Mean Field Game ANN for Downlink Computation Offloading in MEC Networks

Min Chen*, Haibo Wang*, Matthieu De Mari†
*Research Institute of Broadband Wireless Mobile Communication, Beijing Jiaotong University, China
†Information Systems Technology and Design Pillar, Singapore University of Technology and Design, Singapore
Corresponding author: Haibo Wang, Email: hbwang@bjtu.edu.cn

***Abstract*—Under the trend of the increasing popularity of computation-intensive applications, computation offloading has been extensively studied to tackle the battery and computing ability limits of smart mobile devices. In this paper, a computation offloading scheme is investigated in an ultra-dense Mobile Edge Computing network, where the distributed offloading strategy of each Edge Computing Node (ECN) is formulated as a multi-user non-cooperative dynamic stochastic game, and transformed into a much tractable mean field game (MFG) to reduce the computational complexity. The two coupled partial differential equations, namely the Hamilton-Jacobi-Bellman (HJB) and Fokker-Planck-Kolmogorov equations are deduced, and an Artificial Neural Network (ANN) is designed to implement supervised learning to approximately solve the HJB equation with an analytic closed-form. The simulation results prove that the proposed ANN could be trained efficiently to solve the HJB equation in lower computational cost, and the MFG-based strategy yields lower cost with compared to conventional methods.**

***Index Terms*—Computation offloading, mean field games, supervised learning**

I. Introduction

Computation offloading via Mobile Edge Computation (MEC) networks enables sophisticated applications to the mobile devices with constrained battery lifetime, and the strict delay requirements could be met due to the short distance between the devices and the computation nodes [1]. In [2], computation offloading decision and computation resource allocation are jointly optimized in a cloud-MEC collaborative scheme. The authors in [3] pursued energy-efficient MEC designs to minimize the energy consumption in both partial and binary offloading. In [4], a two-stage heuristic optimization algorithm is proposed for the joint optimization problem of computation offloading and resource allocation. In addition to centralized optimization, the distributed offloading scheme has research significance in ultra-dense networks, where gaming is usually applied for distributed strategies, and mean field game (MFG) for large-scale interactions in gaming. In [5], a dynamic stochastic game is formulated to investigate the energy efficiency performance of optimal proactive scheduling strategies. The effects of the interference between D2D users are considered to formulate a MFG for a distributed power control method in [6]. In [7], a mean-field-type game approach is proposed for each computing node to compute the portion of the aggregate computation task for offloading.

This work was supported by the National Natural Science Foundation of China (Grant No.61471026).

The MFG relies on two coupled partial differential equations (PDEs), namely a Hamilton-Jacobi-Bellman (HJB) equation and a Fokker-Planck-Kolmogorov (FPK) equation. Finite Difference Method (FDM), along with Finite Element Method (FEM) and Finite Volume Method (FVM) have been acknowledged as effective approaches to ordinary and partial differential equations [8], but they yield high complexity for establishing a meshing and solving the PDE numerically, and the obtained solutions are discrete with limited differentiability. In this work, we implement supervised learning to solve the PDE by an Artificial Neural Network (ANN), which provides a differentiable and closed analytic form approximate solution, and only needs one training to be able to solve the PDE for any set of parameters.

Our main contributions are summarized as follows:

- We formulate the computation offloading problem of Edge Computing Nodes (ECNs) as a non-cooperative Dynamic Stochastic Game (DSG), in consideration of the prohibitively complexity brought by the great number of ECNs, we transform the DSG into an MFG, which is a two-body gaming.
- The optimal strategy for the MFG relies on the HJB equation, which is solved by a designed ANN, and the solution is also compared with that solved by FDM to verify the validity and effectiveness.
- Simulation results reveal that the proposed neural network could fit the HJB equation well, and the offloading strategy obtained by the MFG yields smaller cumulative cost than conventional strategies.

II. System Model and Problem Formulation

We investigate an Orthogonal Frequency Division Multiplexing (OFDM) MEC network as shown in Fig. 1, where smart mobile devices can offload their computation-intensive tasks to the task aggregator (TA) in the area or cells they attached to. The TA can collect all the computation tasks in its serving region, and around which the ECNs, denoted by $\mathcal{N}$={ECN_1, $\cdots$, ECN_n, $\cdots$, ECN_N}, have idle computation resource may devote their energy to offload computation tasks from the TA. The ECNs may not necessarily belong to the same operator, e.g., the ECNs are deployed by some Edge Computing operators whereas the TA belongs to an IoT/Cloud network operator, therefore each ECN will be paid by the TA for the processed tasks in monetary form.

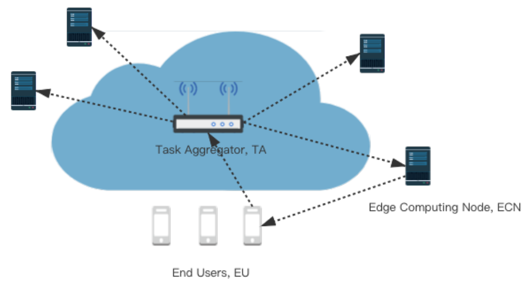


Fig. 1. System model

A. Channel Model

Let $h_i(t)$ denote the channel gain between ECN_i and the TA at time-slot t under a block-fading channel model [9], $h_i(t)$ remains constant during each duration Δt of time slot $t \in \mathcal{T} = \{0, ..., T\}$, the channel dynamics are modeled as an Ornstein-Uhlenbeck process as follows [10]:

$$\Delta h_i(t) = \alpha(t, h_i(t))\Delta t + \sigma_{h,i}\Delta W(t), \quad (1)$$

where $\alpha(t, h_i(t))$ is a deterministic smooth function, and $\sigma_{h,i}$ with positive value characterizes the long term variance of the process, and $W(t)$ is a standard Brownian motion which models fast fading and prediction uncertainty [11].

