to enjoy a higher reward [Fig. 3(c)] in contrast to the results obtained by the semiautonomous GT decision-making model.

The above obtained results conclude to the following fundamental and interesting observations regarding the fully autonomous RL and the semiautonomous GT decision-making models. Both of them result in similar benefits regarding the users' computing requests' satisfaction, their corresponding achieved utility, and their optimal data offloading strategies. On the other hand, the RL-based model supports better the free market competition among the MEC servers, which autonomously learn and decide the optimal announced prices, without the need for the involvement of a (centralized) market regulatory entity. In this case, the MEC servers operate in a myopic and selfish manner resulting in higher achieved rewards, even for the servers that announce lower prices. On the other hand, the GT-based decision-making model concludes faster to the users' optimal data offloading strategies and the MEC servers' optimal announced prices, compared to the RL-based model.

C. *Impact of Users' Behavioral Characteristics*

In this section, we study the impact of the users' behavioral characteristics, as they are captured by the risk-aware parameter α_n and the loss aversion parameter k_n, on the overall system's operation. Concerning the different values of the risk-aware parameter α_n, Fig. 4 shows that the higher the value of α_n is, the higher is the amount of data that the user offloads. By increasing α_n, the sensitivity on gains as well as on losses increases exponentially based on (6). Also, given that the amount of offloaded data increases, the probability of failure increases as well. However, still the MEC servers are more likely to succeed in fulfilling their tasks, thus, making the first branch of (6) more likely to occur. As a result, increasing the exponent α_n results in higher expected utility for the users based on (8). In turn, due to the users' higher willingness to offload data, the servers have the flexibility to announce higher prices, which in combination with the larger volume of the offloaded data, lead to higher rewards for the servers.

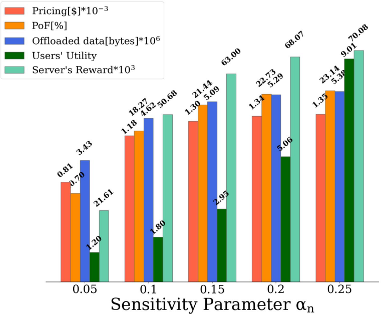


Fig. 4. Offloading strategies and system performance versus risk-aware parameter α_n.

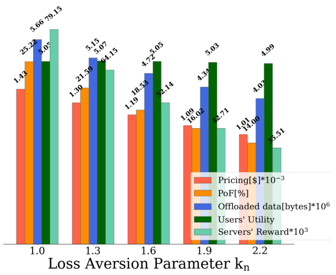


Fig. 5. Offloading strategies and system performance versus loss aversion parameter k_n.

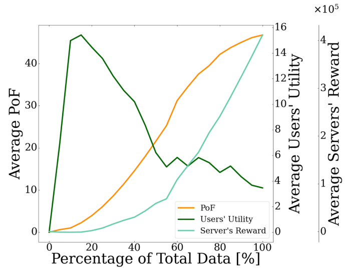


Fig. 6. System performance versus offloaded data.

On the other hand, increasing the loss aversion parameter k_n, which expresses how the users weigh the losses, has the opposite effect. As k_n increases, Fig. 5 shows that the users tend to offload less data in order to reduce their potential losses. It is noted that the gains experienced by the users are not affected by the parameter k_n as we can see from (6). Also, we observe that the users' expected utility decreases by a smaller amount as compared to the corresponding changes in the risk-aware parameter α_n, due to the linear dependence of the utility function on the parameter k_n. Also, the MEC servers' probability of failure maintain relatively low values, signifying that losses eventually play a less important role in the resulting expected utility. Due to the decreasing amount of offloaded data, the servers aim at incentivizing the users to offload more data by announcing lower prices to the users, leading to an overall decreased reward for the servers.

