When the User-Centric Network Meets Mobile Edge Computing: Challenges and Optimization

Langtian Qin, Hancheng Lu, and Feng Wu

The authors combine the user-centric network with MEC services and proposed a novel framework called user-centric MEC.

Abstract

As an emergent computing paradigm, mobile edge computing (MEC) can provide users with strong computing, storage, and communication services by moving the server to the user side. In recent years, applications such as virtual reality and augmented reality have brought higher requirements on transmission and computing capabilities. However, in the traditional cellular-based MEC, users at the edge of the cell will suffer severe signal attenuation and inter-cell interference, leading to a great reduction in achievable rate and proneness to transmission outage and offloading failure. To overcome this limitation, we combine the user-centric network (UCN) with MEC computing services and propose a novel framework called user-centric MEC (UCMEC). Through the dense deployment of access points, UCMEC can provide users with efficient, reliable, low-cost user-centric wireless transmission and edge computing services. To further exploit the benefits of UCMEC, we jointly optimize the task partition, transmit power control, and computing resource allocation decision to minimize the total energy consumption under delay constraints. Simulation results show that our proposed optimization scheme can bring users lower energy consumption and delay, and higher successful offloading probability than traditional MEC.

Introduction

The emergence of mobile edge computing (MEC) provides a new paradigm for data processing and computing [1]. MEC deploys the server on the "edge" side, for example, on base stations (BSs) of radio access networks. Users with limited computing resource can partition their computing tasks and offload part of them to the edge server for processing. Since the edge server is close to the user side, MEC can provide users with faster computing, communication, and storage services compared to cloud computing [2]. In applications such as virtual reality (VR) and augmented reality (AR), users need to transmit and process large amounts of real-time video data, which put forward high requirements on the transmission and computing abilities. However, in traditional cellular-based MEC, users at the edge of the cell suffer severe signal interference and have low wireless achievable rate. When these users offload their tasks to the edge server, the interruption outage probability greatly increases [3]. Thus, users can only process tasks locally, which consumes more energy and suffers higher delay.

As a key technology in 5G, the user-centric network (UCN) breaks the concept of "cell" in traditional cellular networks, which can significantly reduce the signal interference and provide users with higher achievable rates [4]. In the UCN, massive access points (APs) are deployed and serve a much lower number of users. Each user will be automatically allocated a set of APs to provide transmission services according to its geographical location, network condition, and business requirement, which is called an AP cluster [5]. A typical user needs to send data to all APs in the cluster in the form of radio frequency (RF) signals. After receiving the user's signals, APs send them to a central processing unit (CPU) through fronthaul links for processing. The CPU will decode multiple copies of signals from the user to recover the baseband data. Such a cooperative transmission mode can better offset the impact of channel fading and signal interference, and hence improve the throughput of the whole network [6]. The efficiency and reliability of the UCN motivate us to combine user-centric transmission mode with MEC computing service, namely user-centric MEC (UCMEC), to meet the high requirements on the transmission and computing abilities from emerging applications. In UCMEC, each user can transmit the offloading task data through a specific AP cluster. After that, a CPU integrated with an edge server decodes the signals and allocates computing resource to process the tasks. Therefore, UCMEC can provide efficient, reliable, low-cost transmission and computing services for all users wherever they are.

However, the combination of UCN and MEC is nontrivial, since the collaborative transmission mode of the UCN will directly affect the task partition, transmit power control, and computing resource allocation decision, thus affecting the overall efficiency of UCMEC. Deep reinforcement learning (DRL) combines the perception ability of deep learning with the explore ability of reinforcement learning, which can provide efficient solutions to the decision making problems in complex systems [7]. In this article, we propose a novel transmission and computing framework called UCMEC. In most existing works related to MEC [7–9], users must offload the tasks through a MEC-enabled BS, which is prone to suffer severe signal attenuation and inter-cell

Digital Object Identifier: 10.1109/MCOM.006.2200283

The authors are with the University of Science and Technology of China, China.

