Optimal Offloading Strategies for Edge-Computing via Mean-Field Games and Control

Kai Cui, Mustafa Burak Yilmaz, Anam Tahir, Anja Klein, Heinz Koeppl

Department of Electrical Engineering and Information Technology

Technische Universität Darmstadt

Darmstadt, Germany

{kai.cui, burak.yilmaz, anam.tahir, anja.klein, heinz.koeppl}@tu-darmstadt.de

Abstract—The optimal offloading of tasks in heterogeneous edge-computing scenarios is of great practical interest, both in the selfish and fully cooperative setting. In practice, such systems are typically very large, rendering exact solutions in terms of cooperative optima or Nash equilibria intractable. For this purpose, we adopt a general mean-field formulation in order to solve the competitive and cooperative offloading problems in the limit of infinitely large systems. We give theoretical guarantees for the approximation properties of the limiting solution and solve the resulting mean-field problems numerically. Furthermore, we verify our solutions numerically and find that our approximations are accurate for systems with dozens of edge devices. As a result, we obtain a tractable approach to the design of offloading strategies in large edge-computing scenarios with many users.

Index Terms—edge-computing, mean-field, computation offloading, Nash equilibria, Pareto optima

Fig. 1. MEC scenario with N UEs offloading their tasks to M MEC servers, where computation resources of MEC servers are shared through a fiber loop connecting them, acting as a single processing pool.

I. INTRODUCTION

In recent years, a rapid growth of data generated from the network edge is witnessed, especially, the Cisco Annual Internet Report 2020 forecasts a rapid deployment of billions of Machine To Machine (M2M) devices until 2023 [1]. Multi-access Edge Computing (MEC) is a key technology to compensate strictly limited M2M devices in their processing by enabling computation offloading to cloudlet servers with computation resources in their vicinity. Additionally, the number of User Edge devices (UE) like smartphones, tablets and laptops also have increased tremendously due to the ease of their availability and low costs. These devices can also gain from offloading their tasks that demand intensive computations and low latencies, e.g., virtual reality, real-time face recognition, natural language processing, to such cloudlets.

A MEC system can be seen as a multi-agent system where each UE is an agent who needs to decide, for each arriving task, whether to offload it to the MEC server or not. There has been great interest in finding the optimal policy for these UEs, to either offload or process locally, depending on factors such as their own available resources, network conditions and offloading computation costs. Even though several computation offloading strategies between UEs and MEC servers have been proposed in the literature, finding scalable solutions for MEC
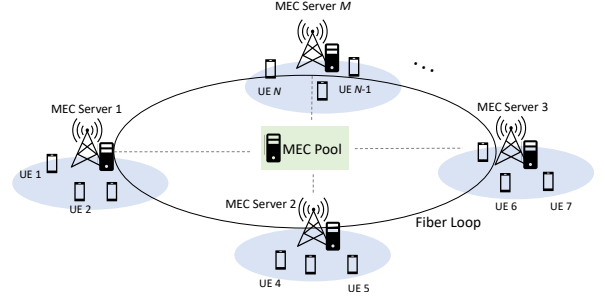
multiagent systems remains an important problem considering the continuously increasing number of agents.

Mean-field approximations are becoming an increasingly popular approach to resolve the curse of dimensionality in large-scale multi-agent systems [2]. The idea is to represent a many-agent system as a single-representative-agent system which interacts with the empirical state distribution (or the mean-field) of all the other agents in the system, making the system tractable. It has been shown that the solution to a mean-field system is a good approximation to an N-agent system as N grow large [3]. There are two main categories in which agents can work in a multi-agent system, either cooperatively to maximize a global goal or competitively to maximize their own reward, or a combination of both. Mean-Field Games (MFGs, [4], [5], [6], [7]) provide a way to analyze and solve large-scale competitive problems in a tractable manner, in particular under the usage of learning-based techniques [8], [9]. On the other hand, Mean-Field Control (MFC) is used to model cooperative settings in many-agent systems [10], [11], see [12], [13], [14], [15], and references therein for some interesting applications. Similarly, we will formulate offloading in edge-computation as both a one-shot problem with theoretical guarantees and alternatively a time-stationary problem, allowing for competitive and cooperative solutions in a unified, tractable manner.

The decision to offload tasks to MEC has been an important area of research since it is crucial to find a good, if not optimal,

policy in scalable applications. Various prior works have used mean-field approximations for similar offloading problems. In [16], the authors model a shared MEC competitive of-floading problem as an MFG in continuous time and solve the resulting coupled Hamilton-Jacobi-Bellman and Fokker-Plank partial differential equations using fixed-point iteration. However, their model considers only the non-cooperative case and results in a continuous time model, while in our work we also consider a cooperative setting and obtain a model for the time-stationary case. Similarly, in [17], [18], the authors consider both cooperative and non-cooperative computational offloading problems, though through the special case of a linear-quadratic model, while we solve a non-linear problem.

