Mean-Field-Type Game-Based Computation Offloading in Multi-Access Edge Computing Networks

Reginald A. Banez, Hamidou Tembine, *Senior Member, IEEE*, Lixin Li, *Member, IEEE*, Chungang Yang, *Senior Member, IEEE*, Lingyang Song, *Fellow, IEEE*, Zhu Han, *Fellow, IEEE*, and H. Vincent Poor, *Life Fellow, IEEE*

***Abstract*— Multi-access edge computing (MEC) has been proposed to reduce latency inherent in traditional cloud computing. One of the services offered in an MEC network (MECN) is computation offloading in which computing nodes, with limited capabilities and performance, can offload computation-intensive tasks to other computing nodes in the network. Recently, mean-field-type game (MFTG) has been applied in engineering applications in which the number of decision makers is finite and where a decision maker can be distinguishable from other decision makers and have a non-negligible effect on the total utility of the network. Since MECNs are implemented through finite number of computing nodes and the computing capability of a computing node can affect the state (i.e., the number of computation tasks) of the network, we propose non-cooperative and cooperative MFTG approaches to formulate computation offloading problems. In these scenarios, the goal of each computing node is to offload a portion of the aggregate computation tasks from the network that minimizes a specific cost. Then, we utilize a direct approach to calculate the optimal solution of these MFTG problems that minimizes the corresponding cost. Finally, we conclude the paper with simulations to show the significance of the approach.**

***Index Terms*— Mean-field-type games, computation offloading, multi-access edge computing networks.**

Manuscript received June 12, 2019; revised November 16, 2019, March 8, 2020, and June 22, 2020; accepted August 23, 2020. Date of publication September 14, 2020; date of current version December 10, 2020. This work was partially supported by US Multidisciplinary University Research Initiative 18RT0073, NSF EARS-1839818, CNS-1717454, CNS-1731424, and CNS-1702850. The associate editor coordinating the review of this article and approving it for publication was K. Kansanen. *(Corresponding author: Reginald A. Banez.)*

Reginald A. Banez is with the Department of Electrical and Computer Engineering, University of Houston, Houston, TX 77004 USA (e-mail: rabanez@uh.edu).

Hamidou Tembine is with the Department of Electrical and Computer Engineering, New York University, Abu Dhabi, UAE (e-mail: ht36@nyu.edu).

Lixin Li is with the School of Electronics and Information, Northwestern Polytechnical University, Xi'an 710129, China (e-mail: lilixin@nwpu.edu.cn).

Chungang Yang is with the State Key Laboratory on Integrated Services Networks, Xidian University, Xi'an 710071, China (e-mail: chgyang2010@163.com).

Lingyang Song is with the School of Electrical Engineering and Computer Science, Peking University, Beijing 100871, China (e-mail: lingyang.song@pku.edu.cn).

Zhu Han is with the Department of Electrical and Computer Engineering, University of Houston, Houston, TX 77004 USA, and also with the Department of Computer Science and Engineering, Kyung Hee University, Seoul 02447, South Korea (e-mail: hanzhu22@gmail.com).

H. Vincent Poor is with the Department of Electrical Engineering, Princeton University, Princeton, NJ 08544 USA (e-mail: poor@princeton.edu).

Color versions of one or more of the figures in this article are available online at https://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TWC.2020.3021907

I. Introduction

MOBILE cloud computing (MCC) offers cloud services such as computing, caching, and communications to mobile end users. These services are processed in the cloud that might be geographically located far from the end users. Consequently, MCC suffers from high latency that is not acceptable in some applications. To alleviate this problem, multi-access edge computing (MEC) has been proposed in which the cloud services are provided at the edge network located in proximity with the end users. Therefore, the move of computing services from the cloud to the edge network effectively reduces latency. Aside from low latency, other benefits of MEC include proximity, high bandwidth, real-time radio network information, and location awareness [1]. MEC can be implemented through a network of computing nodes distributed over a geographic area. These computing nodes form the multi-access edge computing network (MECN) that provides computing services to the end users.

Computation offloading is one of the main services provided by an MECN where an end user equipment, such as a smartphone, can offload computation-intensive tasks, or portions of it, to the MECN instead of performing the task locally. The decision to offload depends on factors such as latency, bandwidth, and energy consumption of the equipment. In the literature, computation offloading has been formulated as a game theoretic problem or an optimization problem with the goal of minimizing the cost incurred by a mobile device or the network subject to constraints such as computing power, latency, and bandwidth. Many works in game theory focus on modeling and optimizing wireless and communication networks of competing and/or cooperating network entities [2]. Furthermore, game theory has been used to improve the performance of wireless networks through efficient and effective allocation of network resources [3].

Meanwhile, mean field games (MFGs), introduced by Lasry and Lions in [4], have been applied in many applications in economics and engineering. MFGs model the interaction of a decision maker with the collective behavior of other decision makers in the game. In order to apply MFG, the assumptions made in the game include large number of decision makers, anonymity and non-atomicity of decision makers [5]. Recent applications of MFG in engineering are power control in D2D networks [6], electric vehicle competition in smart grids [7], security enhancements in mobile ad-hoc networks [8], and power allocation in full-duplex ultra-dense cellular networks [9]. However, there is a relaxed version of MFG, called mean-field-type game (MFTG), in which the assumptions made in MFG do not necessarily have to hold.

In this paper, the idea of computation offloading is extended to offloading among computing nodes [10]. Specifically, an MECN aggregates the computation tasks from the end users and then it offloads portions of the aggregated tasks to the computing nodes. Since an MECN can be implemented through finite number of computing nodes where a computing node can have a significant effect on the utility of the network, we propose non-cooperative and cooperative MFTG approaches to computation offloading in MECN. In these approaches, the goal of each computing node is to minimize cost by controlling its own offloading strategy subject to the state dynamics of the MECN. Each computing node does not need to know the offloading strategy of other computing nodes in order to determine its own offloading strategy. Instead, a computing node only needs to know the mean field terms that correspond to the aggregate effects of other computing nodes to the network.

The contributions of this work are summarized as follows:

- We propose and formulate computation offloading as a non-cooperative MFTG problem in which each computing node minimizes its own cost function subject to the state dynamics of the network. In the non-cooperative approach, the computing nodes operate in a decentralized manner where each computing node can compute its own computation offloading strategy without full knowledge of the strategies of other computing nodes.
- We propose and formulate a cooperative MFTG problem of computation offloading where the computing nodes jointly minimize a global cost function subject to the state dynamics of the network. In the cooperative approach, the computing nodes operate in a centralized manner where the network determines the offloading strategy of each computing node that minimizes the cost incurred by the edge network.
- We solve for the optimal computation offloading control profile that minimizes the cost in each case using a direct approach proposed in the literature. This approach does not require solving coupled partial differential equations which can be challenging. Instead, it involves calculation of mean-field terms that represent the behavior of the entire network.
- We design and propose the non-cooperative and cooperative MFTG computation offloading algorithms based on the direct approach of solving MFTG computation offloading problems. The non-cooperative algorithm is implemented in a decentralized manner while the cooperative algorithm is implemented in a centralized manner.
- We provide simulations that demonstrate the effectiveness of the proposed MFTG-based algorithms as well as the computation offloading behavior of a computing node under varying conditions. Moreover, we compare the proposed non-cooperative and cooperative MFTG computation offloading algorithms with typical computation offloading algorithms.

This work is organized as follows. Section II provides a brief survey of works in computation offloading. In Section III, the system model for computation offloading in an MECN is presented. Next, computation offloading is formulated as non-cooperative and cooperative MFTGs in Section IV. Then, these MFTG problems are solved using a direct approach in Section V. In Sections VI and VII, the proposed algorithms for MFTG-based computation offloading and the performance evaluation metrics are presented, respectively. Section VIII provides simulations that demonstrate the results of utilizing MFTG in computation offloading. Finally, the paper is concluded in Section IX.

II. Related Works

Offloading of computation-intensive tasks from mobile devices to MECNs has garnered a lot of interests in the research community. In this section, we briefly describe some these computation offloading methods.

Various game theoretic methods have been applied to model computation offloading among many computing units. In [11], the authors utilized a game theoretic approach to computation offloading problem among mobile users in a multi-user, multi-channel wireless MECN. Meanwhile, in order to utilize the computation resources in the cloud, collaborative computation offloading between the centralized cloud server and the MEC servers was studied in [12].

Many research have jointly optimized computation offloading with other network technologies and issues. The authors of [13] formulated computation offloading among mobile devices as a joint optimization of the radio and computation resources that minimizes a user's energy consumption while satisfying latency requirements. An energy-efficient dynamic offloading and resource scheduling formulated as a minimization problem was investigated in [20]. In [21], interference management was integrated in computation offloading and formulated together as an optimization problem. To reduce execution delay, computation offloading was integrated with cache placement in MEC to store and share popular computation results to mobile users [22].

Several works have focused on integrating computation offloading feature in networks involving wireless power. Computation offloading in mobile cloud computing powered by wireless energy transfer was studied in [19]. The authors proposed the use of CPU-cycle statistics information and channel state information to enforce policies that maximize the probability of successful computation of data subject to the energy harvesting and latency constraints. In [23], the authors combined the concepts of MEC with wireless power transfer so that the MEC access point can transmit wireless power to mobile users which can be used for local computing. Then, the authors of [14] proposed a Lyapunov optimization-based dynamic algorithm for MEC with energy-harvesting devices that jointly decides on the offloading, CPU frequency, and transmit power.

Energy-efficient computation offloading algorithms have been the focus of several works as well. An energy-efficient computation offloading scheme was proposed in [15] where the energy consumption of the offloading system was minimized while still satisfying the latency requirements of the tasks. Meanwhile, energy-efficient task offloading in software defined ultra-dense network was investigated in [16].

Partial offloading where only a part of an application is offloaded to computing entities has been studied by several works. The authors of [17] considered partial offloading due to limited bandwidth in wireless networks. Also, the authors of [18] proposed a cooperative partial computation offloading between cloud computing and MEC-enabled IoT.

Before concluding this section, we briefly discuss some works involving MFTGs. In [24], energy storage problem in a microgrid was formulated as an MFTG. The mean and variance of the energy level were added to the cost function and used MFTG to keep track and maintain the desired energy level in the microgrid. Meanwhile, MFTG was utilized as a particle filter for video-based vehicular tracking in Intelligent Traffic Systems (ITS) [25]. A mean field term was included in the formulation to provide accurate and robust state (i.e., vehicle position) prediction. In [26], MFTG was applied in blockchain token economics. This work introduced variance in the utility function to capture the risk of cryptographic tokens associated with the uncertainties of technology adoption, network security, regulatory legislation, and market volatility.

The main difference of this work is that computation offloading in MECN has been formulated as an MFTG in which each computing node has a desired level of computation tasks it can handle. This level is dictated by the energy consumption and computing capability of the computing node. Moreover, this work utilizes a direct approach that does not require solving coupled partial differential equations to solve for the optimal computation offloading strategy of each computing node. Lastly, we have considered both non-cooperative and cooperative scenarios among the computing nodes.

III. System Model

Fig. 1 shows the system model proposed in this paper. The end user devices (EUs) offload computation-intensive tasks that cannot be performed locally to the task aggregator (TA) in the area or cell the EU is located. Each EU decides to offload based on algorithms presented in the literature such as in [27]. Then, the TA combines all the computation tasks submitted by the EUs in the area. It organizes the computation tasks to reduce redundancy and overloading of computation tasks. Moreover, it performs a portion of the tasks and directly sends the results to the corresponding EUs. Afterwards, the TA offloads parts of the remaining aggregate computation tasks to the edge computing nodes (ECNs) in the cell. Each ECN is capable of performing computation-intensive tasks and is more powerful than a typical mobile EU equipment. For instance, a typical ECN has a computing power of about 10,000 to

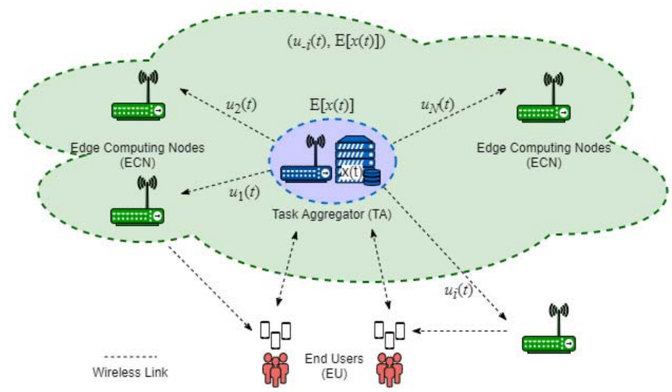


Fig. 1. Proposed system model for computation offloading in MECNs.