0163-6804/23/$25.00 © 2023 IEEE

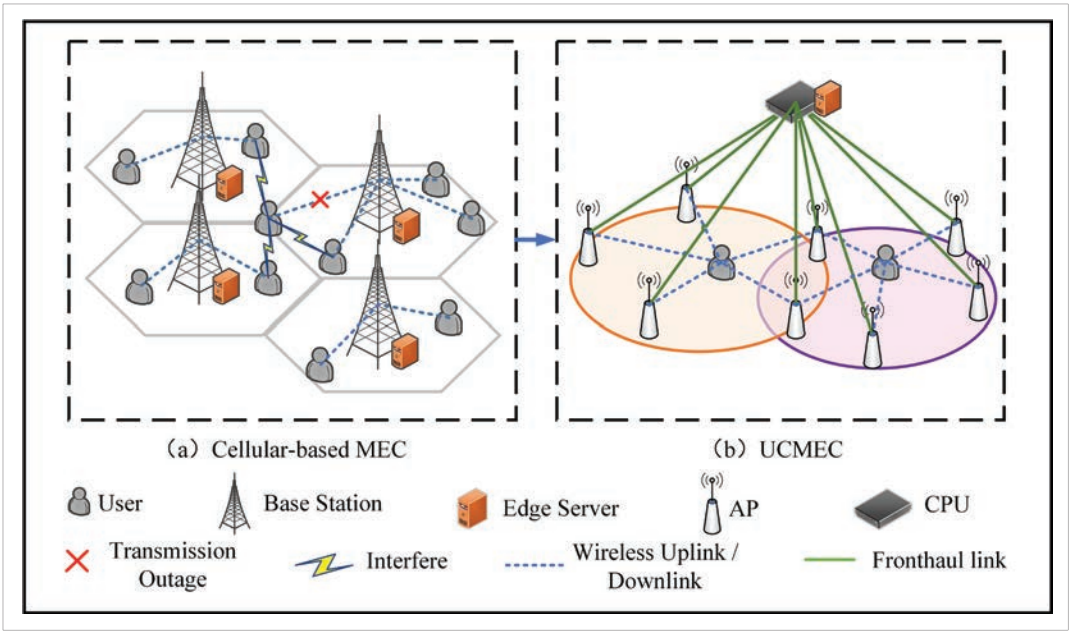

FIGURE 1. System architecture of a) cellular-based MEC; b) UCMEC.

After exploiting the challenges in UCMEC architecture, a DRL-based joint optimization scheme is proposed to minimize the total energy consumption by optimizing the task partition, power control, and computing resource allocation strategy. The simulation results demonstrate that compared to the traditional MEC, UCMEC architecture can significantly reduce the delay and energy consumption, as well as improve the successful offloading probability of users.

interference. By providing transmission services cooperatively through AP clusters and deploying a MEC-enabled CPU, UCMEC breaks the concept of "cell" in traditional cellular-based MEC and can provide users with reliable transmission and powerful computing services. After exploiting the challenges in UCMEC architecture, a DRL-based joint optimization scheme is proposed to minimize the total energy consumption by optimizing the task partition, power control, and computing resource allocation strategy. The simulation results demonstrate that compared to the traditional MEC, UCMEC architecture can significantly reduce the delay and energy consumption, as well as improve the successful offloading probability of users. In particular, the proposed optimization scheme can reduce the total energy consumption by up to 67.48 percent and the average total delay by up to 48.1 percent compared to the traditional MEC, and obtain the highest successful offloading probability under different parameters.

The rest of this article is organized as follows. The system architecture and key challenges of UCMEC are first described. Then the DRL-based joint optimization scheme is proposed to overcome these challenges, followed by the performance evaluation and analysis. Finally, we summarize this article and discuss future issues.

Architecture and Key Challenges of UCMEC

The comparison between the traditional cellular-based MEC architecture and UCMEC architecture is shown in Fig. 1. In Fig. 1a, each BS is integrated with an edge server and provides MEC services to all users within its coverage. When the user is at the edge of the cell, due to the long distance from the BS, the signal transmission suffers severe channel fading and inter-cell interference, thus substantially reducing the achievable rate. Therefore, wireless transmission of edge users is often prone to outage, resulting in MEC offloading failure. By breaking the concept of the traditional cell, UCMEC can provide users with efficient and reliable transmission and computing services wherever they are. As shown in Fig. 1b, network entities in UCMEC include users, APs, and the CPU integrated with an edge server. The connection between the user and the AP in UCMEC includes three stages:

- Uplink channel estimation
- Downlink data transmission
- Uplink data transmission

In the first phase, the users send pilot data in order to enable channel estimation at the APs. In the second phase, APs use channel estimations to perform channel-matched beamforming and send data symbols on the downlink to the user they served. Finally, in the third phase, users send uplink data symbols to the APs that sent downlink data symbols (i.e, the AP cluster). To reduce the total energy consumption and delay as well as to make full use of computing resource in UCMEC, there are still several challenges that need to be addressed.