B. Reward Model

The ECNs compete for the tasks to gain reward paid by the TA, and the reward depends on the computation offloading supply of the whole market. According to [12], the price of goods should gradually fall and flatten out with supply, so we define the price:

$$\psi_i(t) = \begin{cases} \kappa & N = 1, \\ \kappa \cdot \exp(-\xi \frac{\sum_{j=1, j\neq i}^{N} r_j(t)}{N-1}) & N \geq 2, \end{cases} \quad (2)$$

where κ is a basic price when there is no competition, ξ, takes a positive value, affects the steepness of the curve, and $r_j(t)$ denotes the downloading rate of ECN_j, so the reward of ECN_i at slot t could be calculated as:

$$R_i(t) = \psi_i(t) \cdot r_i(t). \quad (3)$$

C. Cost Model

The cost of ECN_i includes the energy cost and the downloading bandwidth cost:

1) Computation Cost: It is assumed that the CPU processing rate of ECN_i is normalized to the downlink transmission rate $r_i(t)$ to stabilize its buffer size, and according to [13], the power consumption is proportional to the square of normalized CPU processing rate. The energy cost of ECN_i at slot t is:

$$C_i^C(t) = \zeta k r_i^2(t), \quad (4)$$

where k and ζ are the proportion coefficient and price coefficient, respectively.

2) Bandwidth Cost: For downloading the tasks, the ECNs also need to pay to the BS for the transmission bandwidth, as often adopted in the MFG in caching works of Bennis et al. [14] [15]. Note that there is no downlink intra-cell interference since OFDM channels are assumed in this system, so denote the transmission power from the TA to each ECN_i by P_i, and denote the power of Additive white Gaussian noise by σ_n^2, the bandwidth $B_i(t)$ can be calculated through the Shannon theorem as $B_i(t) = r_i(t)/\log_2(1 + \frac{P_i h_i(t)}{\sigma_n^2})$, so denote the coefficient by ρ, the bandwidth cost at slot t is:

$$C_i^B(t) = \rho r_i(t)/\log_2(1 + \frac{P_i h_i(t)}{\sigma_n^2}). \quad (5)$$

To this end, the cost of ECN_i to fulfill the computation offloading task from TA during slot t can be calculated as:

$$C_i(t) = C_i^C(t) + C_i^B(t). \quad (6)$$

D. Problem Formulation

Each ECN_i needs to decide the optimal offloading rate $r_i(t)$ at the beginning of each slot t in a bounded action set $[0, r_i^{max}]$, based on the observation of the *state* $S_i(t)$. According to the analysis of reward and cost in Section II, the *state* of ECN_i should include the channel gain and the energy remained, which is defined as a two-dimensional variable:

$$S_i(t) = [h_i(t), E_i(t)]. \quad (7)$$

During a given time duration, based on the observation of the *state*, ECN_i tries to minimize the objective function:

$$\begin{aligned} &\min_{r_i(t)} \mathbb{E}\left[\sum_{t=0}^{T}(C_i(t) - R_i(t)) + F(E_i(T))\right] \\ \text{s.t.}\ \ &\Delta E_i(t) = -k r_i^2(t)\Delta t, \\ &\Delta h_i(t) = \alpha(t, h_i(t))\Delta t + \sigma_{h,i}\Delta W(t), \\ &E_i(0) = E_0, \end{aligned} \quad (8)$$

where $F(E_i(T))$ is the penalty function designed to penalize ECN_i for failing to make use of all the energy for offloading at the last slot T, which is defined as:

$$F(E_i(T)) = \phi e^{\epsilon E_i(T)} - \phi. \quad (9)$$

Here, $F(E_i(T)) = 0$, if $E_i(T) = 0$. If $E_i(T) > 0$, $F(E_i(T))$ takes a sufficiently large positive value for punishment. It is assumed that the initial energy $E_i(0)$, channel state $h_i(0)$, and the stochastic differential equations (SDE) parameters $\alpha(t)$ and σ_h in (1) are known at $t = 0$. Each ECN solves its version of the optimization problem in (8) at the same time, resulting in an N-user non-cooperative DSG.