100,000 times that of a mobile phone [28]. After the ECNs perform their respective offloaded computation tasks, they transmit the results to the requesting EUs.

Consider a cell of an MECN consisting of one TA and a finite set $\mathcal{N}$ of ECNs with $|\mathcal{N}| = N$. The time horizon defined as $t \in [0, T]$ is finite, where $T > 0$ is the terminal time. Let the network state $x(t)$ be the number of aggregate computation tasks to be offloaded by the TA to the ECNs at time t. In addition, denote the network state dynamics $x'(t) = dx(t)/dt$ as the change or evolution of the number of aggregate computation tasks with respect to time. Also, let the admissible computation offloading control $u_i(t)$ be the portion of $x(t)$ offloaded by ECN i from the TA at time t. The goal of each ECN $i \in \mathcal{N}$ is to determine its optimal control $u_i^*(t)$ that minimizes its cost, defined by a cost function J_i, subject to network state dynamics $x'(t)$.

In the following subsections, the cost function and network state dynamics equation are presented. Important parameters that influence the optimal control of an ECN are also introduced.

A. Cost Functions

In this work, we assume that the ECNs follow a quadratic cost function because of its desirable economic properties such as monotonicity, concavity, and non-decreasing [29].

Let the consumed energy of ECN i per CPU cycle be $\varepsilon_i = \kappa_{e,i} f_i^2$ [30], where $\kappa_{e,i}$ is a constant depending on the architecture of the CPU of ECN i and f_i is the computing capability (i.e., the number of CPU cycles per unit time) of ECN i. To calculate the cost associated with the energy consumed by a certain number of computation tasks, we define the energy cost coefficient e_i as the cost (per unit time) of energy spent by computing node i per squared number of CPU cycles,

$$e_i = w_{e,i} \varepsilon_i^2 = w_{e,i} (\kappa_{e,i} f_i^2)^2, \tag{1}$$

where the constant $w_{e,i}$ is the weight or significance assigned to energy consumption cost. A higher value of $w_{e,i}$ means that an ECN prioritizes minimizing its energy consumption.

Meanwhile, to calculate the cost corresponding to the execution or computation time of a certain number of computation tasks, we define the computation time cost coefficient τ_i as the cost (per unit time) associated with execution time spent by ECN i per squared number of CPU cycles,

$$\tau_i = \frac{w_{d,i}}{f_i^2}, \tag{2}$$

where the constant $w_{d,i}$ is the weight assigned by ECN i to the computation time cost. A higher value of $w_{d,i}$ means that ECN i prioritizes minimizing the cost from computation time.

Lastly, to quantify the cost earned by the TA from offloading computation tasks to ECN i, we define the offloading cost coefficient ρ_i as the cost (per unit time) incurred by the TA per squared number of aggregate computation task. It takes into account the cost not associated with computation by an ECN i such as processing and transmission from the TA to ECN i.

Combining these cost coefficients with the network state $x(t)$ and the control $u_i(t)$, we arrive at the following running cost function that tells how much the cost increases or decreases with time,

$$L_i(x(t), u_i(t), t) = \frac{1}{2}\left[\rho_i x^2(t) + (\tau_i + e_i)u_i^2(t)\right]. \tag{3}$$

Since the goal of computation offloading is to offload tasks from the TA to the ECNs, we want to penalize the number of computation tasks $x(t)$ that remains at the TA at terminal time T. That is, $x(T)$ is considered as a part of the cost to be minimized. Since the cost at terminal time T is proportional to the number of computation tasks $x(T)$, the terminal cost function for ECN i is stated as

$$\Phi_i(x(T), T) = \frac{1}{2}\rho_i x^2(T). \tag{4}$$

In other words, it computes the cost incurred by ECN i based on the network state $x(t)$ at $t = T$.

B. Network State Dynamics Equation

The network state dynamics $x'(t)$ refers to the evolution of the network state $x(t)$ with respect to time t. In our computation offloading system model, $x'(t)$ refers to the dynamics or change in the number of aggregate computation tasks at the TA with time. Let $q_{\text{in}}(t)$ be the incoming rate of the computation tasks to the TA. Then, the number of computation tasks $x(t)$ at the TA is related, $q_{\text{in}}(t) = r_0\ x(t)$, with r_0 defined as

$$r_0 = \frac{R_0}{C_0} = \frac{1}{C_0}\sum_{j=1}^{M} B_j \log_2(1+\gamma_j),$$
$$= \frac{1}{C_0}\sum_{j=1}^{M} B_j \left(1 + \frac{P_j g_{0,j}}{N_0 + I_j}\right), \tag{5}$$

where R_0 is the maximum incoming rate of computation task the TA handle and C_0 is the capacity or the maximum number of computation tasks the TA can store. Physically, r_0 is the frequency at which computation tasks arrive at the TA. Meanwhile, R_0 is the sum of the rates the TA receives from M EUs, B_j is the channel bandwidth for EU j, and γ_j is the signal-to-interference-plus-noise (SINR) ratio between the TA and EU j, where P_j refers to EU j transmit power, $g_{0,j}$ is the channel gain between the TA and EU j, N_0 is the background noise power, I_j is the interference noise power experience by EU j.

On the other hand, the outgoing rate $q_{\text{out}}(t)$ of computation task from the TA is affected by the computation offloading control $u_i(t)$ of ECN $i \in \mathcal{N}$. Hence, $q_{\text{out}}(t) = \sum_{i=1}^{N} r_i u_i(t)$ with

$$r_i = \frac{R_i}{C_i} = \frac{B_i \log_2(1+\gamma_i)}{C_i} = \frac{B_i}{C_i}\log_2\left(1+\frac{P_i g_{i,0}}{N_0+I_i}\right), \tag{6}$$

where R_i is the maximum outgoing rate of computation task to ECN i, C_i is the capacity or the maximum number of computation tasks ECN i can handle, and consequently, r_i is the frequency at which computation tasks arrive at ECN i. In addition, B_i is the channel bandwidth of ECN i and γ_i is the SINR between the TA and ECN i, where P_i is the transmit power of ECN i, $g_{i,0}$ is the channel gain between the TA and ECN i, N_0 is the background noise power, and I_i is the interference power experienced by ECN i.

Since the total rate of computation task $x'(t) = q_{\text{in}}(t) - q_{\text{out}}(t)$, then the network state dynamics equation can be written as

$$dx(t) = \left(r_0 x(t) - \sum_{i=1}^{N} r_i u_i(t)\right) dt, \tag{7}$$

which is similar to the state dynamics equation used in [31].

To summarize, the cost function is affected by $u_i(t)$ since the the cost depends on the number of tasks ECN i offloads from the TA. On the other hand, the state dynamic equation is affected by $r_i u_i(t)$ since the change in the number of $x(t)$ depends on the rate $r_i u_i(t)$ at which tasks are offloaded to ECN i. Moreover, the linear state dynamic equation in (7) can also represent a state dynamic equation of the form $dx(t) = f(t)\,dt$ through linearization at sampling time $t = t_n$ of the function $F(t) = \int_0^t f(s)\,ds$,

$$F(t) = F(t_n) + \frac{\partial F}{\partial t}|_{t=t_n} \cdot (t - t_n),$$

and the solution $x(t)$ of (7),

$$x(t) = x(t_n) + \frac{\partial x}{\partial t}|_{t=t_n} \cdot (t - t_n).$$

Consequently, $f(t_n) = \frac{\partial F}{\partial t}|_{t=t_n} = \frac{\partial x}{\partial t}|_{t=t_n}$. For instance, if $x(t) = x_0\ e^{g(t)}$, then $f(t_n) = \frac{\partial g}{\partial t}|_{t=t_n} \cdot x(t_n)$. If $x(t)$ follows a distribution function such as a Poisson process, then the state dynamic equation $dx(t)$ can model the transition between the sampling times.

In the next section, we extend these formulations in order to adapt an MFTG approach. The main feature of this method is the addition of mean field terms in the cost functions and the state dynamics equation so that each ECN aims to compute for an optimal control $u_i^*(t)$ that minimizes the variance of the state of the network $x(t)$ as well as the variance of its control $u_i(t)$.

IV. Mean-Field-Type Game Problem Formulation

The theory of MFG, introduced in [4], [34]–[36], has been used in a variety of applications which are formulated as games among a large number of decision makers that aim to optimize their own payoffs or cost functions subject to a state dynamic equation. The main concept behind MFG is that each decision maker determines its optimal strategy (i.e., the strategy or action that optimizes its payoff or cost function) based on an aggregate information about the states of other decision makers. In other words, a decision maker computes its optimal strategy based on a statistical distribution of the states of other decision makers (i.e., a mean-field term) instead on a full knowledge of the states of other decision makers.

According to [5], most MFG models share the following assumptions: (i) there are infinitely many decision makers, (ii) the decision makers are indistinguishable, and (iii) a decision maker has negligible effect on the global utility. However, in engineering applications, these assumptions may be difficult to prove. Consequently, a more relaxed MFTG has been proposed in the literature. In MFTG, the number of decision makers may be infinite or finite, the decision makers may not be indistinguishable, and finally, a decision maker may have a significant effect on the global utility. Applications of MFTG include distributed power networks [37], network security [38], and multilevel building evacuation [39].

In this section, we formulate computation offloading in MECN as an MFTG. First, we derive the cost functions to be minimized by an ECN. Then, we show the state dynamics which is the differential equation constraint of the minimization problem. The MFTG cost functions and state dynamic equation contain mean field terms that quantify the behavior or strategy of all the computing nodes. These terms are added to the cost function so that each computing node can minimize the variance in the network state as well as the variance in computing node control. In the last subsection, we state two MFTG computation offloading problems: a non-cooperative MFTG problem where the ECNs minimize their own cost function independently, and a cooperative MFTG problem where the ECNs minimize a single global cost function.

A. Preliminaries

The network state or state of the TA $x(t)$ refers to the number of aggregate computation tasks to be offloaded to the ECNs. An admissible computation offloading control or strategy $u_i(t)$ of ECN i refers to a portion of $x(t)$ it can offload from the TA, while the set $\mathcal{U}_i$ denotes the set of all admissible controls of ECN i. Vector $u(t) = [u_i(t)]_{i\in\mathcal{N}}$ contains the control of all the ECNs in the cell, while the vector $u_{-i}(t) = [u_i(t)]_{i\in\mathcal{N}\setminus i}$ contains the control of all ECNs in the cell except ECN i.

The following subsections present the cost functions and state dynamic equation in an MFTG setting. The main difference in the formulations to follow is the inclusion of mean field terms $\bar{x}(t) = \mathbb{E}[x(t)]$ and $\bar{u}(t) = \mathbb{E}[u(t)]$. Consequently, a tilde $\sim$ is put on top of the MFTG cost function $\tilde{J}_i$ and state dynamic function $\tilde{f}$ to differentiate them from their mean-field-free counterparts. Afterwards, the resulting MFTG-based computation offloading problems are stated.