Moreover, in the following Fig. 6, we present the tradeoffs between different system performance metrics while forcing the users to offload specific portions of their data. Specifically, by keeping all the users' data locally for processing, the users experience zero utility, since the local processing depicts the reference point of the prospect-theoretic utility function (6). Consequently, since no data are offloaded, the MEC servers announced price and their probability of failure remain zero. In the scenario that very few data are offloaded, the servers impose low prices since they want to incentivize users to offload more data, resulting in low rewards. On the other hand, the users still experience high utility, given the fact that the server is guaranteed to succeed in executing their tasks due to the low probability of failure. However, when the percentage of offloaded data exceeds a threshold (e.g., 20% of the users' data in our case), the users tend to experience lower utility. This phenomenon is observed due to the fact that even if the users offload more data and enjoy the servers' computing capabilities, the servers ultimately increase their prices to compensate for their offered computing resources. The above threshold depends on the users' behavioral characteristics, as well as on the servers' computing capabilities and market competition. Additionally, due to the increased exploitation of the MEC servers' computing resources, the probability of failure increases, leading to further decrease in the users' expected utility.

D. Comparative Evaluation

Subsequently, we present a detailed comparative evaluation of the proposed framework—under the two operational alternatives and models—against four different benchmarking scenarios, with respect to determining the optimal MEC server's prices. In particular, we compare the proposed fully autonomous RL model and the semiautonomous GT one, against the following strategies: 1) RL-AVG, where the MEC servers constantly announce the average prices that the RL model has learned over 30 000 iterations, 2) MAX, 3) MIN, and 4) RANDOM, where the MEC servers always announce a maximum, minimum, and random price to the users, respectively.

Fig. 7(a) and (b) demonstrate the cumulative MEC servers' rewards (9) over the iterations of the RL model, over two different scenarios corresponding to 100 and 30 000 iterations, respectively. The results reveal that the MAX scenario, as expected, constantly presents the worst rewards for the MEC servers, as their computing services become extremely expensive for the users, and the latter ones prefer to locally process their data on their devices. On the other hand, the RL-AVG scenario constantly achieves the best rewards for the MEC servers, as they always announce the educated optimal prices that the RL-model has chosen. The MIN and RANDOM scenarios on the other hand, present worse results than the GT and the RL models, in particular after the point that the latter one has performed sufficient exploration [Fig. 7(b)] of the available pricing strategies. Thus, even if the MEC servers set a low price to attract more users (MIN scenario), this decision results in worse rewards compared to the optimal decision-making performed by the GT and RL scenarios, due to the combined effect of the low price and the phenomenon of the tragedy of the commons, which results in the overexploitation of the MEC servers' computing resources.

Placing our emphasis on the GT and RL scenarios, we observe that the GT model achieves fast a stable optimal outcome (Fig. 7), while the RL model progressively explores the MEC servers' strategy space and eventually results in similar, and even slightly better rewards for the MEC servers. Moreover, Fig. 7(c) comes as a verification to the above argument and observation, since even though initially the RL leads to greater regret for the servers compared to the GT approach, after approximately

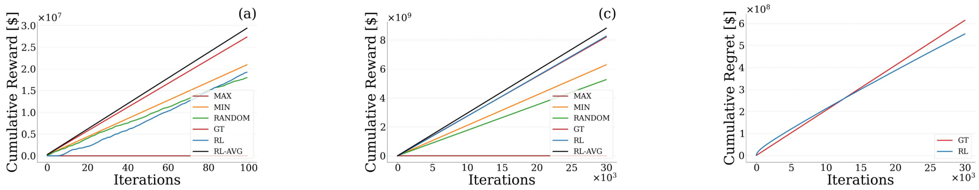


Fig. 7. Cumulative reward of servers: Comparative evaluation.

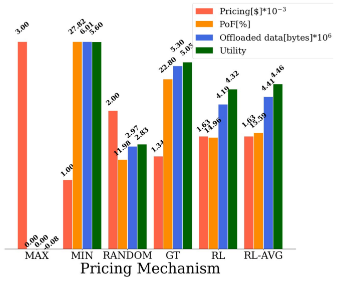


Fig. 8. System Performance for various pricing mechanisms.

12 000 iterations this trends reverses and the RL approach leads to lower and diminishing regret, thus becoming more favorable in the long run. Please recall that the regret as it has been defined in Section IV-B, represents the difference between the cumulative profit that the servers would have obtained if they had been playing the best pricing strategy from the beginning (which in practice is unknown and is only theoretical)—here is the RL-AVG strategy—and the cumulative profit that the servers actually receive until iteration i under the corresponding strategy. The latter observation is well aligned with the findings in Section V-B, where it was concluded that the RL model benefits more the MEC servers, presenting superior rewards compared to the GT model, while guaranteeing similar performance for the users.