Mean-field approximations have also been used to model large-scale MEC systems with D2D collaborations [19] particularly on graphs as a deterministic ODE system, though the model does not consider entirely selfish nodes. Finally, authors in [20] model and solve a large-scale resource-sharing problem using mean-field theory and both cooperative and non-cooperative strategies, i.e. a more centralized setting without local computational capabilities. In contrast to our work, their models focus on graph-based job forwarding and continuous-action resource-sharing problems, whereas our model focuses on offloading decisions. For a variety of other works on applications of the mean-field approach in communication systems, see also e.g. [21], [22], [23], [24]. Apart from the discussed differences, all of the models in prior works and our work are diverse and apply to a variety of differing scenarios. Here, in contrast to previous work, our model will consider the both cooperative and competitive optimization of binary offloading decisions in edge-computing, where users may choose to offload or compute locally. In this work, we will formulate a unified mean-field framework for both the competitive and cooperative setting of offloading decisions in edge-computing. In particular, our tractable solution considers both a one-shot and time-stationary scenario that is rigorously approximating the finite user system.

We begin our analysis with a computationally expensive-to-solve one-shot game, a knapsack problem, where many edge devices must independently decide whether to offload or not for a given distribution of task configurations. Our contribution can be summarized as follows: (i) We pass to the infinite-user limit in order to obtain a tractable problem with a complexity independent of the number of users; (ii) The model is theoretically motivated by showing novel existence and approximate optimality properties of solutions in large finite systems, both in terms of Nash equilibria and Pareto optima; (iii) Since, in practice, a one-shot game may not be sufficiently realistic, we extend our model to a new time-stationary model with a Poisson task arrival process and find analogous competitive and cooperative solutions in the limit of large systems; and finally (iv), the proposed models are verified in simulation and solved or learned with complexity independent of the number of users, while concurrently giving a good solution to large finite systems. As a result, we tractably solve an otherwise intractable many-agent offloading problem.

II. MATHEMATICAL MODEL

In the considered scenario, a multi-cell ultra-dense network includes M MEC servers and a total number of N UEs associated with these MEC servers. It is assumed that M MEC servers are connected through a fiber loop to share their computational resources, pooled into a single centralized but distributed MEC pool, where we also assume the bandwidth across the MEC servers is large enough for connecting all UEs and the delay in resource sharing through fiber loop is neglected, similar to the framework presented in [16]. The scenario is depicted in Fig. 1 with M MEC servers and N UEs. Each UE, modeled by $i = 1, \dots, N$, is given a configuration

$$C^i := (W^i, L^i, f^i, R^i) \in \mathbb{K} \subseteq \mathbb{R}^4_{\geq 0} \tag{1}$$

with task transmission length W^i in bits, task processing complexity L^i in CPU cycles to be computed (either by offloading to a MEC server or by computing locally), transmission rate R^i between user and MEC server measured in bits per second, and finally a local processing rate f^i in CPU cycles per second. Each user i may make a decision $X_i \in \{0, 1\}$ of whether to offload their task.

For a given overall MEC server processing rate $f_{\text{pool}} \in \mathbb{R}$, we assume each offloading UE is allocated a proportional amount of processing power from the MEC processing pool to their offloaded task's complexity. The time T_i^{tx} to transmit the i-th users' task to a MEC server and the time T_i^{off} to compute at the MEC server are given by

$$T_i^{\text{tx}} = \frac{W^i}{R^i}, \quad T_i^{\text{off}} = \frac{L^i}{f_{\text{pool}} \cdot \frac{1}{\sum_j X_j}} \tag{2}$$

assuming that each offloading task is assigned equal processing power. Alternatively, one could also easily consider completion of all offloaded tasks at once, i.e. $T_i^{\text{off}} = \frac{X_j \sum_j L^j}{f_{\text{pool}}}$. The time to compute a task locally is given by

$$T_i^{\text{loc}} = \frac{L^i}{f^i}. \tag{3}$$

The result of the computed tasks are assumed to be negligible in size compared to original task size W^i, therefore, the time needed for the reception of results is not considered.

A. Competitive game setting

In the competitive, selfish setting, each user i independently decides on whether to offload or not via the random variable $X_i \in \{0, 1\}$ so as to minimize only their own expected computation time, i.e. either time to compute locally or offload

$$\mathbb{P}(X_i = 1) \cdot (T_i^{\text{tx}} + T_i^{\text{off}}) + \mathbb{P}(X_i = 0) \cdot T_i^{\text{loc}}$$
$$= \mathbb{E}\left[X_i(T_i^{\text{tx}} + T_i^{\text{off}}) + (1 - X_i)T_i^{\text{loc}} \right] \tag{4}$$

Under full information, we obtain a standard static game with the classical solution concept of mixed Nash equilibria: Each of the users chooses whether to offload according to a policy π_i which gives the conditional probability of offloading

$$\mathbb{P}(X_i = 1 \mid C^1, \dots, C^N) \equiv \pi_i(C^1, \dots, C^N) \tag{5}$$

which results in the minimization objective of each user i,

$$J_i^N(\pi_1,\ldots,\pi_N) = \mathbb{E}\left[X_i(T_i^{\text{tx}}+T_i^{\text{off}})+(1-X_i)T_i^{\text{loc}}\right]. \quad (6)$$

An approximate ε-Nash equilibrium is now defined as a tuple of policies $(\pi_1,\ldots,\pi_N)$ such that no user can gain by unilaterally changing their policy, i.e. for any $i = 1,\ldots,N$,

$$J_i^N(\pi_i,\pi_{-i}) \le \max_{\pi\in\Pi} J_i^N(\pi,\pi_{-i}) + \varepsilon \quad (7)$$

where π_{-i} denotes all policies other than the i-th policy. Here, the minimal such ε is also referred to as the exploitability of policies. An exact Nash equilibrium for $\varepsilon = 0$ is indeed guaranteed to exist as long as $\mathbb{K}$ is compact (e.g. [25]).