B. Cost Functions

The total cost function $\tilde{J}_i(u)$ of ECN i consists of the running cost function $\tilde{L}_i(x, u, \bar{x}, \bar{u}, t)$, which corresponds to the accumulated cost of ECN i for performing a portion $u_i(t)$ of $x(t)$, and the terminal cost function $\tilde{\Phi}_i(x, \bar{x}, T)$, which penalizes the computing node at terminal time $t = T$ depending on how far the network state $x(t)$ is from a target state (e.g., $x(t) = 0$, when all of the aggregate computation tasks are offloaded). Mathematically,

$$\tilde{J}_i(u) = \mathbb{E}\left[\int_0^T \tilde{L}_i(x, u, \bar{x}, \bar{u}, t)\, dt + \tilde{\Phi}_i(x, \bar{x}, T)\right]. \quad (8)$$

However, the running cost $\tilde{L}_i(x, u, \bar{x}, \bar{u}, t)$ for MFTG differs from that in (3) since $\tilde{L}_i$ depends on the expected values of the network state $\bar{x}(t)$ and the control $\bar{u}$. The expected values have been included in the cost function because these values are assumed to be known, and consequently, the difference to these expected values, $x(t) - \bar{x}(t)$ and $u_i(t) - \bar{u}_i(t)$. Hence, $\mathbb{E}\big[\tilde{L}_i(x, u, \bar{x}, \bar{u}, t)\big]$ is given by (9) at the bottom of the page, where $\bar{\rho}_i$, $\bar{\tau}_i$ and $\bar{e}_i$ refer to the mean of cost coefficients defined in Section III-A.

Similarly, the terminal cost $\tilde{\Phi}_i(x, \bar{x}, T)$ depends as well on the expected value of the network state at time T so that the goal of an ECN i is to minimize var$[x(t)]$

$$\begin{aligned}\mathbb{E}\big[\tilde{\Phi}_i(x, \bar{x}, T)\big] &= \frac{1}{2}\mathbb{E}\big[\rho_i x^2(T) + \bar{\rho}_i \bar{x}^2(T)\big],\\ &= \frac{1}{2}\mathbb{E}\big[\rho_i (x(T) - \bar{x}(T))^2 + (\rho_i + \bar{\rho}_i)\bar{x}^2(T)\big],\\ &= \frac{1}{2}\big(\rho_i \mathrm{var}[x(T)] + (\rho_i + \bar{\rho}_i)\bar{x}^2(T)\big). \quad (10)\end{aligned}$$

The quadratic cost functions presented in this work refer to the penalty incurred by the network through the ECNs for executing a specific number of computation tasks. While the ECNs are constrained in terms of their local energy consumption and execution time through e_i and τ_i, respectively, the penalty allows the ECNs follow a network-wide algorithm

$$\begin{aligned}\mathbb{E}\big[\tilde{L}_i(x, u, \bar{x}, \bar{u}, t)\big] &= \frac{1}{2}\mathbb{E}\big[\rho_i x^2(t) + \bar{\rho}_i \bar{x}^2(t) + (\tau_i + e_i)u_i^2(t) + (\bar{\tau}_i + \bar{e}_i)\bar{u}_i^2(t)\big]\\ &= \frac{1}{2}\mathbb{E}\big[\rho_i (x(t) - \bar{x}(t))^2 + (\rho_i + \bar{\rho}_i)\bar{x}^2(t) + (\tau_i + e_i)(u_i(t) - \bar{u}_i(t))^2 + (\tau_i + \bar{\tau}_i + e_i + \bar{e}_i)\bar{u}_i^2(t)\big]\\ &= \frac{1}{2}\big(\rho_i \mathrm{var}[x(t)] + (\rho_i + \bar{\rho}_i)\bar{x}^2(t) + (\tau_i + e_i)\mathrm{var}[u_i(t)] + (\tau_i + \bar{\tau}_i + e_i + \bar{e}_i)\bar{u}_i^2(t)\big) \quad (9)\end{aligned}$$

that optimizes the network performance. Hence, the physical meaning of the costs refers to the penalty set by the network to the ECNs. These penalties are based on physical quantities the computing nodes spend when performing computation tasks. The number of computation tasks offloaded by each ECN is limited by these penalties. Moreover, the costs are also indicators of network performance since a low cost may indicate a low terminal cost $\tilde{\Phi}_i(x, \bar{x}, T)$ which means a low number of un-offloaded tasks remain in the TA; also, a low cost may indicate a low running cost $\tilde{L}_i(x, u, \bar{x}, \bar{u}, t)$ which means the ECNs execute the number of computation tasks that satisfy their local energy and time constraints.

C. Network State Dynamics Equation

The network state dynamics $x'(t)$ of refers to the evolution of the state $x(t)$ of the TA with respect to time t. In (7), the network state dynamics is affected by the current network state $x(t)$ and the controls $u_i(t), \forall i \in \mathcal{N}$. Like in the MFTG cost functions, the expected values $\bar{x}$ and $\bar{u}$ are included in the MFTG state dynamic equation $x'(t)$ so that the evolution of the state can be formulated in terms of the deviation of $x(t)$ and $u(t)$ from their respective expected values. Consequently, the updated network state dynamics of a computation offloading in an MFTG setting is

$$dx(t) = \tilde{f}(x, u, \bar{x}, \bar{u})\, dt + \sigma\, dW(t), \tag{11}$$

where $W(t)$ is a standard Wiener process, σ is a coefficient that captures the randomness in the state dynamics, the drift term $\tilde{f}(x, u, \bar{x}, \bar{u}, t)$ is given by

$$\tilde{f}(x, u, \bar{x}, \bar{u}, t) = r_0 x(t) + \bar{r}_0 \bar{x}(t) - \left(\sum_{i=1}^{N} r_i u_i(t) + \sum_{i=1}^{N} \bar{r}_i \bar{u}_i(t) \right), \tag{12}$$

and the coefficients are defined as $\bar{r}_0 = \mathbb{E}[\frac{R_0}{C_0}]$ and $\bar{r}_i = \mathbb{E}[\frac{R_i}{C_i}] = \mathrm{E}[\frac{B_i \log_2(1+\gamma_i)}{C_i}]$. The drift term can be written in an equivalent form

$$\tilde{f}(x, u, \bar{x}, \bar{u}, t) = r_0(x(t) - \bar{x}(t)) + (r_0 + \bar{r}_0)\bar{x}(t) - \left(\sum_{i=1}^{N} r_i (u_i(t) - \bar{u}_i(t)) + \sum_{i=1}^{N} (r_i + \bar{r}_i)\bar{u}_i(t) \right), \tag{13}$$

which expresses the network state dynamics as the sum of the mean fields $\bar{x}(t)$ and $\bar{u}_i(t)$ and the terms $x(t) - \bar{x}(t)$ and $u_i(t) - \bar{u}_i(t)$. In our computation offloading scenario, since we keep track of the mean number of computation tasks $x(t)$, then we only need to know the difference $x(t) - \bar{x}(t)$ of the current number $x(t)$ from mean number $\bar{x}(t)$. The same principle is applied with $u_i(t)$ and $\bar{u}_i(t)$.

D. Non-Cooperative Problem

Consider an MECN consisting of $N \geq 2$ ECNs. Each ECN is capable of offloading and performing computation tasks from the TA. In addition, suppose the MECN implements a non-cooperative scenario where each ECN computes its offloading strategy by minimizing its own cost function. If the cost function of ECN i is defined by (8), then in a non-cooperative setting each ECN i tries to solve the MFTG problem in (14) at the bottom of the page, where $\bar{x}(t) < +\infty$. Any control $u_i^*(t)$ that satisfies (14) is the best-response of computing node i to $(u_{-i}, \mathbb{E}[x(t)])$.

Definition 1: Any control $u_i^*(t) \in \mathcal{U}_i$ satisfying (14) *is called a risk-neutral best-response control of computing node i to the control $u_{-i} \in \Pi_{j\in\mathcal{N}}\mathcal{U}_j$ of the other computing nodes $j \neq i$.*

The set of best-response controls of computing node i is defined by $\mathcal{BR}_i : \Pi_{j\in\mathcal{N}}\mathcal{U}_j \rightarrow 2^{\mathcal{U}_i}$, where $2^{\mathcal{U}_i}$ is the set of subsets of $\mathcal{U}_i$. Using the concept of best-response control strategy, a Nash equilibrium of (14) is (u_i^*, u_{-i}^*), where every ECN i solves their best-response control u_i^*.

$$\begin{aligned} \inf_{u_i\in\mathcal{U}_i} \ \tilde{J}_i(u) = \frac{1}{2}\mathbb{E}\Bigg[&\int_0^T \left[\rho_i(x(t) - \bar{x}(t))^2 + (\rho_i + \bar{\rho}_i)\bar{x}^2(t) + (\tau_i + e_i)(u_i(t) - \bar{u}_i(t))^2 + (\tau_i + \bar{\tau}_i + e_i + \bar{e}_i)\bar{u}_i^2(t)\right] dt \\ &+ \rho_i(x(T) - \bar{x}(T))^2 + (\rho_i + \bar{\rho}_i)\bar{x}^2(T)\Bigg] \\ \text{subject to } dx(t) = &\left[r_0(x(t) - \bar{x}(t)) + (r_0 + \bar{r}_0)\bar{x}(t) - \left(\sum_{i=1}^{N} r_i(u_i(t) - \bar{u}_i(t)) + \sum_{i=1}^{N} (r_i + \bar{r}_i)\bar{u}_i(t) \right) \right] dt + \sigma(t)\, dW(t) \\ x(0) = &\ x_0 \end{aligned} \tag{14}$$

$$\begin{aligned} \inf_{u_i\in\mathcal{U}_i} \ \tilde{J}_0(u) = \frac{1}{2}\mathbb{E}\Bigg[&\sum_{i=1}^{N} \int_0^T \left[\rho_i(x(t) - \bar{x}(t))^2 + (\rho_i + \bar{\rho}_i)\bar{x}^2(t) + (\tau_i + e_i)(u_i(t) - \bar{u}_i(t))^2 \right. \\ &\left. + (\tau_i + \bar{\tau}_i + e_i + \bar{e}_i)\bar{u}_i^2(t)\right] dt + \rho_i(x(T) - \bar{x}(T))^2 + (\rho_i + \bar{\rho}_i)\bar{x}^2(T)\Bigg] \\ \text{subject to } dx(t) = &\left[r_0(x(t) - \bar{x}(t)) + (r_0 + \bar{r}_0)\bar{x}(t) - \left(\sum_{i=1}^{N} r_i(u_i(t) - \bar{u}_i(t)) + \sum_{i=1}^{N} (r_i + \bar{r}_i)\bar{u}_i(t) \right) \right] dt + \sigma(t)\, dW(t) \\ x(0) = &\ x_0 \end{aligned} \tag{16}$$

Definition 2: A Nash equilibrium of the mean-field-type game in (14) *is a control profile* (u_i^*, u_{-i}^*)*, such that for every computing node* i*,*

$$\tilde{J}_i(u_i^*, u_{-i}^*) \leq \tilde{J}_i(u_i, u_{-i}^*), \quad \forall u_i \in \mathcal{U}_i. \tag{15}$$

E. Cooperative Problem

Suppose the ECNs try to jointly minimize a single global cost function $\tilde{J}_0(u) = \mathbb{E}[\sum_{i=1}^{N} \tilde{J}_i(u)]$ where $u = (u_1, \ldots, u_N)$ is the computation offloading control profile in a cooperative setting. Then, the corresponding cooperative MFTG problem is given in (16) at the bottom of the previous page. Any control profile $u^* = (u_1^*, \ldots, u_N^*)$ that satisfies (16) is a global optimum control profile that minimizes the global cost function $\tilde{J}_0$.

The next section provides the method proposed in [40] to solve for a solution of linear-quadratic MFTGs such as (14) and (16).