III. Non-Cooperative Dynamic Stochastic Game (DSG) and Mean Field Theory

A. Derivation of The Optimal Strategy

To derive the optimal *strategy* to (8), the continuous-form *value function* [16] of ECN_i is given as follow:

$$v_i^*(t, S) = \min_{r_i(t)} \mathbb{E}\left[\int_t^T (C_i(u) - R_i(u))du + F(E_i(T))\right], \quad (10)$$

which captures the minimum cost ECN_i would get from slot t until T, under *state* $S_i(t)$. And the optimal running cost $v_i^*(t,S)$ is the unique solution to the HJB equation [17], which is given as follows:

$$\partial_t v_i^*(t,S) = -\min_{r_i(t)}[\partial_S v_i^*(t,S)\cdot\partial_t S + C_i(t) - R_i(t)], \quad (11)$$

according to (8), the Hamiltonian of (11) is defined as [18]:

$$\begin{aligned} &H(r_i(t), S_i(t), \nabla v_i^*(t,S)) = \min_{r_i(t)}[-kr_i^2(t)\partial_{E_i}v_i^*(t,S) \\ &+\alpha(t)\partial_{h_i}v_i^*(t,S) + \frac{\sigma_h^2}{2}\partial^2_{h_ih_i}v_i^*(t,S) + C_i(t) - R_i(t)], \end{aligned} \quad (12)$$

and since the optimal running cost trajectory $v_i^*(t,S)$ is the unique solution to the HJB equation [17], we take partial derivative with respect to $r_i(t)$ and equals it to zero, the optimal offloading strategy for ECN_i can be obtained as:

$$r_i^*(t) = \frac{\psi_i(t) - \frac{\rho}{\log_2(1+\frac{P_ih_i(t)}{\sigma_n^2})}}{2k[\zeta - \partial_{E_i}v_i^*(t,S)]}. \quad (13)$$

The Hamiltonian of the DSG (12) is easy to be proved smooth by checking the derivatives, which validates the existence of the Nash equilibrium for the DSG [19]. However, to find a Nash equilibrium for the N-user DSG requires solving N coupled HJB equations for each ECN_i [17], which makes it impractical when N grows large. To tackle the curse of dimensionality, we propose to transform the N-user non-cooperative DSG into a MFG to simplify the solving.

B. Transformation of DSG

According to [20], we consider the four following hypotheses (*H1-H4*) for the investigated computation offloading game:

- *H1-Rational expectations and behaviors of the players:* Each ECN individually makes a rational offloading decision to minimize the cost function
- *H2-Interchangeability of the states among the players:* ECNs sharing the same *state* would make the same decision, to this end, only one unique MFG-based strategy $r^*(t,S)$ needs to be adopted instead of N strategies, which greatly simplifies the game.
- *H3-The existence of the continuum of the players:* The presence of a large number of ECNs in the game ensures the existence of the continuum of the players.
- *H4-Social interaction between players through the mean field:* Each ECN interacts with the mean field game instead of interacting with all the other ECNs.

The DSG could be transformed into an MFG based on the hypotheses (*H1-H4*), and the mean field is defined as a statistical distribution of the *state* space. Denote $M(t,S)$ as the proportion of ECNs in *state* S at time-slot t, the probability density of ECNs in a specific *state* could be expressed as:

$$m(t,S) = \lim_{N\to\infty} M(t,S) = \lim_{N\to\infty}\frac{1}{N}\sum_{i=1}^{N}\mathbb{1}_{S_i(t)=S}, \quad (14)$$

where the indicator function $\mathbb{1}$ returns 1 if the condition is satisfied, and 0 otherwise.

In a mean field game, each player (ECN in this work) interacts with the mean field instead of interacting with all the other players separately. Since the ECNs' strategies influence each other on the price, a mean field price needs to be defined based on the MFG-based offloading strategy and the mean field $m(t,S)$, which is given as follows:

$$\Psi(t) \approx \phi\exp(-\xi\int_{\mathcal{H}}\int_{\mathcal{E}} m(t,S)r^*(t,S)\mathrm{d}h\mathrm{d}E), \quad (15)$$

where $\mathcal{H}$ and $\mathcal{E}$ are the value space of h and E, respectively.

With the mean field $m(t,S)$ and mean field price $\Psi(t)$ defined, the N-body DSG in (8) could be transformed into an equivalent 2-body MFG, where a generic player interacts with the mean field:

1) First body - A generic ECN: Fix all the other ECNs' strategies, one ECN can compute its optimal offloading strategy via the mean field HJB equation:

$$\partial_t v^*(t,S) = -\min_{r(t)}[\partial_S v^*(t,S)\cdot\partial_t S(t) + C(t) - R(t)], \quad (16)$$

2) Second body - The mean field: It could be deduced from (14) that the mean field is a collective of the other $N-1$ ECNs' strategies, whose evolution corresponds to the Fokker-Planck-Kolmogorov (FPK) equation [21]:

$$\begin{aligned} \partial_t m(t,S) &= \frac{1}{2}\sigma_h^2\partial^2_{hh}m(t,S) \\ &-\partial_h(\alpha(t)m(t,S)) + k\partial_E(r^2(t)m(t,S)). \end{aligned} \quad (17)$$

The mean field equilibrium (MFE) of this MFG is defined by the *value function* in (16) and the mean field in (17), and the optimal trajectory can be obtained by solving the two coupled equations iteratively, the iteration could be guaranteed converged since all the involved functions are smooth [22].

IV. Solving HJB Equation via Supervised Learning

FDM has been acknowledged as an effective approach to differential equations [8], but the obtained solutions are discrete and limited in differentiability [23]. So in this work, we implement supervised learning to solve the HJB equation by an ANN, which provides an approximate differentiable and closed analytic form solution.