Unfortunately, it is known that the computation of Nash equilibria is hard, see [26]. Instead, we shall consider the many-agent case through mean-field analysis to obtain a tractable solution for large systems. At the same time, the solution will consist of decentralized policies. To this end, we shall now assume that there exists an underlying distribution $\mu_0 \in \mathscr{P}(\mathbb{K})$ of user specifications, i.e. for $i = 1,\ldots,N$ we have random variables $C^i = (W^i, L^i, f^i, R^i) \sim \mu_0$. To obtain a reasonable solution, we must also assume that the MEC pool computing power scales suitably with the number of users, i.e. $f_{\text{pool}} = N \cdot f_{\text{per}}$ for some $f_{\text{per}} \in \mathbb{R}$, since otherwise in the limit of many agents, offloading will become pointless. In practice, for fixed f_{pool} and N in given finite N-agent systems, this may be realized by defining $f_{\text{per}} := \frac{f_{\text{pool}}}{N}$.

We now consider a decentralized control setting by allowing each agent to decide whether to offload depending only on their own configuration C_i. For motivation, note that since all other agents are exchangeable from the perspective of a single agent, only the own state and overall distribution of behaviors of other agents matters. Furthermore, in the limit of $N \to \infty$, the other users' distribution is uninformative, since under a common offloading strategy, the distribution converges to some fixed mean-field by the law of large numbers. Additionally, decentralized control policies may be motivated in practice by limited agent information. Since the computation of Nash equilibria in this setting nonetheless remains hard, this motivates the mean-field formulation.

For tractability, we formulate a mean-field game as $N \to \infty$, as popularized by [3] and [4] for stochastic differential games. Here, we propose a mean-field model with near-Nash properties as N grows large, as we will also verify theoretically. Consider a policy π shared by all users. The policy induces a joint distribution $\mu = \mu_0 \otimes \pi$ over user states and offloading decisions. Under this fixed distribution μ, the objective of a single, representative user becomes

$$J^\mu(\pi) = \mathbb{E}\left[X(\tilde{T}^{\text{tx}}+\tilde{T}^{\text{off}})+(1-X)\tilde{T}^{\text{loc}}\right] \quad (8)$$

where we have expectations of random variables of the representative agent $(W, L, f, R, X) \sim \mu \otimes \pi$, and random transfer or processing times of the mean-field system W, L, f, R

$$\tilde{T}^{\text{tx}} = \frac{W}{R}, \quad \tilde{T}^{\text{off}} = \frac{L\int X\,\mathrm{d}\mu}{f_{\text{per}}}, \quad \tilde{T}^{\text{loc}} = \frac{L}{f}. \quad (9)$$

The $N \to \infty$ analogue of Nash equilibria is the mean-field equilibrium, defined as a tuple (π^*, μ^*) of policy and mean-field, such that the policy is optimal under the mean-field generated by itself, i.e. defined through the fixed point equation

$$\pi^* = \arg\min_{\pi} J^{\mu^*}(\pi), \quad (10\text{a})$$

$$\mu^* = \mu_0 \otimes \pi^*. \quad (10\text{b})$$

Analytically, for any fixed mean-field μ, we could find such a best response policy $\text{BR}(\mu)$ by defining

$$\pi^*(W,L,f,R) = \mathbf{1}_{\tilde{T}^{\text{tx}}(W,L,f,R)+\tilde{T}^{\text{off}}(W,L,f,R)<\tilde{T}^{\text{loc}}(W,L,f,R)}. \quad (11)$$

However, simply iterating the two fixed point equations is generally not guaranteed to converge to an equilibrium. Thus, we will learn equilibria through fictitious play [9].

B. Cooperative control setting

In contrast to the selfish, competitive setting, in a cooperative setting it may be of interest to minimize the average processing time of all users. One may formulate a centralized optimization problem as

$$\min_{X_1,\ldots,X_N} \quad \frac{1}{N}\sum_{i=1}^N X_i(T_i^{\text{tx}}+T_i^{\text{off}})+(1-X_i)T_i^{\text{loc}} \quad (12\text{a})$$

$$\text{subject to} \quad X_i \in \{0,1\} \quad \forall i \in \{1,\ldots,N\} \quad (12\text{b})$$

under full information. However, again this problem is known to be difficult to solve exactly for large N, as it is a knapsack problem [27]. Furthermore, we may again be interested in a decentralized solution, where each agent uses an independent policy, eliminating the need for centralized knowledge and only requiring knowledge of the local configuration C^i. Reformulating as optimization over decentralized policies π_i and optimizing over the expected cost, we have

$$\min_{\pi^1,\ldots,\pi^N} \quad \mathbb{E}\left[\frac{1}{N}\sum_{i=1}^N X_i(T_i^{\text{tx}}+T_i^{\text{off}})+(1-X_i)T_i^{\text{loc}}\right] \quad (13\text{a})$$

$$\text{subject to} \quad \pi_i \colon \mathbb{K} \to [0,1] \quad \forall i \in \{1,\ldots,N\} \quad (13\text{b})$$

where each offloading decision $X_i \sim \text{Bernoulli}(\pi_i(C_i))$ follows from the policy π_i.