V. Linear-Quadratic Mean-Field-Type Game Solution Using a Direct Method

The MFTG problems defined in (14) and (16) are called linear-quadratic MFTGs (LQ-MFTG) since the cost functional is quadratic and the state dynamics is linear with respect to the state and control. Because of their special form, the authors in [40] proposed a direct approach in computing the optimal control $u_i^*(t)$ of LQ-MFTG. The proposed method can solve an LQ-MFTG without solving coupled partial differential equations. The authors proved that the proposed direct approach to LQ-MFTG yields the same solution as an analytical approach. Based on this method, this section presents the main concepts in deriving the solution for the non-cooperative and cooperative MFTG problems introduced in the previous section. The solution $u_i^*(t)$ to each problem refers to the computation offloading control or the number of computation tasks $u_i^*(t)$ ECN i must offload from the TA in order to minimize the corresponding cost of the problem. In other words, the optimal control is the number of computation tasks to be offloaded by an ECN such that the penalty incurred by the network due to the number of executed computation tasks by the ECN and the remaining tasks at the TA are minimized.

A. Non-Cooperative Solution

The direct method for the LQ-MFTG problem starts with choosing a guess cost functional $\phi_i(x,t)$. Since the cost functional J_i is quadratic, the corresponding $\phi_i(x,t)$ is quadratic as well,

$$\phi_i(x,t) = \frac{1}{2}\alpha_i(x-\bar{x})^2 + \frac{1}{2}\beta_i\bar{x}^2 + \gamma_i\bar{x} + \delta_i,$$

where α_i, β_i, γ_i, and δ_i are restricted to time-invariant coefficients for $[0,T]$.

Then, apply the Ito's formula in (17) at the bottom of the page for a drift-diffusion process to $\phi_i(x,t)$ with $t = T$. The next step is to compute and substitute the partial derivatives $\partial_t\phi_i, \partial_x\phi_i$, and $\partial_{xx}\phi_i$ to $\phi_i(x(T),T)$ and take its expectation, $\mathbb{E}[\phi_i(x(T),T) - \phi_i(x(0),0)]$. Afterwards, the gap $\tilde{J}_i - \mathbb{E}[\phi_i(x(0),0)]$ is calculated.

Finally, the optimal control u_i^* is derived from $\min_{u_i\in\mathcal{U}_i} \tilde{J}_i(u)$ using the appropriate optimality principles. A control u_i is called a feedback control if it is a function of time t and the state $x(t)$. To compute the best-response control u_i^* of computing node i to feedback strategies $u_j, j \neq i$, complete the square of the gap $\tilde{J}_i - \mathbb{E}[\phi_i(x(0),0)]$ as shown in (18) at the bottom of the page.

Consequently, the equivalent objective functional becomes

$$\inf_{u_i\in\mathcal{U}_i} \tilde{J}_i = \frac{1}{2}\alpha_i(0)\mathrm{var}[x(0)] + \frac{1}{2}\beta_i(0)(\mathbb{E}[x(0)])^2 + \frac{1}{2}\mathbb{E}\Big[\int_0^T \sigma^2(t)\alpha_i(t)\,dt\Big]. \tag{19}$$

Using this equivalent objective functional, we arrive at the following theorem for its optimal control u_i^*.

Theorem 1: Let the cost functional $\tilde{J}_i(u)$ *of an LQ-MFTG problem take the form* $\phi_i(x,t) = \frac{1}{2}\alpha_i(x-\bar{x})^2 + \frac{1}{2}\beta_i\bar{x}^2 + \gamma_i\bar{x} + \delta_i$*, where* α_i*,* β_i*,* γ_i*, and* δ_i *are constants. Then, the optimal control* $u_i^*(t)$ *associated with the problem is given by*

$$u_i^*(t) = \frac{r_i}{\tau_i + e_i}\alpha_i(x-\bar{x}) + \frac{r_i + \bar{r}_i}{\tau_i + \bar{\tau}_i + e_i + \bar{e}_i}\beta_i\bar{x}, \tag{20}$$

where the constants α_i *and* β_i *solve the following equations, respectively,*

$$\begin{aligned} &\frac{r_i^2}{\tau_i + e_i}\alpha_i^2 + 2\Bigg(\sum_{j=1,j\neq i}^{N} \frac{r_j^2}{\tau_j + e_j}\alpha_j - r_0\Bigg)\alpha_i - \rho_i = 0,\\ &\frac{(r_i+\bar{r}_i)^2}{\tau_i + \bar{\tau}_i + e_i + \bar{e}_i}\beta_i^2 + 2\Bigg(\sum_{j=1,j\neq i}^{N} \frac{(r_j+\bar{r}_j)^2}{\tau_j + \bar{\tau}_j + e_j + \bar{e}_j}\beta_j \\ &\qquad - (r_0 + \bar{r}_0)\Bigg)\beta_i - (\rho_i + \bar{\rho}_i) = 0, \end{aligned} \tag{21}$$

and the mean field term $\bar{x}(t)$ *is given by*

$$\bar{x}(t) = \bar{x}(0)e^{\int_0^t \left((r_0+\bar{r}_0) - \sum_{i=1}^{N} \frac{\beta_i(r_i+\bar{r}_i)^2}{\tau_i+\bar{\tau}_i+e_i+\bar{e}_i}\right) ds}, \tag{22}$$

and $\bar{u}_i$ *has been expressed as* $\beta_i(r_i+\bar{r}_i)/(\tau_i+\bar{\tau}_i+e_i+\bar{e}_i)\bar{x}$.

$$\phi_i(x(T),T) = \phi_i(x(0),0) + \int_0^T \left(\partial_t\phi_i + \tilde{f}(x,u,\bar{x},\bar{u},t)\partial_x\phi_i + \frac{\sigma^2}{2}\partial_{xx}\phi_i\right)dt + \int_0^T \sigma(t)\partial_x\phi_i\,dW(t) \tag{17}$$

$$\begin{aligned} \tilde{J}_i - \mathbb{E}[\phi_i(x(0),0)] = &\frac{1}{2}\mathbb{E}\Big[\int_0^T (\tau_i + e_i)\Big(u_i - \bar{u}_i - \frac{r_i}{\tau_i + e_i}\alpha_i(x-\bar{x})\Big)^2 dt\Big] + \frac{1}{2}\mathbb{E}\Big[\int_0^T \sigma^2\alpha_i\,dt\Big] \\ &+ \frac{1}{2}\mathbb{E}\Big[\int_0^T (\tau_i + \bar{\tau}_i + e_i + \bar{e}_i)\Big(\bar{u}_i - \beta_i\frac{r_i+\bar{r}_i}{\tau_i+\bar{\tau}_i+e_i+\bar{e}_i}\bar{x}\Big)^2 dt\Big] \end{aligned} \tag{18}$$

Proof: The optimal control u_i^* is obtained by minimizing the following terms with respect to control u_i and $\bar{u}_i$,

$$\frac{\partial}{\partial u_i}\left[(\tau_i+e_i)\Big(u_i-\bar{u}_i-\frac{r_i}{\tau_i+e_i}\alpha_i(x-\bar{x})\Big)^2 + (\tau_i+\bar{\tau}_i+e_i+\bar{e}_i)\left(\bar{u}_i-\beta_i\frac{r_i+\bar{r}_i}{\tau_i+\bar{\tau}_i+e_i+\bar{e}_i}\bar{x}\right)^2\right]=0,$$

which yields $u_i=\frac{r_i}{\tau_i+e_i}\alpha_i(x-\bar{x})+\bar{u}_i$, where $\bar{u}_i=\beta_i(r_i+\bar{r}_i)/(\tau_i+\bar{\tau}_i+e_i+\bar{e}_i)\bar{x}$. Meanwhile, the mean field $\bar{x}(t)$ is derived by taking the expectation of the state dynamic equation in (14) and then solving the resulting differential equation for $\bar{x}(t)$. ∎

Theorem 1 states that the optimal number of computation tasks ECN i must offload from the TA in a non-cooperative scenario is given in (20). This number minimizes the cost incurred by ECN i where α_i and β_i satisfy the conditions (21), and the mean field $\bar{x}(t)$ satisfies (22). We can say that $u_i^*(t)$ not only depends on $x(t)$ but also on how much $x(t)$ exceeds $\bar{x}(t)$. Moreover, α_i and β_i reflect the weights of how much $u_i^*(t)$ depends on $x(t)-\bar{x}(t)$ and $\bar{x}(t)$, respectively.

B. Cooperative Solution

To obtain the global optimum solution to the cooperative LQ-MFTG problem in (16), we follow the procedure stated in the previous subsection. Hence, the LQ-MFTG problem in (16) is equivalent to

$$\inf_{u_1,\dots,u_N}\tilde{J}_0=\frac{1}{2}\alpha_0(0)\mathrm{var}[x(0)]+\frac{1}{2}\beta_0(0)(\mathbb{E}[x(0)])^2+\frac{1}{2}\mathbb{E}\Big[\int_0^T\sigma^2(t)\alpha_0(t)\,dt\Big]. \quad (23)$$

The corresponding optimal control $u_i^*(t)$ is given by the following theorem.

Theorem 2: Let the cost functional $\tilde{J}_0(u)$ of an LQ-MFTG problem take the form $\phi_0(x,t)=\frac{1}{2}\alpha_0(x-\bar{x})^2+\frac{1}{2}\beta_0\bar{x}^2$, where α_0 and β_0, are constants. Then, the optimal control $u_i^*(t)$ associated with the problem is given by

$$u_i^*(t)=\frac{r_i}{\tau_i+e_i}\alpha_0(x-\bar{x})+\frac{r_i+\bar{r}_i}{\tau_i+\bar{\tau}_i+e_i+\bar{e}_i}\beta_0\bar{x}, \quad (24)$$

where the constants α_0 and β_0 solve the following equations, respectively,

$$\left(\sum_{i=1}^N\frac{r_i^2}{\tau_i+e_i}\right)\alpha_0^2-2r_0\alpha_0-\rho_0=0,$$
$$\left(\sum_{i=1}^N\frac{(r_i+\bar{r}_i)^2}{\tau_i+\bar{\tau}_i+e_i+\bar{e}_i}\right)\beta_0^2-2(r_0+\bar{r}_0)\beta_0-(\rho_0+\bar{\rho}_0)=0, \quad (25)$$

and the mean field term $\bar{x}(t)$ is given by

$$\bar{x}(t)=\bar{x}(0)e^{\int_0^t\left((r_0+\bar{r}_0)-\beta_0\sum_{i=1}^N\frac{(r_i+\bar{r}_i)^2}{\tau_i+\bar{\tau}_i+e_i+\bar{e}_i}\right)ds}, \quad (26)$$

and $\bar{u}_i$ has been expressed as $\beta_0(r_i+\bar{r}_i)/(\tau_i+\bar{\tau}_i+e_i+\bar{e}_i)\bar{x}$.

Proof: The optimal control u_i^* is obtained by minimizing the following terms with respect to control u_i and $\bar{u}_i$,

$$\frac{\partial}{\partial u_i}\left[(\tau_i+e_i)\Big(u_i-\bar{u}_i-\frac{r_i}{\tau_i+e_i}\alpha_0(x-\bar{x})\Big)^2 + (\tau_i+\bar{\tau}_i+e_i+\bar{e}_i)\left(\bar{u}_i-\beta_0\frac{r_i+\bar{r}_i}{\tau_i+\bar{\tau}_i+e_i+\bar{e}_i}\bar{x}\right)^2\right]=0,$$

which yields $u_i=\frac{r_i}{\tau_i+e_i}\alpha_0(x-\bar{x})+\bar{u}_i$, where $\bar{u}_i=\beta_0(r_i+\bar{r}_i)/(\tau_i+\bar{\tau}_i+e_i+\bar{e}_i)\bar{x}$. Meanwhile, the mean field $\bar{x}(t)$ is derived by taking the expectation of the state dynamic equation in (16) and then solving the resulting differential equation for $\bar{x}(t)$. ∎

Theorem 2 states that the optimal number of computation tasks ECN i must offload from the TA in a cooperative scenario is given by (24). This number minimizes the cost incurred by ECN i where α_0 and β_0 satisfy (25), and the mean field $\bar{x}(t)$ satisfies (26). We can conclude that $u_i^*(t)$ not only depends on $x(t)$ but also on how much $x(t)$ exceeds $\bar{x}(t)$. Moreover, α_0 and β_0 capture how dependent $u_i^*(t)$ is on $x(t)-\bar{x}(t)$ and $\bar{x}(t)$, respectively.