Offloading Task Partition: In the traditional cellular-based MEC, users only need to send the task data to one BS. However, users in UCMEC need to send the offloading task data to all APs in the AP cluster at the same time. The increase of wireless connection will bring more interference and energy consumption. Therefore, users in UCMEC need to carefully determine the proportion of offloading tasks to reduce energy consumption on the premise of meeting the task delay requirements.

Transmit Power Control: During the offloading process, the transmit power needs to be carefully determined to ensure the transmission efficiency and meet the achievable rate fairness of users. Too low transmit power will lead to transmission interruption due to severe interference and noise. If the transmit power is too large, it will cause more energy consumption of the user. In addition, higher transmit power will bring greater interference to other users in the network, thus affecting the overall rate of the network. In UCMEC, since each user needs to send RF signals to multiple APs, more wireless connections will

In the traditional cellular-based MEC, the MEC server only allocates computing resource to users connect to the BS where the server is integrated. However, when users in UCMEC offload the tasks to the MEC-enabled CPU, the CPU needs to allocate the limited computing resource to all users; thus, the decision space for computing resource allocation will be larger.

make the transmit power control more complicated. Moreover, the transmit power control is closely related to the analysis of achievable rates, and the novel cooperative transmission mode makes the modeling and analysis of achievable rates completely different from traditional MEC, thus making the transmit power control in UCMEC more challenging.

Computing Resource Allocation: The amount of resource allocated by the CPU will directly determine the processing delay and the idle state energy consumption of users. In the traditional cellular-based MEC, the MEC server only allocates computing resource to users connect to the BS where the server is integrated. However, when users in UCMEC offload the tasks to the MEC-enabled CPU, the CPU needs to allocate the limited computing resource to all users; thus, the decision space for computing resource allocation will be larger. At the same time, the computing resource allocation in UCMEC needs to consider the impact of user-centric wireless transmission, which is more complicated than the computing resource allocation in traditional MEC.

Joint Optimization of Offloading and Resource Allocation

To address the challenges described above, we propose a DRL-based optimization scheme. The mathematic details of the system model and the proposed optimization scheme are available at https://github.com/qlt315/UCMEC_COMMAG. Consider a UCMEC system with M single-antenna users and N single-antenna APs. Each user has a computation-intensive task with a maximum delay tolerance constraint. Users can choose to process the tasks locally or offload the tasks to the CPU with more computing resource. In UCMEC, since the user is provided with wireless transmission services by a specific AP cluster, the user will suffer intra-cluster interference (interference caused by users sharing the same AP cluster with the user) and inter-cluster interference (interference from other users using different AP clusters). In this article, the users use orthogonal frequency-division multiple access (OFDMA) to connect with the AP cluster. Thus, the intra-cluster interference can be completely eliminated. For the wireless transmission, we use the three-segment path loss model to describe the large-scale and small-scale fading of the wireless channel [10]. The three-segment path loss model divides the path loss into three intervals according to the distance between the AP and the user. Compared to the traditional path loss model, the three-segment path loss model can describe the shadow fading of the wireless channel between the user and the AP more accurately. Since the channel between the user and the AP is equal to the square root of the large-scale fading coefficient multiplied by the small-scale fading coefficient, the path loss model and the radio access method directly determine the expression of user achievable rate, thus affecting the transmission delay. We assume the AP cluster of each user includes all APs whose channel gain with the user is greater than a certain threshold [11]. Therefore, the signals received by a typical AP can be expressed by multiplying the transmit power, the transmit symbol, the channel between the AP, and all users it services, and then add the channel noise. We assume that the CPU decodes signals by the large-scale fading decoding (LSFD) method. The main idea of LSFD is that the decoding vector and power coefficients depend only on the large-scale fading coefficients. Since these coefficients are independent of frequency and change slower than small-scale fading coefficients, compared to the traditional detection method, LSFD can significantly reduce the computational complexity in the CPU, which is very desirable in practical systems. After receiving the signal sent by APs, LSFD vectors are used to perform linear combining on local estimates to obtain the estimation of the signals [12]. Therefore, we can get the signal-to-interference-plus-noise ratio (SINR) of each user, which is the combination of the transmit power, LSFD vectors, and channel coefficients. Through SINR and system bandwidth, we can get the achievable rate of each user by the Shannon formula. In addition, according to the local computing resource, effective switching capacitance and task partition decision, power control decision, and computing resource allocation decision, we can obtain the delay and energy consumption expressions of local task processing, wireless transmitting, and edge task processing. To simplify the analysis, we assume that all APs are connected to the CPU through wired links which have sufficient capacities. However, our model can be simply extended to scenarios where the fronthaul delay and energy consumption are considered. Assuming that all APs are connected to the CPU through wireless or free space optical fronthaul with limited bandwidth, the fronthaul delay between a specific AP and the CPU can be obtained through the data size offloaded by all users served by the AP and Shannon's formula. Similarly, the energy consumption of fronthaul can also be obtained according to the fronthaul delay.