As $N \to \infty$, under the policy π for all agents, we can obtain the corresponding mean-field control problem, which is more tractable than directly solving the N-user system, given as

$$\min_{\pi} \quad \int X(\tilde{T}^{\text{tx}}+\tilde{T}^{\text{off}})+(1-X)\tilde{T}^{\text{loc}}\,\mathrm{d}(\mu_0\otimes\pi) \quad (14)$$

again with the previous definitions. Note that although we impose a shared, common policy π, sharing a policy across all agents will indeed be sufficient for optimality [28].

Since the problem is now reduced to the choice of $\pi\colon \mathbb{K} \to [0,1]$, the combinatorial optimization problem has been reduced to optimization over a bounded function π with complexity independent of N. If we further assume that μ_0 has finite support, i.e. $K := |\mathbb{K}| < \infty$ and

$$\mu_0 = \sum_{j=1}^K p_j \delta_{(W_j,L_j,f_j,R_j)}, \quad \sum_{j=1}^K p_j = 1, \quad (15)$$

for some $p_j \geq 0$, $(W_j, L_j, f_j, R_j) \in \mathbb{K}$, then we obtain

$$
\begin{aligned}
&\int X(\tilde{T}^{\text{tx}} + \tilde{T}^{\text{off}}) + (1-X)\tilde{T}^{\text{loc}} \, \mathrm{d}(\mu_0 \otimes \pi) \\
&= \sum_{j=1}^{K} p_j \pi_j \left(\frac{W_j}{R_j} + \frac{L_j \sum_{k=1}^{K} p_k \pi_k}{f_{\text{per}}} \right) + p_j(1-\pi_j)\frac{L_j}{f_j} \\
&= \sum_{j=1}^{K} \sum_{k=1}^{K} \frac{p_j p_k L_j}{f_{\text{per}}} \pi_j \pi_k + \sum_{j=1}^{K} \left(\frac{p_j W_j}{R_j} - \frac{p_j L_j}{f_j} \right) \pi_j + \sum_{j=1}^{K} \frac{p_j L_j}{f_j} \\
&= \pi^T Q \pi + c^T \pi + \text{const.}
\end{aligned}
$$

for $\pi \equiv (\pi_1, \ldots, \pi_K)^T$, $\pi_j = \pi(W_j, L_j, f_j, R_j)$ and appropriate Q, c. Therefore, we obtain a non-convex quadratic program

$$
\min_{\pi_1, \ldots, \pi_K} \quad \pi^T Q \pi + c^T \pi \tag{16a}
$$

$$
\text{subject to} \quad \pi_j \in [0,1] \quad \forall j \in \{1, \ldots, K\} \tag{16b}
$$

with box constraints, which though NP-hard [29] in the cardinality of the support of μ_0, can be solved numerically. Most importantly, the complexity remains independent of N, giving us a tractable solution for sufficiently small K. To handle more general densities μ_0 with non-finite but compact support $\mathbb{K}$, we may discretize distributions and solve the resulting finite-support problem. As a result, we have obtained a tractable solution to the otherwise intractable offloading problem for many devices, as we will verify in the sequel.

III. TIME-STATIONARY MODEL

While the previous model assumes an instantaneous problem where we let all users play a one-shot game, another important and interesting setting is to assume a continuous flow of tasks arriving over time. While a theoretically rigorous analysis of this setting is beyond the scope of our work, we nonetheless consider this setting at its time-stationary equilibrium and solve it numerically.

At all times, let the arrival process of tasks be given by a Poisson process with constant rate λN, which is equivalent to Poisson arrival rates λ for each of N users. At equilibrium, in the limit there must be a time-stationary bandwidth per user f_{alloc} allocated to a user choosing to offload their task. This bandwidth is given by dividing the total processing power $f_{\text{pool}} = N f_{\text{per}}$ by the number of jobs in the system. Since the processing time for any offloaded job arriving at equilibrium is given by $T^{\text{tx}} = \frac{W}{R}$ and $T^{\text{off}} = \frac{L}{f_{\text{alloc}}}$, the expected number of jobs in the system as $N \to \infty$ will be given by

$$
\mathbb{E}[N_{\text{tot}}] = \lambda N \mathbb{E}\left[X(T^{\text{tx}} + T^{\text{off}}) \right] \tag{17}
$$