As described in Section III, the optimal strategy of any ECN relies on the partial derivative of the *value function*, which is the solution to the HJB equation (16), and satisfies the following initial and boundary conditions:

$$\begin{cases} v = 0, \text{when } e = 0, \\ v = F(e), \text{when } t = T, \end{cases} \quad (18)$$

so according to [24], a trial solution v^{TR} could be defined to satisfy the initial and boundary conditions (18):

$$v^{TR} = \exp(t-T)\cdot F(e) + e(t-T)\vec{Z}, \quad (19)$$

where Z is the output of the designed neural network, which includes the weights for training, and the input of the neural network is the *state*, which is given by a three-column array:

$$\vec{X} = \begin{pmatrix} t_0 & e_0 & h_0 \\ t_0 & e_0 & h_1 \\ ... & ... & ... \\ t_0 & e_0 & h_{dim-1} \\ t_0 & e_1 & h_0 \\ ... & ... & ... \\ t_{dim-1} & e_{dim-1} & h_{dim-1} \end{pmatrix}, \tag{20}$$

where dim is the dimension of each domain of the *state*, each row of $\vec{X}$ represents a combination of t, e, and h. Assume the number of neurons in the hidden layer is y, then the weights matrix between the input layer and the hidden layer is:

$$\vec{W} = \begin{pmatrix} w_{0,0} & w_{0,1} & ... & w_{0,y-1} \\ w_{1,0} & w_{1,1} & ... & w_{1,y-1} \\ w_{2,0} & w_{2,1} & ... & w_{2,y-1} \end{pmatrix}, \tag{21}$$

so the hidden layer could be expressed as: $\vec{Y} = \vec{X} \cdot \vec{W} + \vec{b}$, where $\vec{b}$ is the bias term. Denote the weights from the hidden layer to the output layer as: $\vec{K} = (k_0, k_1, ..., k_{y-1})^T$, the output layer could be expressed:

$$\vec{Z} = \sigma(\vec{Y}) \cdot \vec{K} = \frac{1}{1 + \exp(-\vec{Y})} \cdot \vec{K}, \tag{22}$$

activate function $\sigma(\cdot)$ is fit to this work since whose derivatives could be expressed by the function itself:

$$\begin{cases} \sigma'(u) = \sigma(u) \cdot [1 - \sigma(u)], \\ \sigma''(u) = \sigma(u) \cdot [1 - \sigma(u)]^2 - \sigma(u)^2 \cdot [1 - \sigma(u)], \end{cases} \tag{23}$$

this property is important to represent the partial derivatives of the *value function*, note that the time-slot in the input layer, is timed by the first row of $\vec{W}$, and similar for e and h, so according to [24], the partial derivatives of the output could be expressed as:

$$\begin{cases} \frac{\partial \vec{Z}}{\partial t} = \vec{W}[0] \times \sigma(\vec{Y}) \times (1 - \sigma(\vec{Y})) \cdot \vec{K}, \\ \frac{\partial \vec{Z}}{\partial e} = \vec{W}[1] \times \sigma(\vec{Y}) \times (1 - \sigma(\vec{Y})) \cdot \vec{K}, \\ \frac{\partial \vec{Z}}{\partial h} = \vec{W}[2] \times \sigma(\vec{Y}) \times (1 - \sigma(\vec{Y})) \cdot \vec{K}, \\ \frac{\partial^2 \vec{Z}}{\partial h^2} = \vec{W}[2]^2 \times \sigma(\vec{Y}) \times (1 - \sigma(\vec{Y})) \times (1 - 2 \times \sigma(\vec{Y})) \cdot \vec{K}, \end{cases} \tag{24}$$

where $(\times)$ is the element-wise multiplication and $(\cdot)$ is the matrix multiplication, with (24), we can easily deduce the partial derivatives of the *value function* (19) with respect to t, e and h, respectively:

$$\begin{cases} \frac{\partial v^{TR}}{\partial t} = \exp(t - T) \cdot F(e) + e[\vec{Z} + (t - T)\frac{\partial \vec{Z}}{\partial t}], \\ \frac{\partial v^{TR}}{\partial e} = \exp(t - T) \cdot F'(e) + (t - T)[\vec{Z} + e\frac{\partial \vec{Z}}{\partial e}], \\ \frac{\partial v^{TR}}{\partial h} = e(t - T)\frac{\partial \vec{Z}}{\partial h}, \\ \frac{\partial^2 v^{TR}}{\partial h^2} = e(t - T)\frac{\partial^2 \vec{Z}}{\partial h^2}, \end{cases} \tag{25}$$

so the loss function to train the ANN is defined as:

$$loss = [\frac{\partial v^{TR}}{\partial t} - k(r^*(t))^2 \frac{\partial v^{TR}}{\partial e} + \alpha(t)\frac{\partial v^{TR}}{\partial h} + \frac{\sigma_h^2}{2}\frac{\partial^2 v^{TR}}{\partial h^2} + \zeta k(r^*(t))^2 + \frac{\rho r^*(t)}{\log_2(1 + \frac{Ph(t)}{\sigma_h^2})} - \Psi(t) r^*(t)]^2. \tag{26}$$