Since we mainly consider the energy consumption and latency for task offloading and wireless transmission, we ignore the energy use and latency of CPU processing. Moreover, to reflect the importance of wireless transmission to MEC service, we define the successful transmission probability as the proportion of users whose achievable rate is greater than the outage threshold. The successful computing probability is defined as the proportion of users whose delay is not greater than the maximum tolerance delay of their tasks. We define the successful offloading probability as the proportion of users who successfully transmit and compute at the same time.

Overview of the DRL-Based Optimization Scheme: Under the constraints of maximum transmit power of users, maximum computing capability of the CPU, and maximum tolerance delay of users, we optimize the task partition decision, power control decision, and computing resource allocation decision to minimize the total energy consumption. To tackle this NP-hard and non-convex problem, we seek the help of DRL. In DRL, an agent integrated with neural networks gets the current states by observing the environment, then takes actions, calculates the corresponding reward, and updates the parameters of neural networks. The agent can finally output the optimal strategy through cycling the above processes [13]. In this article, we propose a DRL-based

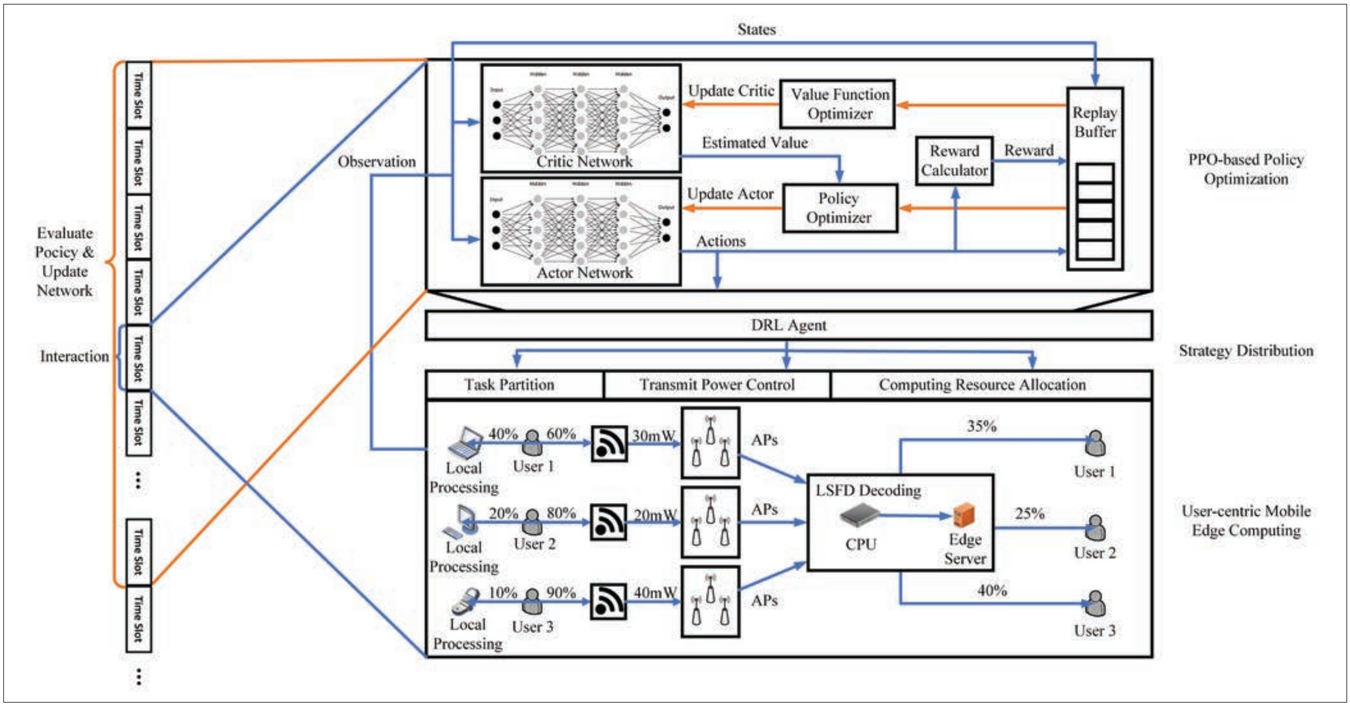

FIGURE 2. PPO-based DRL joint optimization scheme diagram.

optimization scheme to obtain the optimal task partition, power control, and computing resource allocation policy. The framework of the proposed scheme is shown in Fig. 2. The DRL agent consists of an actor network, a critic network, a replay buffer, a reward calculation module, a policy optimizer, and a value function optimizer. In each time slot, we define the state space as the SINR of each user. The action space is defined as the task partition, transmit power, and computing resource allocation decision. The reward consists of three parts: the total energy consumption of all users, the penalty item for the delay exceeding the maximum tolerance delay of each user, and the penalty item for the total allocated computing resource exceeding the total computing resource of the edge server. In each time slot (black lines in Fig. 2), the DRL agent observes the UCMEC system, obtains the SINRs of all users, and then inputs them into the actor and critic networks. The actor network outputs the current actions, while the critic network outputs the estimated value function based on the current observation. Then the reward at the