Finally, in Fig. 8 we present a comprehensive evaluation of various system performance metrics for all the different considered alternative strategies, in order to better validate the relative efficiency and effectiveness of our proposed approaches, in a more holistic manner. Specifically, we observe that both GT and the RL approaches outperform all the alternative baseline methods in balancing the rewards for both users and servers. For instance, selecting the RANDOM approach may result in lower probability of failure for the servers as less data are offloaded to them, however, relatively poor performance is observed with respect to the rest of the metrics, noting that both users' utility and offloading data (Fig. 8) and servers' profit (Fig. 7) remain low. On the other hand, by setting a constant pricing equal to the minimum one (i.e., MIN), users offload more data to the servers thus achieving greater utility; however, this happens at the cost of reduced reward for the servers (Fig. 7). On the opposite side, setting a constant pricing equal to the maximum price (i.e., MAX) forces users to keep all their data for local execution, thus resulting in almost zero probability of failure, but extremely low reward for the users.

Turning our attention to the GT approach, from the results in Fig. 8 we notice that it presents a final solution more beneficial for the users, since the corresponding game converges to a stable outcome with lower average price than its counterpart of the RL approach. This in turn allows the offloading of a greater amount of data to the servers, and consequently results in higher perceived expected utility by the users. On the other hand, the RL approach presents a behavior that favours the servers perspective. That is, though the higher concluding price leads to lower offloaded data and expected utility, it still allows for higher profit for the servers [Fig. 7(c)]. It should also be noted that the RL approach, as expected, closely follows the performance of the constant price of the average RL pricing, which strengthens our case and arguments regarding obtaining low regret values for the respective servers' choices.

VI. Conclusion

In this article, we proposed a behavior and price-aware multiuser–multiserver–multiaccess edge computing operation framework, conceptualized and realized based on the principles of prospect theory, game theory, and RL. The users' behavior on the one hand, and the potential servers' computing resource usage and overexploitation on the other hand, are captured via appropriately designed prospect-theoretic utility functions and the theory of the tragedy of the commons, respectively. The interactions among the users and the MEC servers are captured via a Stackelberg game. A noncooperative game among the users is introduced to determine their optimal data offloading strategies to the MEC servers, while a GT and an RL model are proposed, in order to enable the MEC servers to determine their optimal announced prices in a semi and fully autonomous manner, respectively. The performance evaluation of the proposed framework is obtained via modeling and simulation, while its superiority against other basic benchmarking alternatives is demonstrated.

Part of our current and future work targets at extending the proposed framework via considering the edge computing market dynamics following a more holistic labor economics-based approach. A dynamic and personalized pricing mechanism where the actual price depends on the capabilities of the user in order to favor less powerful devices, may promote fair usage of the network resources and provide greater control over the proposed framework. It is also noted that in the current work we focused on the introduction of a communication agnostic data offloading framework, in order to better evaluate the impact of the users behavioral characteristics and the MEC servers pricing policies on the resulting strategies. However the incorporation of communication considerations (e.g., interference and/or achievable transmission rate) in the overall proposed framework is of high practical importance.

Appendix A
Proof of Theorem 1

Toward proving that the game G has at least one NE, we can show that the game G is submodular by proving the properties in Definition 2 [36]. Since the strategy space of the game $\mathcal{B}_n =$

$[0, b_n]$ is closed and bounded, $\mathcal{B}_n \forall n \in N$ is a compact subset of the Euclidean space. Additionally, the prospect-theoretic utility function in (8) is by definition smooth since we can calculate its derivatives of any order in $\mathcal{B}_n$.