TABLE I

Parameters of Simulation

Parameter	Value
Noise Spectral Density	-174 dBm/Hz
Path Loss Exponent α	4
normalized maximum offloading rate	0.7
Reward steepness coefficient ξ	1
Penalty value coefficient ϕ	10
Penalty steepness coefficient ϵ	2
Energy consumption coefficient k	1
Energy price ζ	15
Bandwidth price ρ	1

V. Simulation

In this section, we present the simulation results of the proposed ANN, the dimension of the *state* (dim) is 10, and the number of neurons in the hidden layer (y) is 1500, the ANN is trained 500 steps with the learning rate set to 0.01. FDM is presented for comparison, whose convergence threshold is set to 5×10^{-4}. The other simulation parameters and their values are listed in Table I

A. Validation of the effectiveness of the proposed ANN

In this subsection, we present the training process of the proposed ANN, and results of *value function* solved by ANN and FDM are compared to validate the effectiveness and validity of the proposed ANN.

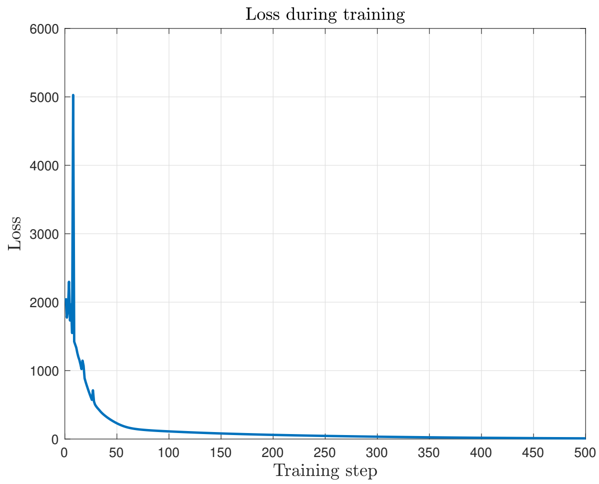
Fig. 2 (a) depicts the loss during training, we can observe that, with trial solution adopted, the loss converges quickly and drops to near 0 after about 300 training steps, which means that the network could be trained efficiently for fitting; Fig. 2 (b) and (c) depict the Optimal *value function* solved by ANN and FDM under a constant channel, respectively, both are very close in shape, which means the proposed network could be trained to solve the optimal *value function* effectively.

Fig. 3 compares the execution time for solving the HJB equation of ANN (not including the training) and FDM, which was carried on a 1.4 GHz Intel Core i5 CPU with 8GB memory. It could be observed that the execution time of ANN is much less than FDM, which shows an approximate linear growth trend with the growth of dim, while with FDM, the time increases quickly as dim increased, due to the nested loop statements. The comparison validates that, after the training, the proposed ANN could solve HJB more quickly, and the execution time increases slower than FDM, as the dimension of *state* increases.

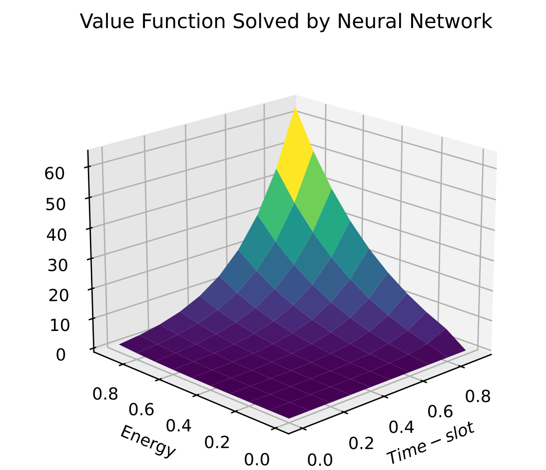
B. The performance of the MFG-based offloading strategy

Essentially, the *strategy* is a trade-off between the various kinds of utility, since the running cost is relatively objective, we present the impact of different penalty functions in Fig. 4, the exponential penalty is given by (9), and the logarithmic penalty is given as follow:

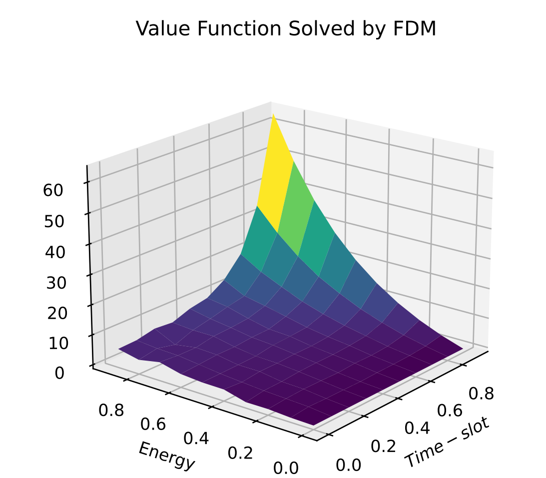
$$F^{(2)}(E_i(T)) = \frac{\phi}{1 + e^{-\epsilon E_i(T)}} - \frac{\phi}{2}. \tag{27}$$