current time slot is calculated according to the current states and actions by the reward calculation module. After that, the agent observes the environment of the next time slot, and stores the current states, actions, reward, and the next states in the replay buffer as a quadruple. After a period of time (orange lines in Fig. 2), the replay buffer sends the samples stored during this period to the policy optimizer and the value function optimizer to evaluate the current policy and update the parameters of the actor and critic networks, and then the buffer is cleared up. With the increase of training steps, the reward gradually increases and eventually tend to a relatively stable value. At this time, it can be considered that the agent has obtained the optimal decisions of task partition, transmit power control, and computing resource allocation.

Proximal Policy Optimization Method: To speed up the convergence of the DRL scheme, we use the proximal policy optimization (PPO) [14] method to optimize the continuity strategy. In the traditional DRL algorithm based on the policy gradient (PG), after updating the parameters by using the samples in the replay buffer, these historical samples are discarded. Therefore, the agent must re-interact with the environment to collect new samples, which is very time-consuming. In addition, in the later training steps of the PG, due to the slow learning rate, the policy will change greatly even when the policy parameters are updated slightly, making the algorithm hard to converge. In PPO, the updated policy rather than the original policy is used to interact with the environment to ensure that the samples can be reused. In addition, PPO introduces an importance sampling mechanism, which clips the deviation between the new policy and the original policy. When the deviation between the new policy and the original policy exceeds the threshold, the policy gradient becomes zero. The network parameters therefore cannot be updated, making the algorithm more stable.

Next, we prove the convergence of PPO. As a policy iteration-based DRL algorithm, PPO aims to find an optimal strategy. Assume that the value function space of PPO is a vector space and its metric is the infinite norm. It can be proved that the value function space is a complete metric space. Similarly, we can prove that the Bellman expectation equation of PPO is a contractive mapping. According to the contraction mapping theorem, it can be proved that the Bellman expectation equation of PPO can converge to a unique fixed point, so the strategy iteration of PPO will converge to a unique point at a linear rate [15].