By using (2)–(6), we can rewrite (8) as

$$\mathbb{E}\left(P_{n,s}\left(\mathbf{b}_s^{\text{MEC}}\right)\right) = \left(b_{n,s}^{\text{MEC}}\right)^{\alpha_n} \left\{ \overline{R}(D_s)(1-\pi(Pr_s)) - k_n \left(\frac{1}{t_n e_n} + p_s \frac{i_n}{b_n}\right)^{\alpha_n} \pi(Pr_s) \right\} \tag{17}$$

where we substitute $\overline{R}(D_s) = \left[R(D_s) - \frac{1}{t_n e_n} - p_s \frac{i_n}{b_n}\right]^{\alpha_n}$ for notation purposes. The $\overline{R}(D_s)$ corresponds to the users' specific rate of return which in our work should be by definition positive so that the users have incentive to offload their data to the MEC servers.

Toward determining the minimum value of $\overline{R}(D_s)$, we observe based on (4) that the function $R(D_s)$ is decreasing with respect to D_s, and thus, its minimum value corresponds to $D_s = 1$. In order to guarantee that the users have the incentive to offload their data to the MEC server he have the following:

$$\overline{R}(D_s = 1) > 0 \Rightarrow p_s < \frac{b_n}{i_n}\left(1 - \frac{1}{t_n e_n}\right) \tag{18}$$

providing a boundary on the price that the servers can impose before the users choose *a priori* to locally process all their data. Also, we have: $\frac{\partial D_s}{\partial b_{n,s}^{\text{MEC}}} > 0$, $\frac{\partial D_s}{\partial b_{j,s}^{\text{MEC}}} > 0$, $\frac{\partial \overline{R}(D_s)}{\partial b_{n,s}^{\text{MEC}}} < 0$, $\frac{\partial \overline{R}(D_s)}{\partial b_{j,s}^{\text{MEC}}} < 0$, $\frac{\partial^2 \overline{R}(D_s)}{\partial b_{j.s}^{\text{MEC}} \partial b_{n.s}^{\text{MEC}}} < 0$, $\frac{\partial Pr_s(D_s)}{\partial b_{n,s}^{\text{MEC}}} > 0$, $\frac{\partial Pr_s(D_s)}{\partial b_{j,s}^{\text{MEC}}} > 0$, $\frac{\partial^2 Pr_s(D_s)}{\partial b_{n,s}^{\text{MEC}} \partial b_{j,s}^{\text{MEC}}} = 0$, $\frac{\partial \pi(Pr_s)}{\partial b_{n,s}^{\text{MEC}}} > 0$, $\frac{\partial \pi(Pr_s)}{\partial b_{j,s}^{\text{MEC}}} > 0$, $\frac{\partial^2 \pi(Pr_s)}{\partial b_{n,s}^{\text{MEC}} \partial b_{j,s}^{\text{MEC}}} = 0$, when $\gamma \in (0, 1)$. Additionally, and for notation purposes, we set $A = k_n(\frac{1}{t_n e_n} + p_s \frac{i_n}{b_n})^{\alpha_n} > 0$, and thus, (17) becomes: $\mathbb{E}(P_n(\mathbf{b}_s^{\text{MEC}})) = (b_{n,s}^{\text{MEC}})^{\alpha_n}\{\overline{R}(D_s)(1-\pi(Pr_s)) - A\pi(Pr_s)\}$. Then, we proceed in calculating the second-order partial derivative of the user's expected prospect-theoretic utility function

$$\begin{aligned}\frac{\partial^2 \mathbb{E}\left(P_n\left(\mathbf{b}_s^{\text{MEC}}\right)\right)}{\partial b_{j,s}^{\text{MEC}} \partial b_{n,s}^{\text{MEC}}} &= \alpha_n \left(b_{n,s}^{\text{MEC}}\right)^{\alpha_n - 1} \left\{ \frac{\partial \overline{R}\left(D_s\right)}{\partial b_{j,s}^{\text{MEC}}}[1-\pi(Pr_s)] - \overline{R}(D_s)\frac{\partial \pi(Pr_s)}{\partial b_{j,s}^{\text{MEC}}} - A\frac{\partial \pi(Pr_s)}{\partial b_{j,s}^{\text{MEC}}} \right\} \\ &\quad + \left(b_{n,s}^{\text{MEC}}\right)^{\alpha_n} \left\{ \frac{\partial^2 \overline{R}(D_s)}{\partial b_{j,s}^{\text{MEC}} \partial b_{n,s}^{\text{MEC}}}[1-\pi(Pr_s)] - \frac{\partial \overline{R}(D_s)}{\partial b_{n,s}^{\text{MEC}}}\frac{\partial \pi(Pr_s)}{\partial b_{j,s}^{\text{MEC}}} - \frac{\partial \overline{R}(D_s)}{\partial b_{j,s}^{\text{MEC}}}\frac{\partial \pi(Pr_s)}{\partial b_{n,s}^{\text{MEC}}} \right\} \end{aligned} \tag{19}$$