(a) Loss during Training



(b) *Value function* Solved by ANN under A Constant Channel



(c) *Value function* Solved by FDM under A Constant Channel

Fig. 2. Fitting performance of the proposed Artificial Neural Network

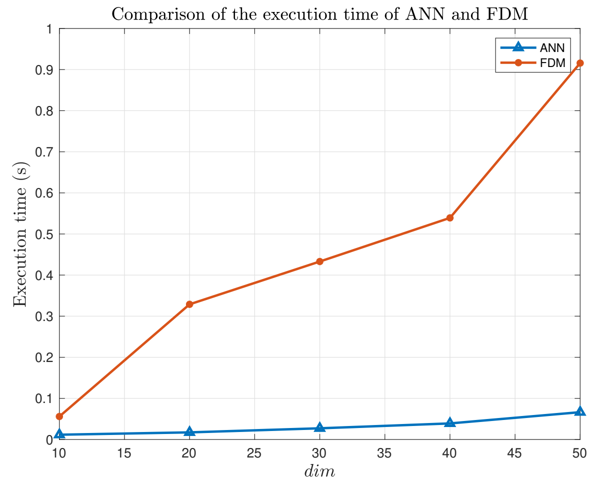


Fig. 3. Comparison of the execution time of ANN and FDM

It could be observed from Fig. 4 (a) that the offloading rate increases with both time and energy domain to avoid being punished. Contrarily, in Fig. 4 (b), the optimal offloading rate decreases with energy at the last time slot, which could be explained as follows: with little time available, the decreased value of the penalty is too small to cover the energy cost, a generic ECN will slow down the offloading rate. We can also deduce this result from (13), the optimal offloading rate is positively correlated to the partial derivative of the *value function* with respect to the energy, which decreases in a logarithmic penalty function.

Fig. 5 depicts the evolution of the optimal mean field distribution $m^*(t,e)$ under a constant channel. Since there is no need to take the derivative of the solution to the FPK equation, we solve the FPK equation via FDM, the discrete form of (17) under a constant channel is given as:

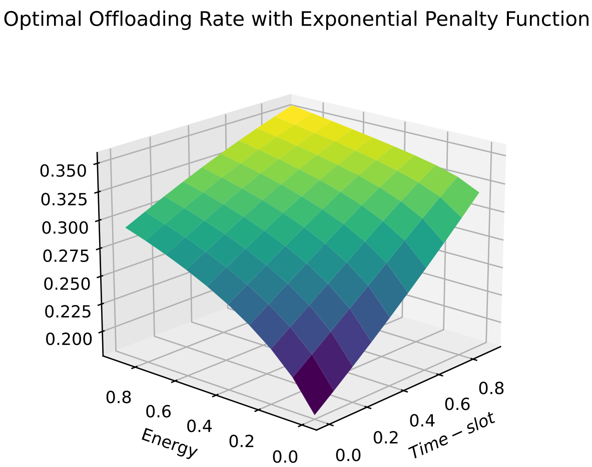
$$\begin{aligned}&\frac{m(t+\Delta t,e)-m(t,e)}{\Delta t}\\&=k\frac{r^2(t,e+\Delta e)m(t,e+\Delta e)-r^2(t,e)m(t,e)}{\Delta e},\end{aligned} \tag{28}$$

where $\Delta t=\Delta e=0.1$. The field is uniformly distributed at the beginning. It could be observed that the remaining energy decreases over time, and almost all the ECNs devote the whole energy for offloading, only a small proportion of them get punished for a little energy remained.

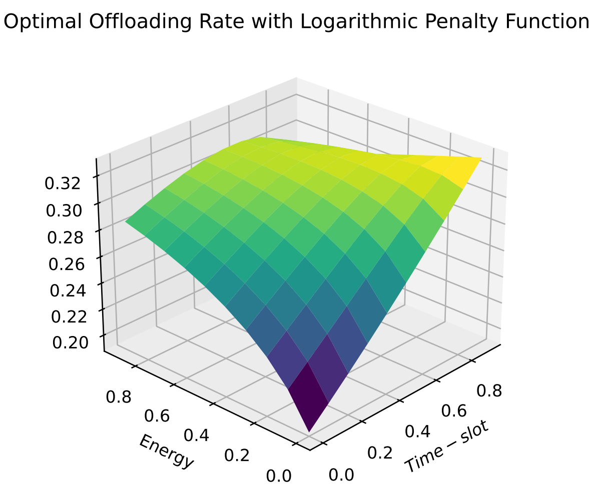
Fig. 6 (a) depicts the offloading rate with the mean field strategy and maximum rate strategy, $\alpha(t)$ is given as follow:

$$\alpha(t)=Af_0\cos\left(f_0 t\right), \tag{29}$$

where $A=0.5$ and $f_0=4\pi$, and the variance σ_h is $\frac{1}{12}$. We can observe that with the mean field strategy, the offloading rate almost remains unchanged when the channel gain is constant because the *state* is fully predictable under this scenario; when the channel gain follows an Ornstein-Unlenbeck process, the instantaneous offloading rate is positively correlated to the channel gain, which affects the bandwidth cost; and with



(a) Optimal Offloading Rate of Exponential Penalty



(b) Optimal Offloading Rate of Logarithmic Penalty

Fig. 4. Comparison of Optimal Offloading Rate solved by Exponential and Logarithmic Penalty