VI. Mean-Field-Type Game-Based Computation Offloading Algorithms

This section presents the proposed algorithms that implement the MFTG-based computation offloading developed in the previous sections. A non-cooperative algorithm based on Theorem 1 is designed to simulate a scenario when the ECNs decide to minimize their own cost function. The algorithm can be implemented in a decentralized manner where each ECN decides for itself the optimal number of tasks to offload from the TA. Meanwhile, a cooperative algorithm based on Theorem 2 is designed for situations when the ECNs decide to minimize a global cost function. The algorithm can be implemented in a centralized manner where the TA decides for every ECN the optimal number of tasks to offload to the ECN. The proposed MFTG-based algorithms calculate the optimal solution $u_i^*(t)$ that corresponds to the portion of computation tasks that each ECN must offload in order to optimize its cost. As illustrated in Section VIII, these algorithms improve the system cost and benefit-cost ratio of the local computing and dynamic greedy algorithms for computation offloading. Thus, the proposed MFTG-based algorithms can improve the targeted network performance.

Fig. 2 illustrates the general procedure involved in the proposed algorithms. First, each ECN i determines its own cost coefficients r_i, τ_i, and e_i. Then, in the non-cooperative setting, ECN i computes the state and mean-state coupling coefficients α_i and β_i, while in the cooperative setting, the TA determines α_0 and β_0. These coefficients capture the effect of the state and mean-state to the optimal computation offloading control. At the same time, the TA determines the state $x(t)$ and mean-state $\bar{x}(t)$. Finally, the TA offloads a number of computation tasks to ECN i based on $u_i^*(t)$. The non-cooperative algorithm emulates a decentralized approach in which each ECN determines its own $u_i^*(t)$, while the cooperative algorithm follows a centralized approach in which the TA determines $u_i^*(t)$ of each ECN.

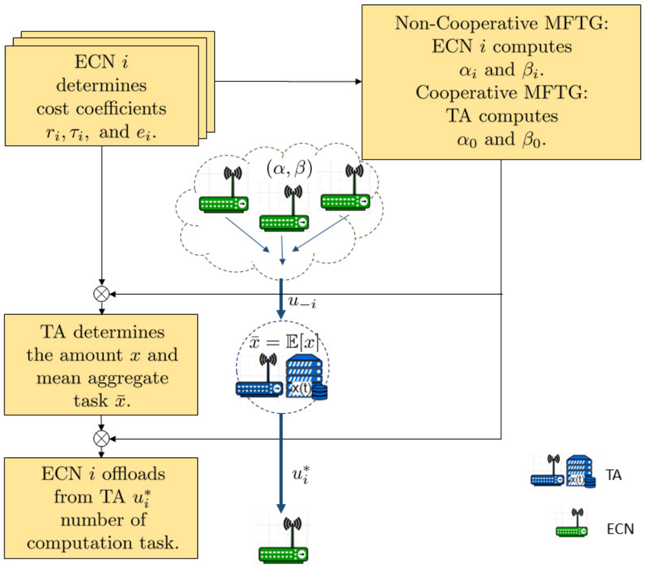


Fig. 2. Illustration of the proposed MFTG computation offloading algorithms.

These algorithms require a sample period T_s and number of samples M, instead of a specified terminal time T. One main reason for this requirement is to avoid network parameter updates every time t, which can now be done every T_s. In addition, the cell dimension L and the number of ECNs N are required as well. The location of each node in the cell is limited within the area defined by $[0, L] \times [0, L]$. The TA is located at $z_0 = [L/2, L/2]$, while the location z_i of each ECN i is distributed randomly in the area. In addition, for each ECN i, its computing capability f_i, cost weights $w_{d,i}$ and $w_{e,i}$ are also defined.

A. Non-Cooperative Computation Offloading

Since the non-cooperative solution using the direct approach stated in Theorem 1 assumes that each ECN has knowledge about the other ECNs, it has to be simplified in order to be implemented more practically. Let

$$\bar{\lambda} = \frac{1}{N}\sum_{j\in\mathcal{N}} \frac{r_j^2}{\tau_j + e_j}\alpha_j = \frac{1}{N}\sum_{j\in\mathcal{N}}\lambda_j,$$
$$\bar{\mu} = \frac{1}{N}\sum_{j\in\mathcal{N}} \frac{(r_j+\bar{r}_j)^2}{\tau_j + \bar{\tau}_j + e_j + \bar{e}_j}\beta_j = \frac{1}{N}\sum_{j\in\mathcal{N}}\mu_j.$$

Then, it follows that

$$\sum_{j\in\mathcal{N}\setminus i}\lambda_j = N\bar{\lambda} - \lambda_i,$$
$$\sum_{j\in\mathcal{N}\setminus i}\mu_j = N\bar{\mu} - \mu_i. \tag{27}$$

Consequently, (21) can be rewritten as

$$\frac{r_i^2}{\tau_i + e_i}\alpha_i^2 + 2\big(N\bar{\lambda} - \lambda_i - r_0\big)\alpha_i - \rho_i = 0,$$
$$\frac{(r_i+\bar{r}_i)^2}{\tau_i + \bar{\tau}_i + e_i + \bar{e}_i}\beta_i^2 + 2\big(N\bar{\mu} - \mu_i - (r_0 + \bar{r}_0)\big)\beta_i - (\rho_i + \bar{\rho}_i) = 0. \tag{28}$$

Algorithm 1 Non-Cooperative MFTG Computation Offloading

```
Set M, T_s, L, N, z_i, f_i, w_{e,i}, and w_{d,i}, ∀i ∈ N.
Initialize ē_i^(0), τ̄_i^(0), r̄_i^(0), ρ̄_i^(0), λ̄, and μ̄.
for m = 1 to M do
  for each ECN i in N do
    Compute e_i, τ_i, r_i using (1), (2), and (6), respectively.
    Compute α_i, β_i using (28).
    for each t in 0 ≤ t ≤ T_s do
      Observe and measure x(t). Calculate x̄(t) using (22).
      Calculate u_i^*(t) using (20).
    end for
    Update
      ē_i^(m) = (1/m)(e_i + (m − 1)ē_i^(m−1)),
      τ̄_i^(m) = (1/m)(τ_i + (m − 1)τ̄_i^(m−1)),
      r̄_i^(m) = (1/m)(r_i + (m − 1)r̄_i^(m−1)),
      ρ̄_i^(m) = (1/m)(ρ_i + (m − 1)ρ̄_i^(m−1)).
    Update λ̄ and μ̄ using (27).
  end for
end for
```

Meanwhile, the mean values $\bar{r}_i$, $\bar{\tau}_i$, $\bar{e}_i$, and $\bar{\rho}_i$ can be found using the law of large numbers. It states that a sample average

$$\bar{S}_m = \frac{1}{m}(y_1 + \cdots + y_m),$$

converges to the expected value $\bar{y} = \mathbb{E}[y]$ as $m \to \infty$. Hence, the relationship between the parameters r_i, τ_i, e_i, and ρ_i and their respective expected values is given by

$$\lim_{m\to\infty}\frac{1}{m}(r_{i,1} + \cdots + r_{i,m}) = \bar{r}_i,$$
$$\lim_{m\to\infty}\frac{1}{m}(\tau_{i,1} + \cdots + \tau_{i,m}) = \bar{\tau}_i,$$
$$\lim_{m\to\infty}\frac{1}{m}(e_{i,1} + \cdots + e_{i,m}) = \bar{e}_i,$$
$$\lim_{m\to\infty}\frac{1}{m}(\rho_{i,1} + \cdots + \rho_{i,m}) = \bar{\rho}_i, \tag{29}$$

$\forall i \in \mathcal{N}$.

As a result, Algorithm 1 shows the non-cooperative computation offloading algorithm based on Theorem 1. After setting up some network parameters, each ECN i needs to initialize $\bar{r}_i$, $\bar{\tau}_i$, $\bar{e}_i$, $\bar{\lambda}$, and $\bar{\mu}$. Then, each ECN i determines r_i, τ_i, and e_i. Also, each ECN i estimates α_i and β_i using (21). Meanwhile, the TA broadcasts $x(t)$ and $\bar{x}(t)$ to the ECNs. Consequently, each ECN i can now calculate and offload from the TA the optimal offloading portion $u_i^*(t)$ that minimizes their own cost. Lastly, ECN i updates $\bar{r}_i$, $\bar{\tau}_i$, $\bar{e}_i$, $\bar{\lambda}$, and $\bar{\mu}$.

B. Cooperative Computation Offloading

Algorithm 2 implements the cooperative computation offloading based on Theorem 2. It starts with setting up some network parameters. Then, each ECN i initializes parameters such as $\bar{r}_i$, $\bar{\tau}_i$, and $\bar{e}_i$ and transmits them to the TA. Next, the TA computes α_0 and β_0 based on (25). Afterwards, the TA can now compute the optimal offloading control $u_i^*(t)$ of each ECN based on the values of $x(t)$ and $\bar{x}(t)$. Then, the TA

Algorithm 2 Cooperative MFTG Computation Offloading

Set M, T_s, L, N, z_i, f_i, $w_{e,i}$, and $w_{d,i}$, $\forall i \in \mathcal{N}$.
Initialize $\bar{e}_i^{(0)}$, $\bar{\tau}_i^{(0)}$, and $\bar{r}_i^{(0)}$.
for $m = 1$ to M **do**
  for each ECN i in $\mathcal{N}$ **do**
    Compute e_i, τ_i, and r_i using (1), (2), and (6), respectively. Compute α_0 and β_0 using (25).
    for each t in $0 \le t \le T_s$ **do**
      Observe and measure $x(t)$. Calculate $\bar{x}(t)$ using (26). Calculate $u_i^*(t)$ using (24).
    end for
    Update
    $\bar{e}_i^{(m)} = \frac{1}{m}(e_i + (m-1)\bar{e}_i^{(m-1)})$,
    $\bar{\tau}_i^{(m)} = \frac{1}{m}(\tau_i + (m-1)\bar{\tau}_i^{(m-1)})$,
    $\bar{r}_i^{(m)} = \frac{1}{m}(r_i + (m-1)\bar{r}_i^{(m-1)})$.
  end for
end for

offloads the corresponding number of computation tasks to each ECN i. Finally, the TA updates $\bar{r}_i$, $\bar{\tau}_i$, and $\bar{e}_i$.

VII. Performance Evaluation

A. Baseline Approaches

To be able to evaluate the performance of the MFTG computation offloading algorithms proposed in this work, we compare them with two typical algorithms in computation offloading. The first algorithm is the local computing based on [14]. It finds the number of computation tasks $x_0(t)$ that can be executed locally in the TA such that it satisfies the required deadline d_0, $x_0(t)/f_0 < d_0$. The cost function of the TA for local computing is defined by

$$J_{lo} = \mathbb{E}\left[\int_0^T (w_{d,0}\tau_{lo}x_0(t) + w_{e,0}e_{lo}x_0(t))\, dt\right], \quad (30)$$

where $\tau_{lo} = 1/f_0$ refers to the number of time to execute a unit of computation task, $e_{lo} = \kappa_e f_0^2$ refers to the energy consumption per unit of computation task, and the constants $w_{d,0}$ and $w_{e,0}$ refer to the weights given by the TA to energy- and time-efficient optimization, respectively.

Another baseline algorithm used in this work is the dynamic greedy algorithm based on [14]. This algorithm finds the number of computation tasks $x_i(t)$ to be offloaded to ECN i that satisfies $x_i(t)/f_i < d_i$ where d_i is the deadline associated with $x_i(t)$. The cost function of ECN i for dynamic greedy computing is defined by

$$J_{dg,i} = \mathbb{E}\left[\int_0^T (w_{d,i}\tau_{dg,i}x_i(t) + w_{e,i}e_{dg,i}x_i(t))\, dt\right], \quad (31)$$

where $\tau_{dg,i} = 1/f_i$ refers to the number of time to execute a unit of computation task, $e_{dg,i} = \kappa_e f_i^2$ refers to the energy consumption per unit of computation task, and the constants $w_{d,i}$ and $w_{e,i}$ refer to the weights given by ECN i to energy- and time-efficient optimization, respectively.