$$\begin{aligned} &= (b_{n,s}^{\text{MEC}})^{\alpha_n - 1} \left\{ \alpha_n \frac{\partial \overline{R}(D_s)}{\partial b_{j,s}^{\text{MEC}}}[1-\pi(Pr_s)] - \alpha_n \overline{R}(D_s)\frac{\partial \pi(Pr_s)}{\partial b_{j,s}^{\text{MEC}}} - A\alpha_n \frac{\partial \pi(Pr_s)}{\partial b_{j,s}^{\text{MEC}}} \right. \\ &\quad + b_{n,s}^{\text{MEC}} \frac{\partial^2 \overline{R}(D_s)}{\partial b_{j,s}^{\text{MEC}} \partial b_{n,s}^{\text{MEC}}}[1-\pi(Pr_s)] \\ &\quad \left. -b_{n,s}^{\text{MEC}} \frac{\partial \overline{R}(D_s)}{\partial b_{n,s}^{\text{MEC}}}\frac{\partial \pi(Pr_s)}{\partial b_{j,s}^{\text{MEC}}} - b_{n,s}^{\text{MEC}} \frac{\partial \overline{R}(D_s)}{\partial b_{j,s}^{\text{MEC}}}\frac{\partial \pi(Pr_s)}{\partial b_{n,s}^{\text{MEC}}} \right\} \end{aligned}$$

and by setting $\psi(D_s) = \frac{\partial \overline{R}(D_s)}{\partial b_{j,s}^{\text{MEC}}}\big[\alpha_n - \alpha_n \pi(Pr_s) - b_{n,s}^{\text{MEC}} \frac{\partial \pi(Pr_s)}{\partial b_{n,s}^{\text{MEC}}}\big] - b_{n,s}^{\text{MEC}} \frac{\partial \overline{R}(D_s)}{\partial b_{n,s}^{\text{MEC}}} \frac{\partial \pi(Pr_s)}{\partial b_{j,s}^{\text{MEC}}}$, we can rewrite (19), as follows:

$$\begin{aligned}\frac{\partial^2 \mathbb{E}(P_n(\mathbf{b}_s^{\text{MEC}}))}{\partial b_{j,s}^{\text{MEC}} \partial b_{n,s}^{\text{MEC}}} &= (b_{n,s}^{\text{MEC}})^{\alpha_n - 1} \left\{ \psi(D_s) - \alpha_n \overline{R}(D_s)\frac{\partial \pi(Pr_s)}{\partial b_{j,s}^{\text{MEC}}} - A\alpha_n \frac{\partial \pi(Pr_s)}{\partial b_{j,s}^{\text{MEC}}} \right. \\ &\quad \left. + b_{n,s}^{\text{MEC}} \frac{\partial^2 \overline{R}(D_s)}{\partial b_{j,s}^{\text{MEC}} \partial b_{n,s}^{\text{MEC}}}[1-\pi(Pr_s)] \right\}. \end{aligned} \tag{20}$$

The compact form of (20) allows us to extract meaningful information regarding the sign of the resulting equation. Since the last three terms are negative based on the aforementioned derivatives, in order to study the equation we can focus on the first term, $\psi(D_s)$. Specifically, we can examine the two cases where $D_s = 0$ and $D_s \approx 1$ to study the properties of the function $\psi(D_s)$.