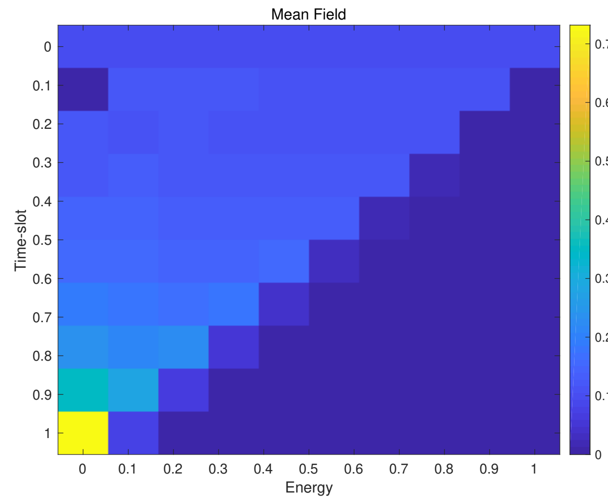


Fig. 5. Evolution of the optimal mean field distribution $m^*(t,e)$

maximum offloading strategy, the rate drops to zero quickly because the energy is consumed rapidly. The cumulated cost of a generic ECN is presented in Fig. 6 (b), it could be observed that with the mean field strategy under a constant channel, the cost increases almost linearly and achieves the minimum value, when the channel follows an Ornstein-Unlenbeck process, the cumulated cost is higher because of the unpredictability of the *state*. With maximum strategy, an ECN is not able to exploit the delay constraint and could not adapt to the variable channel, which results in maximum cost.

VI. Conclusion

In this paper, we investigate the optimal offloading strategy for ECNs in an ultra-dense network while taking into account the remained energy and variable channel state, which is formulated as a DSG and, transformed into an MFG to relax the computational complexity of the Nash equilibrium. To calculate the MFE, an ANN is designed to solve the HJB equation, and the simulation results reveal that the proposed ANN can solve the HJB equation efficiently, and the MFG-based strategy achieves lower cost than maximum offloading strategy.

References

[1] P. Mach and Z. Becvar, "Mobile Edge Computing: A Survey on Architecture and Computation Offloading," in *IEEE Communications Surveys Tutorials*, vol. 19, no. 3, pp. 1628-1656, thirdquarter 2017, doi: 10.1109/COMST.2017.2682318.

[2] J. Zhao, Q. Li, Y. Gong and K. Zhang, "Computation Offloading and Resource Allocation For Cloud Assisted Mobile Edge Computing in Vehicular Networks," in *IEEE Transactions on Vehicular Technology*, vol. 68, no. 8, pp. 7944-7956, Aug. 2019, doi: 10.1109/TVT.2019.2917890.

[3] F. Wang, J. Xu and Z. Ding, "Multi-Antenna NOMA for Computation Offloading in Multiuser Mobile Edge Computing Systems," in *IEEE Transactions on Communications*, vol. 67, no. 3, pp. 2450-2463, March 2019, doi: 10.1109/TCOMM.2018.2881725.

[4] H. Li, H. Xu, C. Zhou, X. Lü and Z. Han, "Joint Optimization Strategy of Computation Offloading and Resource Allocation in Multi-Access Edge Computing Environment," in *IEEE Transactions on Vehicular Technology*, vol. 69, no. 9, pp. 10214-10226, Sept. 2020, doi: 10.1109/TVT.2020.3003898.

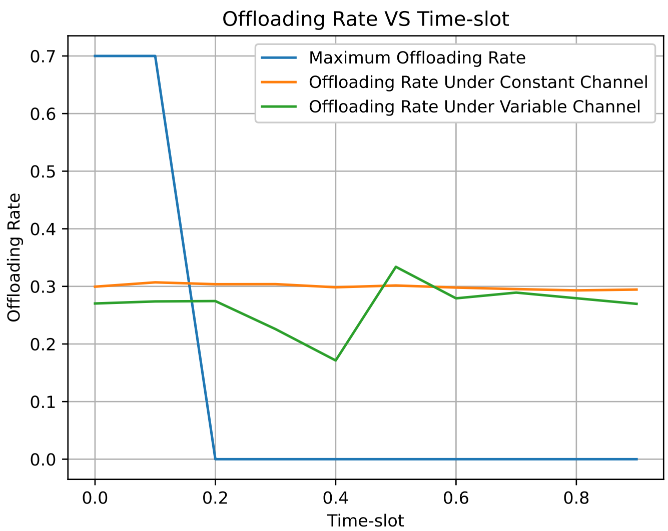
[5] M. de Mari, E. Calvanese Strinati, M. Debbah and T. Q. S. Quek, "Joint Stochastic Geometry and Mean Field Game Optimization for Energy-Efficient Proactive Scheduling in Ultra Dense Networks," in *IEEE Transactions on Cognitive Communications and Networking*, vol. 3, no. 4, pp. 766-781, Dec. 2017, doi: 10.1109/TCCN.2017.2761381.

[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," in *IEEE Transactions on Wireless Communications*, vol. 16, no. 2, pp. 1205-1217, Feb. 2017, doi: 10.1109/TWC.2016.2641959.