Compared to other DRL algorithms, PPO has significant advantages. It does not rely too much on the design of reward function like soft

We assume that the standard deviation of shadow fading is 8 dB, the distance thresholds of the three-segments path loss model are 10 m and 15 m, and the channel gain threshold of UCN clustering is 0.4.

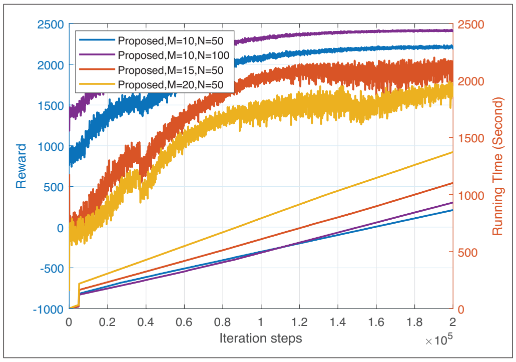

FIGURE 3. Iteration process and running time of the proposed scheme for different user and AP numbers.

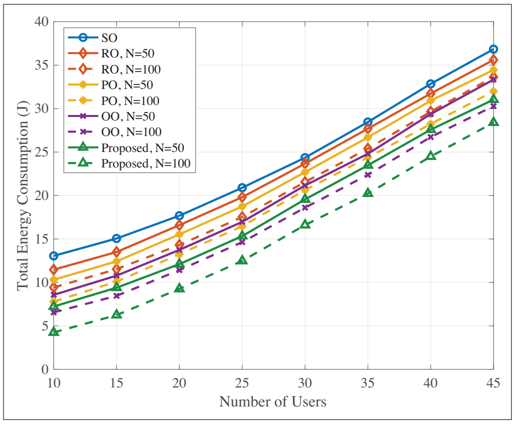

FIGURE 4. Total energy consumption vs. number of users for different AP numbers.

actor critic (SAC). In addition, it overcomes the shortcomings of the slow, unstable training process and complex parameter adjustment of deep deterministic policy gradient (DDPG). PPO has become the preferred baseline algorithm of leading artificial intelligence (AI) research institutions such as Deepmind and OpenAI due to its advantages of stability and simplicity.

Performance Evaluation

In this section, extensive simulations are conducted to validate the proposed optimization scheme in UCMEC. We assume that all APs and users are randomly distributed in a square area of 500 × 500 m^2. The computing resources of the edge server deployed on the CPU are 40 CPU cycles frequency in gigahertz, and the local computing resource of each user is evenly distributed in [0.6, 1] CPU cycles frequency in gigahertz. We assume that the maximum transmit power of each user is 100 mW, the effective switching capacitor is 10^{-27}, and the idle power is 10 mW. The user's computing task data size is evenly distributed in [0.05, 0.1] MB, the computing density is evenly distributed in [10000, 18000] CPU cycles/bit, and the maximum tolerance delay is evenly distributed in [1, 2] s. For the channel model, we assume that the noise power is –174 dBm/Hz, the carrier frequency is 1.9 GHz, the system bandwidth is 20 MHz, the antenna heights of APs and users are 15 m and 1.65 m, respectively, and the achievable rate threshold of user transmission outage is 5 Mb/s. We assume that the standard deviation of shadow fading is 8 dB, the distance thresholds of the three-segment path loss model are 10 m and 15 m, and the channel gain threshold of UCN clustering is 0.4. For the DRL algorithm, both the actor and critic networks have two fully connected hidden layers, and each hidden layer contains 64 neurons. The parameters of the neural networks are initialized orthogonally, and the generalized advanced estimator (GAE) method is adopted to calculate the policy gradient. In addition, the learning rate of actor and critic networks is initialized to 3×10^{-4}. We use the method of linear attenuation of the learning rate so that the learning rate decreases linearly from the initial value to zero with the increase of training steps. We set the discount factor to 0.95, the GAE parameter to 0.95, and the PPO clip parameter to 0.2. We set the replay buffer size and batch size to 2048 and 64, respectively.