For $D_s = 0$, we have

$$\psi(0) = \frac{\partial \overline{R}(0)}{\partial b_{j,s}^{\text{MEC}}}\alpha_n < 0 \tag{21}$$

while for $D_s \approx 1$, we have

$$\begin{aligned}&\psi(D_s \approx 1) \\ &= -b_{n,s}^{\text{MEC}} \left[\frac{\partial \overline{R}(1)}{\partial b_{j,s}^{\text{MEC}}} \frac{\partial \pi(Pr_s(1))}{\partial b_{n,s}^{\text{MEC}}} + \frac{\partial \overline{R}(1)}{\partial b_{n,s}^{\text{MEC}}} \frac{\partial \pi(Pr_s(1))}{\partial b_{j,s}^{\text{MEC}}} \right] > 0. \end{aligned} \tag{22}$$

Following the Bolzano theorem [37], since $\psi(D_s)$ is a continuous function on D_s, there exists at least one value $x \in (0, 1)$ such that $\psi(x) = 0$. We have already shown that $\psi(0) < 0$, and thus, if x is the smallest value in (0, 1), where $\psi(x) = 0$, that means that $\psi(D_s) < 0 \forall D_s \in (0, x)$.

Continuing on (20), we have proven that

$$\frac{\partial^2 \mathbb{E}(P_n(\mathbf{b_n^{MEC}}))}{\partial b_{j,s}^{\text{MEC}} \partial b_{n,s}^{\text{MEC}}} < 0, \forall D_s \in (0, x), x \in (0, 1). \tag{23}$$

Based on the above, we can conclude that the noncooperative game G is submodular $\forall D_s \in (0, x)$ given that $p_s < \frac{b_n}{d_n}\left(1 - \frac{1}{t_n e_n}\right)$ and $\gamma \in (0, 1)$, and thus, that the game G has at least one Pure NE point.

References

[1] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A survey on mobile edge computing: The communication perspective," *IEEE Comm. Surv. Tuts.*, vol. 19, no. 4, pp. 2322–2358, Oct.–Dec. 2017.

[2] X. Chen, "Decentralized computation offloading game for mobile cloud computing," *IEEE Trans. Parallel Distrib. Syst.*, vol. 26, no. 4, pp. 974–983, Apr. 2015.

[3] P. A. Apostolopoulos, E. E. Tsiropoulou, and S. Papavassiliou, "Risk-aware data offloading in multi-server multi-access edge computing environment," *IEEE/ACM Trans. Netw.*, vol. 28, no. 3, pp. 1405–1418, Jun. 2020.

[4] Y. Liu, F. R. Yu, X. Li, H. Ji, and V. C. Leung, "Distributed resource allocation and computation offloading in fog and cloud networks with non-orthogonal multiple access," *IEEE Trans. Veh. Technol.*, vol. 67, no. 12, pp. 12137–12151, Dec. 2018.

[5] L. Yang, H. Zhang, X. Li, H. Ji, and V. C. Leung, "A distributed computation offloading strategy in small-cell networks integrated with mobile edge computing," *IEEE/ACM Trans. Netw.*, vol. 26, no. 6, pp. 2762–2773, Dec. 2018.

[6] Z. Ning et al., "Partial computation offloading and adaptive task scheduling for 5G-enabled vehicular networks," *IEEE Trans. Mobile Comput.*, vol. 21, no. 4, pp. 1319–1333, Apr. 2022.

[7] Z. Ning et al., "Mobile edge computing enabled 5G health monitoring for Internet of Medical Things: A decentralized game theoretic approach," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 2, pp. 463–478, Feb. 2021.

[8] X.-L. Huang, X. Ma, and F. Hu, "Machine learning and intelligent communications," *Mobile Netw. Appl.*, vol. 23, no. 1, pp. 68–70, 2018.

[9] S. Ghoorchian and S. Maghsudi, "Multi-armed bandit for energy-efficient and delay-sensitive edge computing in dynamic networks with uncertainty," *IEEE Trans. Cogn. Comm. Netw.*, vol. 7, no. 1, pp. 279–293, Mar. 2021.

[10] P. Dai et al., "Multi-armed bandit learning for computation-intensive services in MEC-empowered vehicular networks," *IEEE Trans. Veh. Technol.*, vol. 69, no. 7, pp. 7821–7834, Jul. 2020.