To bridge the gap between the baseline algorithms with linear cost functions and the proposed MFTG-based algorithms with quadratic cost functions, a quadratic term is added to the linear costs so that

$$J_{lo} = \mathbb{E}\left[\int_0^T (a_1\xi_0\ x_0(t) + a_2\xi_0^2\ x_0^2(t))\, dt\right],$$
$$J_{dg,i} = \mathbb{E}\left[\int_0^T (a_1\xi_i x_i(t) + a_2\xi_i^2 x_i^2(t))\, dt\right], \quad (32)$$

where $\xi_0 = w_{d,0}\tau_{lo} + w_{e,0}e_{lo}$, $\xi_i = w_{d,i}\tau_{dg,i} + w_{e,i}e_{dg,i}$, a_1 and a_2 as constants with $a_2 \ll a_1$.

B. Performance Metrics

The following metrics are calculated in order to compare the performance of the computation offloading approaches presented in this paper.

An offloading control fraction $p_i(t)$ is the ratio between the offloading control $u_i(t)$ and the state $x(t)$ of the TA, $p_i(t) = u_i(t)/x(t)$. Consequently, an optimal offloading control fraction $p_i^*(t)$ is written mathematically as

$$p_i^*(t) = \frac{u_i^*(t)}{x(t)}, \quad (33)$$

where $u_i^*(t)$ is the optimal offloading control of ECN i.

The two main parameters in the MFTG formulation of computation offloading that limit the control of an ECN are energy consumption and computation or execution time. Consequently, the performance of the computation offloading methods are evaluated through energy efficiency and time efficiency. By efficiency, we mean how much computation tasks are executed per unit of network resource. Hence, energy efficiency is defined as the ratio between the number of computation tasks and the associated energy consumption. For an MECN with N ECNs, the network energy efficiency is written as

$$\eta_e = \frac{x(t)}{\sum_{i=1}^N \kappa_{e,i} f_i^2 u_i(t)}, \quad (34)$$

where the ratio is taken between the total number of tasks at the TA and the total energy consumed by all the ECNs.

Meanwhile, time efficiency refers to the ratio between the number of computation tasks and the corresponding computation or execution time spent. For an MECN with N ECNs, the network time efficiency is given by

$$\eta_d = \frac{x(t)}{\sum_{i=1}^N \frac{u_i(t)}{f_i}}, \quad (35)$$

where the ratio is taken between the total number of tasks at the TA and the cumulative computation time of the tasks through the ECNs.

System cost is another way of comparing the computation offloading methods. It consists of the computation offloading cost and the overhead cost associated with each computation offloading algorithm. For both MFTG approaches, overhead exists between an ECN and the TA. Thus, the system costs for the non-cooperative and cooperative MFTG methods are given by

$$C_{nc} = \sum_{i=1}^N \left(\tilde{J}_i + 2\delta_i\theta_{i,0}\right),$$
$$C_{co} = \tilde{J}_0 + \sum_{i=1}^N 2\delta_i\theta_{i,0}, \quad (36)$$

where δ_i is the cost associated per overhead while $\theta_{i,0}$ is the number of overhead between ECN i and the TA. For the local computing algorithm, since the TA does not collaborate with any computing nodes, the overhead is zero. In the dynamic greedy offloading, overhead exists not only between the TA and ECNs but also between any two ECNs. Thus, the system costs for these two baseline approaches are

$$C_{lo} = J_{lo},$$
$$C_{dg} = \sum_{i=1}^{N} \left(J_{dg,i} + N\delta_{dg,i}\theta_{dg,i}\right), \tag{37}$$

where $\delta_{dg,i}$ is the cost associated per overhead while $\theta_{dg,i}$, is the number of overhead from ECN i to another computing node. In this work, overhead refers to the delay associated with the transmission time of overhead messages between an any two computing nodes. For an overhead message of length b, the transmission time is b/r, where r is the rate at which the message is transmitted.

Lastly, to be able to compare the computational overhead and benefits of the proposed algorithms, we perform a benefit-cost analysis on the proposed algorithms as well as the typical algorithms in computation offloading. The metric we used to compare the algorithms is called the benefit-cost ratio B/C. The benefit B of each algorithm is the weighted sum of the energy and time efficiencies

$$B = w_d\eta_d + w_e\eta_e, \tag{38}$$

where the constants w_d and w_e denote the weights given to the efficiencies and $w_d + w_e = 1$. The cost C used for each algorithm is the system cost defined previously.

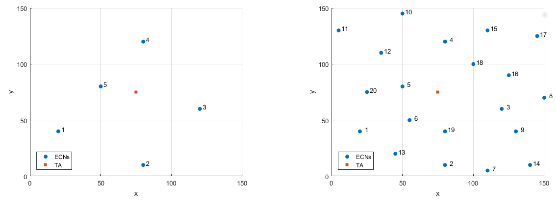
VIII. Simulation Results and Discussion

A. Simulation Setup

The simulations in this paper can be extended to networks containing multiple cells assuming that each cell operate independently of each other. That is, the TA of a cell can offload tasks only to ECNs located in its cell. Moreover, the interference between cells are minimized using techniques such as FDMA and SDMA. In addition, each simulation has been performed over 100 iterations and the average of the results has been drawn in each figure.

Consider one network cell with an area of 150×150 m^2 containing one TA located at the center of the cell. The number of ECNs has been varied from 2 to 20. The location of each ECN is randomly distributed within the cell. Fig. 3 shows the locations of the ECNs for the sparse MECN with $N = 5$ and the dense MECN with $N = 20$ utilized in the following simulations. The end users are located randomly within the cell. The number of end users are set at 50. The computation tasks arrive at the TA randomly, and the users are assumed to submit an average of 5 Tcycles of computation tasks.

Assume that the TA has a transmit power of 100 mW, a maximum incoming rate of computation task $R_0 =$ 10 Gbps, and a maximum capacity C_0 of 10 Tb worth of computation tasks. Meanwhile, the computing nodes have a transmit power p_i of 100 mW, capacity C_i of 100 Gb worth of computation tasks. The computing capability f_i of each computing node is randomly selected from 10, 12, and 14 Tcycles/s. The cost weights $w_{d,i}$ and $w_{e,i}$ for the computation time and energy consumption are both set to 0.5. For SINR γ_i computations, the channel gain model used between any two nodes i and j is $g_{i,j} = d_{i,j}^{-\alpha}$ where $d_{i,j}$ denotes the distance between the two nodes and the path loss exponent $\alpha = 4$. Meanwhile, the background noise N_0 is set at -100 dBm. The quadratic cost constants for the baseline algorithms are set at $a_1 = 0.9$ and $a_2 = 0.1$.



(a) A sparse MECN with $N = 5$. (b) A dense MECN with $N = 20$.

Fig. 3. Location of the ECNs.

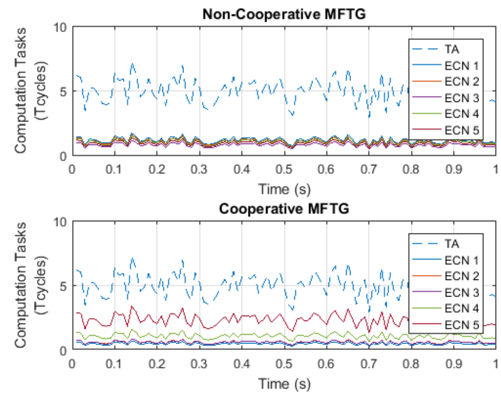
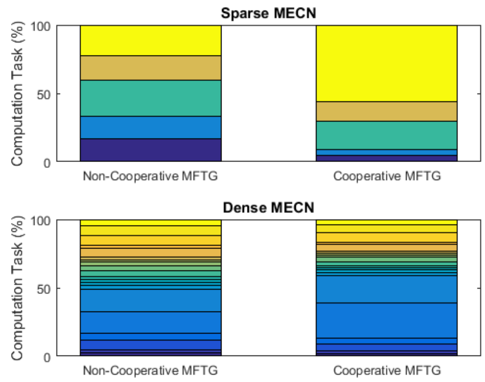


Fig. 4. Optimal offloading control of the ECNs in time domain.

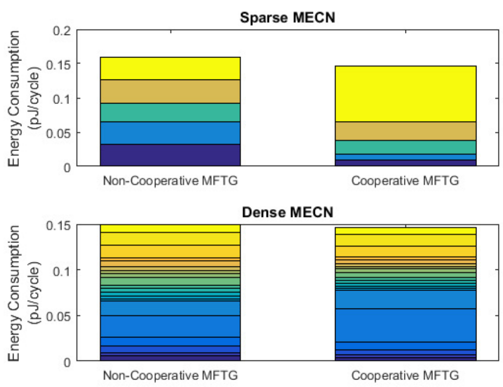
B. Optimal Offloading Control

In the first of part the simulations, the optimal offloading control $u_i^*(t)$ of ECN i based on the feedback controls $u_{-i}^*(t)$ of other ECNs is computed using the MFTG computation offloading algorithms. Fig. 4 shows the plots of $u_i^*(t)$ for the sparse MECN in both non-cooperative and cooperative MFTG scenarios as well as the number of computation tasks $x(t)$ at the TA. We can conclude from this figure that the two MFTG algorithms divide the computation tasks at the TA to the ECNs differently.

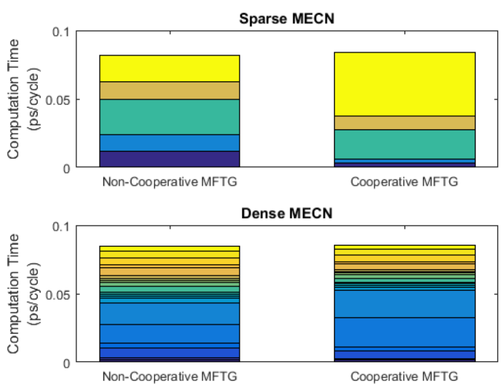
The partition of computation tasks among the ECNs is shown in Fig. 5 where each color denotes the particular share of an ECN. Fig. 5a shows the average percentage of offloaded computation tasks from the TA to each ECN i in the sparse and dense MECN. In the sparse MECN, the non-cooperative MFTG approach distributes the



(a) Average percentage of computation task.

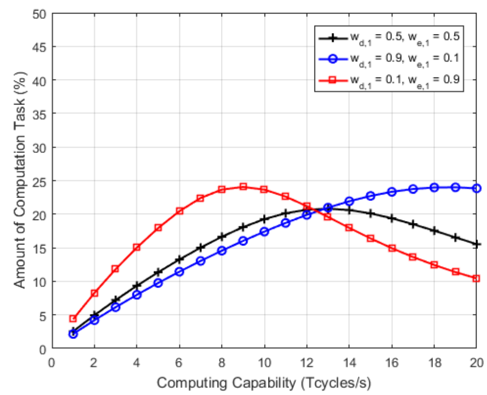


(b) Average energy consumption of ECNs.

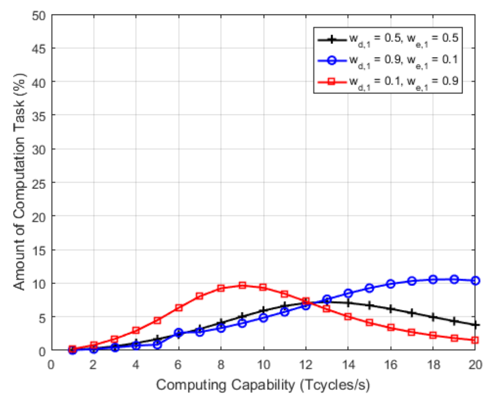


(c) Average computation time of ECNs.