and is given by a sum of N Poisson variables N_{tot}^i, the numbers of jobs in the system from each user i. Therefore, by the central limit theorem, the fluctuations of N_{tot} are on the order of $O(\sqrt{N})$, resulting in the allocated processing rate per user

$$
\begin{aligned}
f_{\text{alloc}} &= \frac{f_{\text{pool}}}{\mathbb{E}[N_{\text{tot}}] + O(\sqrt{N})} = \frac{f_{\text{per}}}{\lambda \mathbb{E}[X(T^{\text{tx}} + T^{\text{off}})] + O(\frac{1}{\sqrt{N}})} \\
&\to \frac{f_{\text{per}}}{\lambda \mathbb{E}\left[\frac{XW}{R} \right] + \frac{\lambda \mathbb{E}[XL]}{f_{\text{alloc}}}}
\end{aligned} \tag{18}
$$

as $N \to \infty$, which for $f_{\text{alloc}} \neq 0$ gives

$$
f_{\text{alloc}} = \frac{f_{\text{per}} - \lambda \mathbb{E}[XL]}{\lambda \mathbb{E}\left[\frac{XW}{R} \right]} \tag{19}
$$

and the natural constraint

$$
f_{\text{per}} - \lambda \int XL \, \mathrm{d}(\mu_0 \otimes \pi) > 0. \tag{20}
$$

Intuitively, this constraint formalizes the notion of sufficient MEC resources, i.e. the rate of assigned jobs times their complexity must not exceed the possible compute assigned per node, as otherwise the MEC servers will be unable to catch up with assigned tasks, resulting in no time-stationary solution. Note that this constraint is trivially fulfilled if

$$
f_{\text{per}} > \lambda \mathbb{E}[L]. \tag{21}
$$

Optimizing the average waiting times of all agents in the cooperative case gives the MFC problem

$$
\min_{\pi \in [0,1]^{\mathbb{K}}} \mathbb{E}\left[X(T^{\text{tx}} + T^{\text{off}}) + (1-X)\tilde{T}^{\text{loc}} \right] \tag{22}
$$

where for finite $\mathbb{K}$ we have

$$
\begin{aligned}
&\mathbb{E}\left[X(T^{\text{tx}} + T^{\text{off}}) + (1-X)\tilde{T}^{\text{loc}} \right] \\
&= \mathbb{E}\left[\frac{XW}{R} + \frac{XL\lambda \mathbb{E}\left[\frac{XW}{R}\right]}{f_{\text{per}} - \lambda \mathbb{E}[XL]} + (1-X)\frac{L}{f} \right] \\
&= \sum_{j=1}^{K} p_j \pi_j \left(\frac{W_j}{R_j} + \frac{L_j \lambda \sum_{k=1}^{K} \frac{p_k \pi_k W_k}{R_k}}{f_{\text{per}} - \lambda \sum_{k=1}^{K} p_k \pi_k L_k} \right) + p_j(1-\pi_j)\frac{L_j}{f_j}.
\end{aligned}
$$

For the competitive MFG, we can analogously define an equilibrium as any fixed point policy π^* such that

$$
\pi^* \in \arg\max_{\pi} \mathbb{E}_{\pi}\left[\frac{XW}{R} + \frac{XL\lambda \mathbb{E}_{\pi^*}\left[\frac{XW}{R}\right]}{f_{\text{per}} - \lambda \mathbb{E}_{\pi^*}[XL]} + (1-X)\frac{L}{f} \right]. \tag{23}
$$

IV. THEORETICAL GUARANTEES

In this section, we state a number of theoretical guarantees for the one-shot mean-field problems. Extensions to the time-stationary case are deferred to future work. The results follow from formulating the problem as certain standard mean-field game and mean-field control problems and applying existing results. In particular, note that our systems can be reformulated as standard mean-field games with action space $\{0,1\}$ and state space $\mathbb{K} \cup (\mathbb{K} \times \mathbb{U})$, see also [7], [28]. For the competitive setting, as $N \to \infty$, the MFG equilibrium exists and will constitute an approximate Nash equilibrium.

Theorem 1: A solution (π^*, μ^*) of the MFG problem (10) exists, and π^* constitutes an ε_N-Nash equilibrium of the finite N-user system with $\varepsilon_N \to 0$ as $N \to \infty$.

Proof: See [7, Theorem 4.1]. ∎

Furthermore, it is known that the fictitious play algorithm will converge in terms of exploitability, giving us the desired approximate Nash equilibrium.

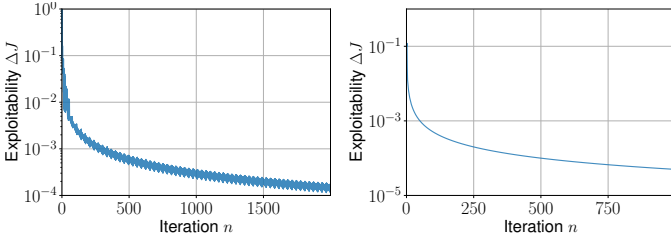*Theorem 2:* The exploitability of the solution of the fictitious play algorithm converges to zero.