[11] S. Misra, S. P. Rachuri, P. K. Deb, and A. Mukherjee, "Multi-armed bandit-based decentralized computation offloading in fog-enabled IoT," *IEEE Internet Things J.*, vol. 8, no. 12, pp. 10010–10017, Jun. 2021.

[12] R. Saha, S. Misra, and P. K. Deb, "FogFL: Fog-assisted federated learning for resource-constrained IoT devices," *IEEE Internet Things J.*, vol. 8, no. 10, pp. 8456–8463, May 2021.

[13] S. Misra, T. Ojha, and A. Mondal, "Game-theoretic topology control for opportunistic localization in sparse underwater sensor networks," *IEEE Trans. Mobile Comp.*, vol. 14, no. 5, pp. 990–1003, May 2015.

[14] A. Mondal, S. Misra, and M. S. Obaidat, "Distributed home energy management system with storage in smart grid using game theory," *IEEE Syst. J.*, vol. 11, no. 3, pp. 1857–1866, Sep. 2017.

[15] P. K. Bishoyi and S. Misra, "Enabling green mobile-edge computing for 5G-based healthcare applications," *IEEE Trans. Green Commun. Netw.*, vol. 5, no. 3, pp. 1623–1631, Sep. 2021.

[16] Y. Wu, K. Ni, C. Zhang, L. P. Qian, and D. H. K. Tsang, "NOMA-assisted multi-access mobile edge computing: A joint optimization of computation offloading and time allocation," *IEEE Trans. Veh. Technol.*, vol. 67, no. 12, pp. 12244–12258, Dec. 2018.

[17] K. Xie et al., "Distributed multi-dimensional pricing for efficient application offloading in mobile cloud computing," *IEEE Trans. Serv. Comput.*, vol. 12, no. 6, pp. 925–940, Nov./Dec. 2019.

[18] M. Liu and Y. Liu, "Price-based distributed offloading for mobile-edge computing with computation capacity constraints," *IEEE Wireless Commun. Lett.*, vol. 7, no. 3, pp. 420–423, Jun. 2018.

[19] Y. Chen, Z. Li, B. Yang, K. Nai, and K. Li, "A Stackelberg game approach to multiple resources allocation and pricing in mobile edge computing," *Future Gener. Comp. Syst.*, vol. 108, pp. 273–287, 2020.

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

[21] D. Han, W. Chen, and Y. Fang, "A dynamic pricing strategy for vehicle assisted mobile edge computing systems," *IEEE Wireless Commun. Lett.*, vol. 8, no. 2, pp. 420–423, Apr. 2019.

[22] B. K. Saha, S. Misra, and S. Pal, "Utility-based exploration for performance enhancement in opportunistic mobile networks," *IEEE Trans. Comput.*, vol. 65, no. 4, pp. 1310–1322, Apr. 2016.

[23] M. Lewandowski, "Prospect theory versus expected utility theory: Assumptions, predictions, intuition and modelling of risk attitudes," *Central Eur. J. Econ. Modelling Econometrics*, vol. 9, no. 4, pp. 275–321, 2017.

[24] P. A. Apostolopoulos, G. Fragkos, E. E. Tsiropoulou, and S. Papavassiliou, "Data offloading in UAV-assisted multi-access edge computing systems under resource uncertainty," *IEEE Trans. Mobile Comput.*, to be published, doi: 10.1109/TMC.2021.3069911.

[25] G. Mitsis, E. E. Tsiropoulou, and S. Papavassiliou, "Data offloading in UAV-assisted multi-access edge computing systems: A resource-based pricing and user risk-awareness approach," *Sensors*, vol. 20, no. 8, 2020, Art. no. 2434.

[26] G. Hardin, "The tragedy of the commons," *J. Natural Resour. Policy Res.*, vol. 1, no. 3, pp. 243–253, 2009.

[27] 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, 2016.

[28] 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.*, 2019, pp. 2134–2142.

[29] 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.

[30] S. R. Etesami, W. Saad, N. B. Mandayam, and H. V. Poor, "Smart routing of electric vehicles for load balancing in smart grids," *Automatica*, vol. 120, 2020, Art. no. 109148.