Fig. 5. Partition of aggregate computation tasks for each ECN.



(a) Non-cooperative MFTG.



(b) Cooperative MFTG.

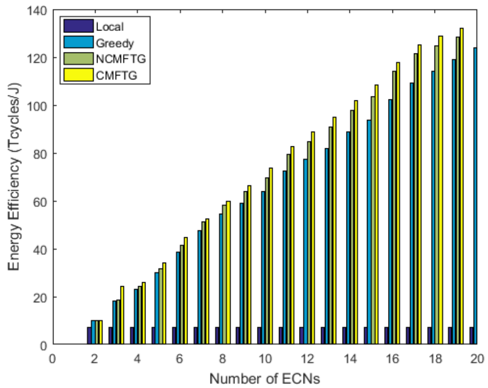
Fig. 6. Effect of computing capability and cost weights to the optimal computation offloading control.

tasks more evenly than the cooperative approach. Meanwhile, in the dense MECN, the distribution of tasks is more similar between the two MFTG offloading algorithms. The figure also implies that the offloading controls change accordingly when the number of ECNs is varied. Meanwhile, Fig. 5b presents the energy consumption per cycle of each ECN, and Fig. 5c shows the computation time per cycle contributed by each ECN.

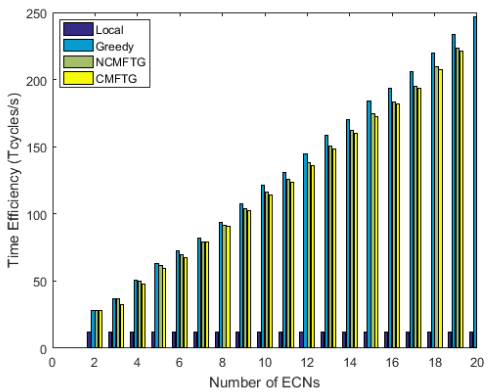
Next, the effects of computing capability f_i and the cost weights $w_{d,i}$ and $w_{e,i}$ of ECN i to its optimal control $u_i^*(t)$ are investigated. While Fig. 6 shows $u_i^*(t)$ as a fraction of $x(t)$ averaged over time for ECN 1, the analyses that follow can be generalized to any ECNs. From the figure, it can be noticed that as the computing capability f_i of an ECN increases, the average percentage of aggregate computation task it offloads rises up to a certain point, then it decreases. The reason for this trend is the compromise between minimizing computation time and energy consumption. When f_i is low, the energy consumption of ECN i is also low; however, the computation time to execute the offloaded tasks is high. As f_i becomes higher, the energy consumption of an ECN increases while the computation time to execute the offloaded tasks becomes lesser.

Meanwhile, as computation time is given more weight by increasing its weight from 0.1 to 0.9, the curve shifts to the right. This means that as an ECN prioritizes minimizing computation time, the computing capability at which it can afford to offload the highest percentage of the aggregate computation task increases. However, as more weight is given to energy consumption from 0.1 to 0.9, the curve shifts to the left. That is, to lower the energy consumption of an ECN, the computing capability at which its offloading percentage is at the highest decreases.

In summary, an ECN with lower computing capability offloads more from the TA if minimizing the energy consumption is more critical, as shown by the red curves.



(a) Average network energy efficiency.



(b) Average network time efficiency.

Fig. 7. Average network efficiency of the computation offloading algorithms.

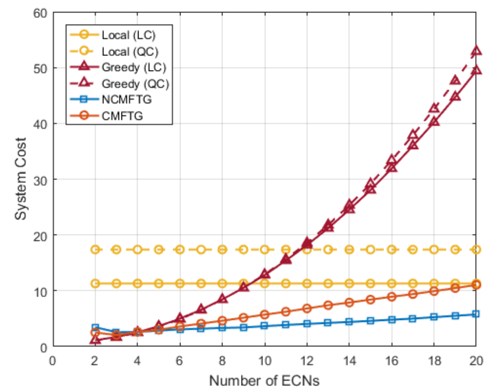


Fig. 8. Average network cost of the computation offloading algorithms.

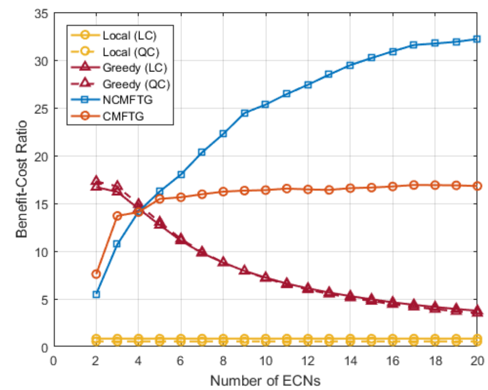


Fig. 9. Benefit-cost ratio of the computation offloading algorithms.

However, if the priority is to minimize computation time, then an ECN with higher computing capability offloads more from the TA, as shown by the blue curves.

C. Network Efficiency

The energy efficiency η_e using different computation offloading approaches are compared in Fig. 7a. From the figure, the cooperative MFTG (CMFTG) approach has better η_e than the non-cooperative MFTG (NCMFTG) approach. However, both MFTG algorithms have higher η_e than the local and dynamic greedy algorithms. This is one of the reasons that justifies the significance of computation offloading in MECN.

Meanwhile, the time efficiency η_d of the network under different computation offloading approaches are displayed in Fig. 7b. We can conclude from the figure that MFTG computation offloading approaches maintain a competitive η_d against the dynamic greedy algorithm.

Hence, we can conclude that the MFTG offloading algorithms can be as efficient as the dynamic greedy algorithm which requires full knowledge of the characteristics of all the ECNs. In the following subsection, we compare the system costs of the computation offloading algorithms.

D. System Cost and Benefit-Cost Ratio

Fig. 8 presents the system cost sustained by each computation algorithm investigated in this work. The system cost of the dynamic greedy approach is higher than both the MFTG approaches because the overhead required to implement the greedy algorithm is larger than the overhead required by the MFTG approaches. The system cost of the local computing approach is shown for comparison purposes even though it does not require the use of ECNs. Between the two MFTG approaches, the cooperative approach has lower system cost than the non-cooperative approach when the number of ECNs is lower. However, as the number of ECNs increases, the system cost of the non-cooperative approach becomes lower than that of the cooperative approach.

Fig. 9 shows the benefit-cost ratio B/C for each computation offloading approaches. We can conclude that the non-cooperative MFTG approach has the best B/C, followed by the cooperative MFTG approach. The benefits of the MFTG approaches are contributed by the energy- and time-efficient partition of computation tasks as well as the low number of network overhead required to implement the offloading.

Moreover, we can tell that the system cost and benefit-cost ratio of the local computing and dynamic greedy algorithms with quadratic cost (QC) are almost equivalent to the system cost and benefit-cost ratio of the original linear cost (LC). Therefore, we can conclude that the form of the cost function does not affect the performance of the algorithms significantly since the main difference between the cost functions of the proposed work and the baseline algorithms is the overhead cost.

IX. Conclusion

Multi-access edge computing networks (MECN) reduce the latency inherent in cloud computing networks by performing the tasks in an edge network near the network users rather than in a cloud network. Computation offloading is one of the services in an MECN in which computation-intensive tasks in a computing node may be offloaded to other computing nodes in the network. In this work, computation offloading problem has been formulated using mean-field-type game (MFTG). Then, non-cooperative and cooperative computation offloading algorithms have been proposed. These algorithms search for the optimal computation offloading controls of each computing node in an MECN. The non-cooperative algorithm is a decentralized approach since each computing node determines its own offloading control. Nevertheless, the cooperative algorithm is a centralized approach in which the network determines the offloading control of each computing node. Lastly, the simulation results have indicated that MFTG is an effective way to model computation offloading in MECNs.

References

[1] S. Wang, X. Zhang, Y. Zhang, L. Wang, J. Yang, and W. Wang, "A survey on mobile edge networks: Convergence of computing, caching and communications," *IEEE Access*, vol. 5, pp. 6757–6779, Mar. 2017.

[2] Z. Han, D. Niyato, W. Saad, T. Basar, and A. Hjørungnes, *Game Theory in Wireless and Communication Networks: Theory, Models and Applications*. Cambridge, U.K.: Cambridge Univ. Press, Jan. 2012.

[3] Z. Han and K. J. Ray Liu, *Resource Allocation for Wireless Networks: Basics, Techniques, and Applications*. Cambridge, U.K.: Cambridge Univ. Press, Apr. 2008.

[4] J. M. Lasry and P. L. Lions, "Mean field games," *Jpn. J. Math.*, vol. 2, no. 1, pp. 229–260, Mar. 2007.

[5] B. Djehiche, A. Tcheukam, and H. Tembine, "Mean-field-type games in engineering," *AIMS Electron. Electr. Eng.*, vol. 1, pp. 18–73, 2017.

[6] C. Yang, J. Li, P. Semasinghe, E. Hossain, S. M. Perlaza, and Z. Han, "Distributed interference and energy-aware power control for ultra-dense D2D networks: A mean field game," *IEEE Trans. Wireless Commun.*, vol. 16, no. 2, pp. 1205–1217, Feb. 2017.

[7] R. Couillet, S. Perlaza, H. Tembine, and M. Debbah, "Electric vehicles in the smart grid: A mean field game analysis," *IEEE J. Sel. Areas Commun.*, vol. 30, no. 6, pp. 1086–1096, Jul. 2012.

[8] Y. Wang, F. R. Yu, H. Tang, and M. Huang, "A mean field game theoretic approach for security enhancements in mobile ad hoc networks," *IEEE Trans. Wireless Commun.*, vol. 13, no. 3, pp. 1616–1627, Mar. 2014.

[9] X. Ge, H. Jia, Y. Zhong, Y. Xiao, Y. Li, and B. Vucetic, "Energy efficient optimization of wireless-powered 5G full duplex cellular networks: A mean field game approach," *IEEE Trans. Green Commun. Netw.*, vol. 3, no. 2, pp. 455–467, Jun. 2019.

[10] 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 Int. Conf. Commun. (ICC)*, Shanghai, China, May 2019, pp. 1–6.

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

[12] H. Guo and J. Liu, "Collaborative computation offloading for multiaccess edge computing over fiber–wireless networks," *IEEE Trans. Veh. Technol.*, vol. 67, no. 5, pp. 4514–4526, May 2018.

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

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

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

[16] M. Chen and Y. Hao, "Task offloading for mobile edge computing in software defined ultra-dense network," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 3, pp. 587–597, Mar. 2018.

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

[18] Z. Ning, P. Dong, X. Kong, and F. Xia, "A cooperative partial computation offloading scheme for mobile edge computing enabled Internet of Things," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4804–4814, Jun. 2019.

[19] C. You, K. Huang, and H. Chae, "Energy efficient mobile cloud computing powered by wireless energy transfer," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 5, pp. 1757–1771, May 2016.

[20] S. Guo, B. Xiao, Y. Yang, and Y. Yang, "Energy-efficient dynamic offloading and resource scheduling in mobile cloud computing," in *Proc. IEEE INFOCOM*, San Francisco, CA, USA, Apr. 2016, pp. 1–6.

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

[22] S. Yu, R. Langar, X. Fu, L. Wang, and Z. Han, "Computation offloading with data caching enhancement for mobile edge computing," *IEEE Trans. Veh. Technol.*, vol. 67, no. 11, pp. 11098–11112, Nov. 2018.

[23] F. Wang, J. Xu, X. Wang, and S. Cui, "Joint offloading and computing optimization in wireless powered mobile-edge computing systems," *IEEE Trans. Wireless Commun.*, vol. 17, no. 3, pp. 1784–1797, Mar. 2018.

[24] J. Barreiro-Gomez, T. E. Duncan, and H. Tembine, "Linear-quadratic mean-field-type games-based stochastic model predictive control: A microgrid energy storage application," in *Proc. Amer. Control Conf. (ACC)*, Philadelphia, PA, USA, Jul. 2019, pp. 3224–3229.