Fig. 2. Learning curve for the exploitability ΔJ in the competitive MFG problem (10) (left) over 5000 iterations n using fictitious play. The fictitious play algorithm quickly converges to the equilibrium $\pi^* \approx (1, 0.65, 0)$. Here, we used $(p_1, p_2, p_3) = (0.2, 0.4, 0.4)$, $f_{per} = 0.5$ and $(C_i)_{i=1,2,3} = ((1,1,1,20),(3,2,1,20),(5,3,1,20))$. Similar results are achieved in the time-stationary MFG problem (23) (right), converging to the equilibrium $\pi^* \approx (1, 0.65, 0)$ for $(p_1, p_2, p_3) = (0.2, 0.4, 0.4)$, $f_{per} = 0.5$, $\lambda \approx 0.225$ and $(C_i)_{i=1,2,3} = ((1,1,5,10),(3,2,5,10),(5,3,5,10))$.

Proof: The system fulfills [9, Assumption 1] and in particular the monotonicity property, since the offloading cost only increases when more agents offload. Therefore, by [9, Corollary 8.2], we have convergence of the fictitious play algorithm to the unique mean-field equilibrium. ∎

Similarly, the cooperative MFC solution has an optimal solution, which will constitute an approximate Pareto optimum in the finite user system.

Theorem 3: For distributions μ_0 with finite support, an optimizer π^* of (16) exists, and π^* constitutes an ε_N-Pareto optimum of the finite N-user system with $\varepsilon_N \to 0$ as $N \to \infty$.

Proof: Existence is trivially guaranteed by the extreme value theorem, since the objective is a continuous function of $\pi \in [0,1]^K$, and $[0,1]^K$ is compact. For approximate Pareto-optimality, see [28, Corollary 1]. ∎

V. NUMERICAL SIMULATION

In this section, we present numerical simulations for the systems established in the prequel. For the quadratic program MFC, we could apply convex quadratic program solvers if the problem is indeed convex. However, in general the MFC problem may be non-convex and thus results in a NP-hard problem [29]. Still, we again stress that the complexity scales only with the size of $\mathbb{K}$ and remains independent of the
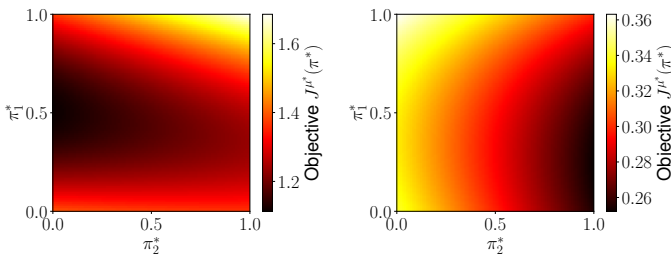

Fig. 3. Exemplary 2D-case for the quadratic MFC problem (16) (left) and $K = 2$, reaching the optimal objective of around 1.26 at around $\pi^* \approx (0.52, 0)$. Here, we used $(p_1, p_2) = (0.8, 0.2)$, $f_{per} = 3$ and $(C_i)_{i=1,2} = ((3,5,3,10),(1.5,1.5,5,25))$. Similar results are achieved for the time-stationary MFC problem (22) (right), where we achieve the optimal value 0.25 at $\pi^* \approx (0.24, 1)$ for $(p_1, p_2) = (0.8, 0.2)$, $f_{per} = 3$, $\lambda = 0.6$ and $(C_i)_{i=1,2} = ((3,1.5,5,12),(1.5,1,2,20))$.
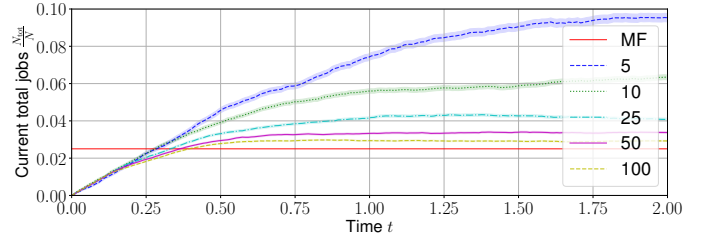

Fig. 4. The evolution of the expected total number of jobs in the queue divided by N with 68% confidence interval, plotted for the configuration from Fig. 2 and various N from 5 to 100, compared against the stationary mean-field solution (MF). We average over 5000 sample trajectories. As we consider increasingly large systems, the expected rescaled number of jobs in the system converges to the limiting mean-field description, letting us conclude that the limiting mean-field system is a good approximation for the finite user system.
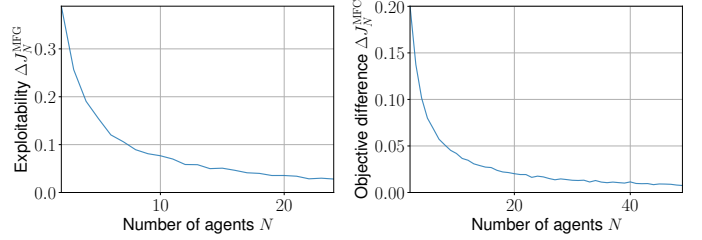

Fig. 5. Comparison of N-agent exploitability in the MFG (10) (left), and N-agent objective deviation from the limiting objective in the MFC (16) (right) for the configurations used in Figures 2 and 3. The exploitability quickly decreases to zero, and similarly the cooperative problem is quickly well-approximated by the MFC.

number of users N. Therefore, our formulation will be of lower complexity than solving the finite user model for large systems. For space reasons, we do not compare run times, since finite model solvers will trivially exceed the run time of our solution for sufficiently large systems. We may follow any global optimization algorithm, and for simplicity we apply a simple grid search, though more sophisticated algorithms such as Bayesian optimization can easily be substituted.