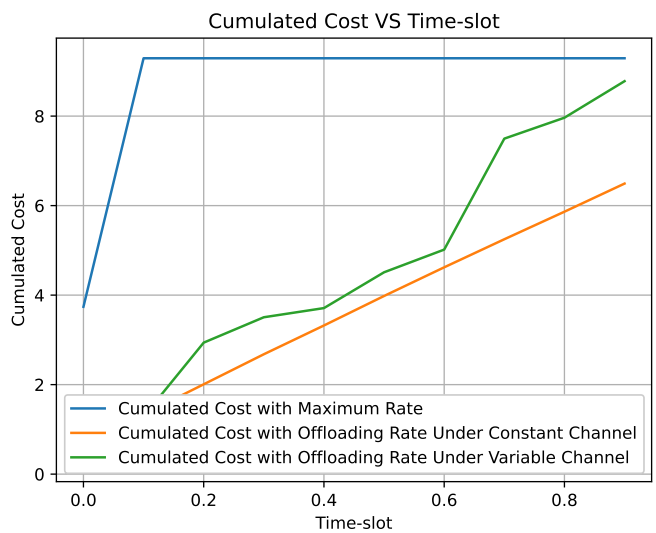
[7] R. A. Banez, L. Li, C. Yang, L. Song and Z. Han, "A Mean-Field-Type Game Approach to Computation Offloading in Mobile Edge Computing Networks," *ICC 2019 - 2019 IEEE International Conference on Communications (ICC)*, 2019, pp. 1-6, doi: 10.1109/ICC.2019.8761282.

[8] Grossmann C, Roos H G, Stynes M, "Numerical treatment of partial differential equations," in *Springer*, vol. 154, 2007.

[9] M. Medard and D. N. C. Tse, "Spreading in block-fading channels," *Conference Record of the Thirty-Fourth Asilomar Conference on Signals, Systems and Computers (Cat. No.00CH37154)*, 2000, pp. 1598-1602 vol.2, doi: 10.1109/ACSSC.2000.911259.



(a) Offloading Rate



(b) Cumulated Cost

Fig. 6. Offloading Rate and Cumulated Cost under Maximum Rate and Mean Field Rate

[10] Debbah M and Tembine H, "Reacting to the Interference Field," in *Mechanisms and Games for Dynamic Spectrum Allocation: Cambridge University Press*, 2013.

[11] R. Zheng, H. Wang and M. De Mari, "Optimal Computation Offloading with a Shared MEC Center: A Mean Field Game Approach," *2019 IEEE Globecom Workshops (GC Wkshps)*, 2019, pp. 1-6, doi: 10.1109/GCWkshps45667.2019.9024649.

[12] Buchanan J M, "The demand and supply of public goods," in *Rand McNally Chicago*, vol. 5, 1968.

[13] X. Meng, W. Wang, Y. Wang, V. K. N. Lau and Z. Zhang, "Delay-Optimal Computation Offloading for Computation-Constrained Mobile Edge Networks," *2018 IEEE Global Communications Conference (GLOBECOM)*, 2018, pp. 1-7, doi: 10.1109/GLOCOM.2018.8647703.

[14] H. Kim, J. Park, M. Bennis, S. Kim and M. Debbah, "Mean-Field Game Theoretic Edge Caching in Ultra-Dense Networks," in *IEEE Transactions on Vehicular Technology*, vol. 69, no. 1, pp. 935-947, Jan. 2020, doi: 10.1109/TVT.2019.2953132.

[15] Y. Jiang, Y. Hu, M. Bennis, F. Zheng and X. You, "A Mean Field Game-Based Distributed Edge Caching in Fog Radio Access Networks," in *IEEE Transactions on Communications*, vol. 68, no. 3, pp. 1567-1580, March 2020, doi: 10.1109/TCOMM.2019.2961081.

[16] Bellman R, "Dynamic programming and stochastic control processes," in *Information and control*, vol. 1, no. 3, pp. 228-239, 1958.

[17] P. Shige, "Stochastic hamilton–jacobi–bellman equations," in *SIAM Journal on Control and Optimization*, vol. 30, no. 2, pp. 284-304, 1992.

[18] P. Semasinghe and E. Hossain, "Downlink Power Control in Self-Organizing Dense Small Cells Underlaying Macrocells: A Mean Field Game," in IEEE Transactions on Mobile Computing, vol. 15, no. 2, pp. 350-363, 1 Feb. 2016, doi: 10.1109/TMC.2015.2417880.

[19] Başar T and Olsder G J, "Dynamic noncooperative game theory," in *Society for Industrial and Applied Mathematics*, 1998.

[20] Lasry J M and Lions P L, "Mean field games," in *Japanese journal of mathematics*, vol. 2, no. 1, pp.229-260, 2007.

[21] Burger M and Schulte J M, "Adjoint methods for hamilton-jacobibellman equations," in *Westfälische Wilhelms-Universität Münster*, 2010.

[22] Benaim M and Le Boudec J Y, "A class of mean field interaction models for computer and communication systems," in *Performance evaluation*, vol. 65, no. 11-12, pp. 823-838, 2008.

[23] Chiaramonte M and Kiener M, "Solving differential equations using neural networks," in *Machine Learning Project*, vol. 1, 2013.

[24] I. E. Lagaris, A. Likas and D. I. Fotiadis, "Artificial neural networks for solving ordinary and partial differential equations," in *IEEE Transactions on Neural Networks*, vol. 9, no. 5, pp. 987-1000, Sept. 1998, doi: 10.1109/72.712178.