[25] J. Gao and H. Tembine, "Distributed mean-field-type filter for vehicle tracking," in *Proc. Amer. Control Conf. (ACC)*, Seattle, WA, USA, May 2017, pp. 4454–4459.

[26] J. Barreiro-Gomez and H. Tembine, "Blockchain token economics: A mean-field-type game perspective," *IEEE Access*, vol. 7, pp. 64603–64613, Jul. 2019.

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

[28] T. Soyata, R. Muraleedharan, C. Funai, M. Kwon, and W. Heinzelman, "Cloud-vision: Real-time face recognition using a mobile-cloudlet-cloud acceleration architecture," in *Proc. IEEE Symp. Comput. Commun.*, Jul. 2012, pp. 59–66.

[29] M. Greer, *Electricity Cost Modeling*. Amsterdam, The Netherlands: Elsevier, 2010.

[30] W. Zhang, Y. Wen, K. Guan, D. Kilper, H. Luo, and D. Wu, "Energy-optimal mobile cloud computing under stochastic wireless channel," *IEEE Trans. Wireless Commun.*, vol. 12, no. 9, pp. 4569–4581, Sep. 2013.

[31] S. Samarakoon, M. Bennis, W. Saad, M. Debbah, and M. Latva-Aho, "Ultra dense small cell networks: Turning density into energy efficiency," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 5, pp. 1267–1280, May 2016.

[32] Y. Wen, W. Zhang, and H. Luo, "Energy-optimal mobile application execution: Taming resource-poor mobile devices with cloud clones," in *Proc. IEEE INFOCOM*, Orlando, FL, USA, Mar. 2012, pp. 2716–2720.

[33] A. P. Miettinen and J. K. Nurminen, "Energy efficiency of mobile clients in cloud computing," in *Proc. 2nd USENIX Conf. Hot Topics Cloud Comput.*, Boston, MA, USA, Jun. 2010, pp. 4–11.

[34] B. Jovanovic, "Selection and the evolution of industry," *Econometrica*, vol. 50, no. 3, pp. 649–670, May 1982.

[35] B. Jovanovic and R. W. Rosenthal, "Anonymous sequential games," *J. Math. Econ.*, vol. 17, no. 1, pp. 77–87, Jan. 1988.

[36] J. Bergin and D. Bernhardt, "Anonymous sequential games with aggregate uncertainty," *J. Math. Econ.*, vol. 21, no. 6, pp. 543–562, Jan. 1992.

[37] A. Tcheukam and H. Tembine, "Mean-field-type games for distributed power networks in presence of prosumers," in *Proc. Chin. Control Decis. Conf. (CCDC)*, Yinchuan, China, May 2016, pp. 446–451.

[38] A. T. Siwe and H. Tembine, "Network security as public good: A mean-field-type game theory approach," in *Proc. 13th Int. Multi-Conference Syst., Signals Devices (SSD)*, Leipzig, Germany, Mar. 2016, pp. 601–606.

[39] B. Djehiche, A. Tcheukam, and H. Tembine, "A mean-field game of evacuation in multilevel building," *IEEE Trans. Autom. Control*, vol. 62, no. 10, pp. 5154–5169, Oct. 2017.

[40] T. E. Duncan and H. Tembine, "Linear-quadratic mean-field-type games: A direct approach," *Games*, vol. 9, no. 1, pp. 7–24, Feb. 2018.

Reginald A. Banez received the B.S. degree in electronics and communications engineering from Mapúa University, Manila, Philippines, in 2008, and the master's and Ph.D. degrees in electrical engineering from the University of Houston, Houston, TX, USA, in 2015 and 2020, respectively. His research interests include communication networks, mean-field games, and machine learning.

Hamidou Tembine (Senior Member, IEEE) graduated in applied mathematics from Ecole Polytechnique, Palaiseau, France, and received the Ph.D. degree from INRIA and University of Avignon, France, and the master's degree in game theory and economics. His main research interests are learning, evolution, and games. He has been a Visiting Researcher with the University of California at Berkeley, Berkeley, CA, USA; University of McGill, Montreal, Quebec, Canada; University of Illinois at Urbana-Champaign (UIUC), Champaign, IL, USA; Ecole Polytechnique Federale de Lausanne (EPFL), Switzerland; and University of Wisconsin-Madison, Madison, WI, USA. He is a prolific researcher and holds over 150 scientific publications, including magazines, letters, journals, and conferences. He is the author of the book on "*Distributed Strategic Learning for Engineers*" (CRC Press, Taylor and Francis 2012) which received book award 2014, and a coauthor of the book "*Game Theory and Learning in Wireless Networks*" (Elsevier Academic Press). He has been the co-organizer of several scientific meetings on game theory in networking, wireless communications, and smart energy systems. He has been a TPC member and reviewer for several international journals and conferences. He is a Next Einstein Fellow, Class of 2017. He is a Simons participant and a Senior Fellow at IPAM, UCLA. In 2014, he received the IEEE ComSoc Outstanding Young Researcher Award for his promising research activities for the benefit of the society. He was a recipient of over 10 best paper awards in the applications of game theory.

Lixin Li (Member, IEEE) received the B.Sc. and M.Sc. degrees in communication engineering, and the Ph.D. degree in control theory and its applications from Northwestern Polytechnical University (NPU), Xi'an, China, in 2001, 2004, and 2008, respectively. From 2008 to 2010, he was a Post-Doctoral Fellow with NPU. In 2017, he was a Visiting Scholar with the University of Houston, Houston, TX, USA. He is currently an Associate Professor with the School of Electronics and Information, NPU. He has authored or coauthored more than 150 peer-reviewed papers in many prestigious journals and conferences, and holds 12 patents. His current research interests include wireless communications, game theory, and machine learning. He received the 2016 NPU Outstanding Young Teacher Award, which is the highest research and education honors for young faculties in NPU.

Chungang Yang (Senior Member, IEEE) received the bachelor's and Ph.D. degrees from Xidian University, Xi'an, China, in 2006 and 2011, respectively.

From September 2010 to March 2011, he held the visiting scholar position with the Department of Electrical and Computer Engineering, Michigan Technological University (MTU). From March 2015 to March 2016, he held the visiting scholar position with the Department of Electrical and Computer Engineering, University of Houston (UH). He is currently a Full Professor with Xidian University, where he leads the research team of "GUIDE, Game, Utility, artificial Intelligent Design for Emerging Communications." His research interests include resource and interference management and network optimization for cognitive radio networks, heterogeneous green cellular networks, with a focus on game theory for wireless communication and computing networks. He has edited two books, and published more than 20 journals and over 20 conferences. His recent research interests are artificial intelligent B5G/6G wireless mobile networks, intent-driven/based networks (IDN/IBN), and space-terrestrial networks (STN).

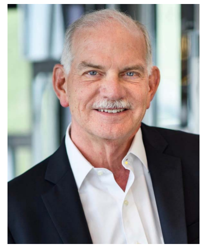
Lingyang Song (Fellow, IEEE) received the Ph.D. degree from the University of York, U.K., in 2007, where he received the K. M. Stott Prize for Excellent Research.

He worked as a Research Fellow with the University of Oslo, Norway, before rejoining Philips Research UK in March 2008. In May 2009, he joined the School of Electronics Engineering and Computer Science, Peking University, where he is currently a Boya Distinguished Professor. His main research interests include wireless communications, mobile computing, and machine learning. He has coauthored of four textbooks, including "*Wireless Device-to- Device Communications and Networks*" and "*Full-Duplex Communications and Networks*" (Cambridge University Press, U.K.). He has co-edited two books: "*Orthogonal Frequency Division Multiple Access (OFDMA)-Fundamentals and Applications*" and "*Evolved Network Planning and Optimization for UMTS and LTE* (IEEE ComSoc Best Readings)" (Auerbach Publications, CRC Press, USA). He is a coauthor of 10 best paper awards and one Best Demo Award, including the IEEE Leonard G. Abraham Prize in 2016 (publication in the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS in the previous three calendar years), best paper awards from the IEEE Communication Society Flagship Conference: IEEE ICC 2014, IEEE ICC 2015, IEEE Globecom 2014, and the Best Demo Award in the ACM Mobihoc 2015. He received the National Science Fund for Distinguished Young Scholars in 2017, the First Prize in Nature Science Award of the Ministry of Education of China in 2017, the National Science Fund for Excellent Young Scholars in 2013, and the IEEE Communications Society Asia Pacific (AP) Young Researcher Award in 2012.

Dr. Song has served as the TPC Co-Chair for ICUFN 2011/2012 and IEEE ICCC 2019. He has served as the Symposium Co-Chair for IEEE ICC 2014/2016, IEEE VTC 2016 Spring, and IEEE Globecom 2016. He has served as the Vice-Chair (since 2016) of the IEEE Communications Society Cognitive Network Technical Committee, and the Vice-Chair (2016–present) of the IEEE Communications Society Asia Pacific Board Technical Affairs Committee. He has served as a Distinguished Lecturer of the IEEE Communications Society (2015–2018), an Area Editor of the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY (2019–present), an Editor of the IEEE TRANSACTIONS ON COMMUNICATIONS (2019–present), an Editor of *China Communications* (2015–present), and the Editor of the TRANSACTIONS ON WIRELESS COMMUNICATIONS (2013–2018). He also serves as a Section Editor for the *Springer Handbook of Cognitive Radio* (2016–present).

Zhu Han (Fellow, IEEE) received the B.S. degree in electronic engineering from Tsinghua University, in 1997, and the M.S. and Ph.D. degrees in electrical and computer engineering from the University of Maryland, College Park, MA, USA, in 1999 and 2003, respectively.

From 2000 to 2002, he was a Research and Development Engineer of JDSU, Germantown, MA, USA. From 2003 to 2006, he was a Research Associate with the University of Maryland. From 2006 to 2008, he was an Assistant Professor with Boise State University, Boise, ID, USA. He is currently a John and Rebecca Moores Professor with the Electrical and Computer Engineering Department as well as with the Computer Science Department, University of Houston, Houston, TX, USA. His research interests include wireless resource allocation and management, wireless communications and networking, game theory, big data analysis, security, and smart grid. He received an NSF Career Award in 2010, the Fred W. Ellersick Prize of the IEEE Communication Society in 2011, the EURASIP Best Paper Award for the *Journal on Advances in Signal Processing* in 2015, the IEEE Leonard G. Abraham Prize in the field of Communications Systems (Best Paper Award in IEEE JSAC) in 2016, and several best paper awards in IEEE conferences. He was an IEEE Communications Society Distinguished Lecturer from 2015 to 2018, AAAS fellow since 2019, and ACM distinguished member since 2019. He is a 1% Highly Cited Researcher since 2017, according to the Web of Science. He is also the winner of the 2021 IEEE Kiyo Tomiyasu Award, for outstanding early to mid-career contributions to technologies, holding the promise of innovative applications, with the following citation: "for contributions to game theory and distributed management of autonomous communication networks."



H. Vincent Poor (Life Fellow, IEEE) received the Ph.D. degree in EECS from Princeton University in 1977.

From 1977 to 1990, he was on the faculty of the University of Illinois at Urbana-Champaign. Since 1990, he has been on the faculty at Princeton, where he is currently the Michael Henry Strater University Professor of Electrical Engineering. From 2006 to 2016, he has served as the Dean of Princeton's School of Engineering and Applied Science. He has also held visiting appointments at several other universities, including most recently at Berkeley and Cambridge. His research interests are in the areas of information theory, signal processing and machine learning, and their applications in wireless networks, energy systems, and related fields. Among his publications in these areas is the recent book *Multiple Access Techniques for 5G Wireless Networks and Beyond* (Springer, 2019). He is a member of the National Academy of Engineering and the National Academy of Sciences, and is a foreign member of the Chinese Academy of Sciences, the Royal Society, and other national and international academies. Recent recognition of his work includes the 2017 IEEE Alexander Graham Bell Medal and a D.Eng. *honoris causa* from the University of Waterloo awarded in 2019.