As can be observed in Fig. 2, for the competitive MFG problem (10), the fictitious play algorithm

$$\pi_{n+1} \equiv \frac{1}{n+1}\left(n\pi_{1:n} + \arg\min_{\pi} J^{\mu_0 \otimes \pi_{1:n}}(\pi)\right) \quad (24)$$

with the past average policy $\pi_{1:n} := \frac{1}{n}\sum_{m=1}^{n} \pi_m$ quickly converges in terms of the exploitability

$$\Delta J(\pi) := \max_{\pi^*} J^{\mu_0 \otimes \pi}(\pi^*) - J^{\mu_0 \otimes \pi}(\pi) \quad (25)$$

which must be equal to zero for an exact equilibrium. For the parameters given in Fig. 2, the resulting equilibrium $\pi^* \approx (0, 0.81, 1)$ is intuitive: Only the second configuration splits between offloading and local computation at a ratio that equilibrates offloading and local computation time, since the second configuration has longer offloading times than the first, and longer local computation time than the third. The result are offloading decisions where each UE gains little by deviating.

In Fig. 3, we can observe the cost function for an illustrative case where $K = 2$. As can be seen in the example, the

optimum computational offloading policy is reached at around $\pi^* \approx (0.52, 0)$. Similarly, a solution can be reached for the time-stationary problem at around $\pi^* \approx (0.24, 1)$. Here, we solve the problem for an illustrative 3D example in a few seconds, though similar results can easily be obtained for larger problems. Thus, we obtained nearly optimal offloading decisions, minimizing the average computation times.

In Fig. 4, we can observe that the time-homogeneous problem empirically shows a number of jobs in the system that converges to the mean-field description when rescaled by N, leading us to the conclusion that the mean-field model we proposed is a good approximation to the finite user system as long as the system is sufficiently large.

Finally, in Fig. 5, we can observe (i) the exploitability in the competitive finite user system, i.e. the expected maximum gain by deviating to any other policy in (4), and (ii) the deviation of the objective (13) from the computed mean-field objective (16) in the cooperative setting. Here we estimated the exploitability for each value of N by taking the maximum over all pure policies $\pi \in \{0, 1\}^{\mathbb{K}}$ over $100\,000$ samples. Similarly, we estimated the deviation between (13) and (16) over $20\,000$ samples of the finite user system. We observe that the exploitability and deviation of objectives tends to zero as the number of agents increases, showing that the mean-field solution solves the finite system well.

VI. CONCLUSION

In this work, we have shown the general applicability of rigorous mean-field frameworks for both competitive and cooperative scenarios in offloading for edge-computing. In particular, we have shown that the mean-field approximation quickly becomes a good approximation and can reliably be solved with a complexity independent of the number of agents. As a result, we have obtained good and tractable solutions for large-scale, decentralized edge-computing systems. In future work, one could extend rigorous theoretical analysis to the time-stationary case. Other interesting directions could be an extension to Markov-modulated task arrival rates and thereby a non-time-stationary case, or an even more distributed setting with multiple separate limited-access MEC pools. Finally, an application to real systems may be of interest.