[31] L. Xiao, N. B. Mandayam, and H. V. Poor, "Prospect theoretic analysis of energy exchange among microgrids," *IEEE Trans. Smart Grid*, vol. 6, no. 1, pp. 63–72, Jan. 2015.

[32] S. Tu et al., "Mobile fog computing security: A user-oriented smart attack defense strategy based on DQL," *Comput. Commun.*, vol. 160, pp. 790–798, 2020.

[33] M. Abdallah, P. Naghizadeh, A. R. Hota, T. Cason, S. Bagchi, and S. Sundaram, "Behavioral and game-theoretic security investments in interdependent systems modeled by attack graphs," *IEEE Trans. Control Netw. Syst.*, vol. 7, no. 4, pp. 1585–1596, Dec. 2020.

[34] K. Zhang, J. Cao, S. Leng, C. Shao, and Y. Zhang, "Mining task offloading in mobile edge computing empowered blockchain," in *Proc. IEEE Intern. Conf. Smart IoT*, 2019, pp. 234–239.

[35] D. Prelec, "The probability weighting function," *Econometrica*, vol. 66, pp. 497–527, 1998.

[36] Y. Zhang and M. Guizani, *Game Theory for Wireless Communications and Networking*. Boca Raton, FL, USA: CRC Press, 2011.

[37] T. M. Apostol, *Calculus*, vol. 1. Hoboken, NJ, USA: Wiley, 1991.

[38] J. B. Rosen, "Existence and uniqueness of equilibrium points for concave n-person games," *Econometrica: J. Econometric Soc.*, vol. 33, pp. 520–534, 1965.

[39] J. C. Gittins, "Bandit processes and dynamic allocation indices," *J. Roy. Stat. Soc.: Ser. B*, vol. 41, no. 2, pp. 148–164, 1979.

[40] T. Lattimore and C. Szepesvári, *Bandit Algorithms*. Cambridge, U.K.: Cambridge Univ. Press, 2020.

[41] P. Auer, N. Cesa-Bianchi, and P. Fischer, "Finite-time analysis of the multiarmed bandit problem," *Mach. Learn.*, vol. 47, no. 2, pp. 235–256, 2002.

[42] A. Badanidiyuru, J. Langford, and A. Slivkins, "Resourceful contextual bandits," in *Proc. Conf. Learn. Theory*, 2014, pp. 1109–1134.

Giorgos Mitsis received the diploma and Ph.D. degrees in electrical and computer engineering from the National Technical University of Athens (NTUA), Athens, Greece, in 2015 and 2021, respectively.

He is currently a Research Associate with the Network Management and Optimal Design Laboratory, NTUA, where he is a Software Engineer in various R&D projects. His research interests include resource orchestration in constrained interdependent cyber-physical systems.

Eirini Eleni Tsiropoulou (Senior Member, IEEE) received the diploma and Ph.D. degrees in electrical and computer engineering from the National Technical University of Athens, in 2008 and 2014, respectively.

She is currently an Assistant Professor with the Department of Electrical and Computer Engineering, University of New Mexico, Albuquerque, NM, USA. Her research interests include cyber-physical systems, with emphasis on network modeling and optimization, resource orchestration, reinforcement learning, game theory, and network economics.

Dr. Tsiropoulou was the recipient of numerous awards, including the NSF CRII Award, in 2019, and the Early Career Award by the IEEE Communications Society Internet Technical Committee, in 2019.

Symeon Papavassiliou (Senior Member, IEEE) received the diploma degree in electrical and computer engineering from the National Technical University of Athens in 1990, and the M.Sc and Ph.D. degrees in electrical engineering from Polytechnic University, New York, USA, in 1992 and 1996, respectively.

He is currently a Professor with the School of ECE, National Technical University of Athens, Athens, Greece. He was a Senior Technical Staff Member with AT&T Laboratories, Middletown Township, NJ, USA, from 1995 to 1999, and an Associate Professor with the ECE Department, New Jersey Institute of Technology, Newark, NJ, USA, until 2004. He has authored or coauthored more than 350 journal and conference papers. His research interests include computer communication networks, with emphasis on the optimization of wireless networks and complex systems.