REFERENCES

[1] Cisco, "Cisco annual internet report (2018–2023)," Tech. Rep. C11-741490-01, 2020.
[2] K. Zhang, Z. Yang, and T. Başar, "Multi-agent reinforcement learning: A selective overview of theories and algorithms," in *Handbook of Reinforcement Learning and Control*, K. G. Vamvoudakis, Y. Wan, F. L. Lewis, and D. Cansever, Eds. Cham: Springer International Publishing, 2021, pp. 321–384.
[3] M. Huang, R. P. Malhamé, P. E. Caines *et al.*, "Large population stochastic dynamic games: closed-loop McKean-Vlasov systems and the Nash certainty equivalence principle," *Commun. Inf. Syst.*, vol. 6, no. 3, pp. 221–252, 2006.
[4] J.-M. Lasry and P.-L. Lions, "Mean field games," *Japanese J. Math.*, vol. 2, no. 1, pp. 229–260, 2007.
[5] O. Guéant, J.-M. Lasry, and P.-L. Lions, "Mean field games and applications," in *Paris-Princeton lectures on mathematical finance 2010*. Springer, 2011, pp. 205–266.
[6] D. A. Gomes, J. Mohr, and R. R. Souza, "Discrete time, finite state space mean field games," *Journal de mathématiques pures et appliquées*, vol. 93, no. 3, pp. 308–328, 2010.
[7] N. Saldi, T. Basar, and M. Raginsky, "Markov–Nash equilibria in mean-field games with discounted cost," *SIAM J. Control Optim.*, vol. 56, no. 6, pp. 4256–4287, 2018.
[8] X. Guo, A. Hu, R. Xu, and J. Zhang, "Learning mean-field games," in *Proc. NeurIPS*, 2019, pp. 4966–4976.
[9] R. Elie, J. Perolat, M. Laurière, M. Geist, and O. Pietquin, "On the convergence of model free learning in mean field games," in *Proc. AAAI*, vol. 34, no. 05, 2020, pp. 7143–7150.
[10] D. Andersson and B. Djehiche, "A maximum principle for SDEs of mean-field type," *Appl. Math. Optim.*, vol. 63, no. 3, pp. 341–356, 2011.
[11] A. Bensoussan, J. Frehse, P. Yam *et al.*, *Mean field games and mean field type control theory*. Springer, 2013, vol. 101.
[12] A. C. Kizilkale and R. P. Malhame, "Collective target tracking mean field control for electric space heaters," in *Proc. Mediterranean Conf. Control Autom.* IEEE, 2014, pp. 829–834.
[13] B. Djehiche and H. Tembine, "Risk-sensitive mean-field type control under partial observation," in *Stochastics of environmental and financial economics*. Springer, Cham, 2016, pp. 243–263.
[14] W. Lee, S. Liu, H. Tembine, W. Li, and S. Osher, "Controlling propagation of epidemics via mean-field control," *SIAM J. Appl. Math.*, vol. 81, no. 1, pp. 190–207, 2021.
[15] J.-P. Fouque and Z. Zhang, "Deep learning methods for mean field control problems with delay," *Front. Appl. Math. Stat.*, vol. 6, p. 11, 2020.
[16] R. Zheng, H. Wang, and M. De Mari, "Optimal computation offloading with a shared MEC center: A mean field game approach," in *Proc. GLOBECOM Workshop.* IEEE, 2019, pp. 1–6.
[17] R. A. Banez, L. Li, C. Yang, L. Song, and Z. Han, "A mean-field-type game approach to computation offloading in mobile edge computing networks," in *Proc. IEEE ICC.* IEEE, 2019, pp. 1–6.
[18] R. A. Banez, H. Tembine, L. Li, C. Yang, L. Song, Z. Han, and H. V. Poor, "Mean-field-type game-based computation offloading in multi-access edge computing networks," *IEEE Trans. Wirel. Commun.*, vol. 19, no. 12, pp. 8366–8381, 2020.
[19] X. Wang, J. Ye, and J. C. Lui, "Joint D2D collaboration and task offloading for edge computing: A mean field graph approach," in *Proc. IEEE/ACM 29th Int. Symp. Qual. Serv. (IWQOS).* IEEE, 2021, pp. 1–10.
[20] A. F. Hanif, H. Tembine, M. Assaad, and D. Zeghlache, "Mean-field games for resource sharing in cloud-based networks," *IEEE/ACM Trans. Networking*, vol. 24, no. 1, pp. 624–637, 2015.
[21] H. Gao, W. Li, R. A. Banez, Z. Han, and H. V. Poor, "Mean field evolutionary dynamics in dense-user multi-access edge computing systems," *IEEE Trans. Wirel. Commun.*, vol. 19, no. 12, pp. 7825–7835, 2020.
[22] R. Zheng, H. Wang, M. De Mari, M. Cui, X. Chu, and T. Q. Quek, "Dynamic computation offloading in ultra-dense networks based on mean field games," *IEEE Trans. Wirel. Commun.*, vol. 20, no. 10, pp. 6551–6565, 2021.
[23] L. Li, Q. Cheng, X. Tang, T. Bai, W. Chen, Z. Ding, and Z. Han, "Resource allocation for NOMA-MEC systems in ultra-dense networks: A learning aided mean-field game approach," *IEEE Trans. Wirel. Commun.*, vol. 20, no. 3, pp. 1487–1500, 2020.
[24] R. A. Banez, L. Li, C. Yang, and Z. Han, "A survey of mean field game applications in wireless networks," in *Mean Field Game and its Applications in Wireless Networks*. Springer, 2021, pp. 61–82.
[25] D. Fudenberg and J. Tirole, *Game theory*. MIT press, 1991.
[26] C. Daskalakis, P. W. Goldberg, and C. H. Papadimitriou, "The complexity of computing a Nash equilibrium," *SIAM J. Comput.*, vol. 39, no. 1, pp. 195–259, 2009.
[27] B. Yılmaz, A. Ortiz, and A. Klein, "Delay minimization for edge computing with dynamic server computing capacity: A learning approach," in *Proc. GLOBECOM*, 2020, pp. 1–6.
[28] K. Cui, A. Tahir, M. Sinzger, and H. Koeppl, "Discrete-time mean field control with environment states," in *Proc. IEEE CDC*, 2021, pp. 5239–5246.
[29] P. M. Pardalos and S. A. Vavasis, "Quadratic programming with one negative eigenvalue is NP-hard," *J. Global Optim.*, vol. 1, no. 1, pp. 15–22, 1991.