To verify the effectiveness of the proposed PPO-based DRL optimization scheme, we consider the following four reference schemes:

- **Single Offloading (SO):** To verify the effectiveness of the user-centric offloading mode, we consider a small cell (as shown in Fig. 1a), where a BS integrated with an edge server serves all users.
- **Random Offloading (RO):** To verify the necessity of optimizing the task partition decision, users randomly select the offloading task proportion.
- **Optimal Offloading (OO):** To verify the necessity of optimizing the computing resource allocation decision, the CPU evenly allocates the computing resource to all users in this scheme.
- **Maximum Power Offloading (PO):** To verify the necessity of optimizing the transmit power decision, the transmit power of all users is taken as the maximum value.

We first evaluate the cost of the proposed PPO-based algorithm, including the number of iterations required for convergence and the running time. We implement the simulations using the Python 3.10 environment and Pytorch 1.12 on a computer with an AMD Ryzen 3600 CPU running on a processor speed of 3.6 GHz, and 16 GB RAM. It can be seen from Fig. 3 that when the number of users or APs increases, the amount of convergence and the running time of the algorithm will increase. This is because the decision space of the agent becomes larger and the number of system parameters increases, so the agent needs to interact with the environment more often to obtain a stable solution. In addition, the increase of the system scale makes the DRL algorithm more unstable, and the oscillation in the iterative process is greater. It should be noted that in each iteration, the agent needs to interact with the environment once, so the number of samples required by the algorithm is equal to the amount of convergence.

As shown in Fig. 4, we evaluate the impact of the number of users on the total energy consumption under different numbers of APs. It can be seen that when the number of users increases, the total energy consumption increases. In addition, when the number of APs increases, the total energy consumption decreases. This is because more intensively deployed APs will increase the number of APs serving each user, thus increasing the SINR of users as well as the achievable rate.

Therefore, users are more willing to offload tasks to the edge server. However, it should be noted that doubling the number of APs will not double the energy consumption reduction accordingly. The reason is that users need to send signals to more APs, resulting in larger energy consumption during transmission, thus offsetting part of the gain. From the figure, we can see that compared to other schemes, the proposed scheme can obtain the lowest energy consumption, which consumes 4.245 J of energy when the numbers of users and APs are 10 and 100, respectively. Compared to SO, the energy consumption is reduced by 67.48 percent.

Figure 5 shows that the delay performance varies with the number of users under different AP numbers. When the number of users increases, the average total delay increases. This is because newly added users occupy computing resource and affect the wireless transmission performance of other users, thus affecting the delay performance of all users. When the number of APs increases, the average total delay decreases. The reason is that when the density of APs increases, not only does the transmission delay decrease, but the successful offloading rate is increased too. Compared to other schemes, the proposed scheme can obtain the best delay performance, which consumes 0.9248 s when the number of users and APs are 10 and 100, respectively. The delay performance is improved by 48.1 percent compared to SO. This implies that the scheme can be applied to delay-sensitive scenarios.

To show the robustness of the proposed scheme, we evaluate the impact of different parameters on the successful offloading probability. We set the number of users, number of APs, and maximum transmit power to 10, 100, and 100 mW, respectively. In Fig. 6, we evaluate the impact of the maximum transmit power on the successful offloading probability. We assume that the computing resource of the edge server is 40 CPU cycles frequency in gigahertz. When the maximum transmit power increases, the successful offloading probability increases because the increase of the maximum power expands the solution space of the optimal power decision, and thus can obtain a better power allocation decision. It should be noted that when the maximum transmit power reaches between 80 mW and 100 mW, all schemes other than PO will not get more gain in successful offloading probability. Therefore, it can be inferred that the global optimal solution is between 80–100 mW. Since all users in PO transmit with the maximum power, when the power is too large, it has a negative effect on the achievable rate, thus reducing the success rate of offloading. It can be seen that with sufficient computing resource, the proposed scheme can achieve optimal performance under different maximum levels of transmit power.

According to the evaluation results, the UCMEC provider can obtain the time required to run the proposed algorithm, the delay and energy consumption of the user side under different user numbers, and the communication success rate under different levels of transmission power. Under specific delay and energy consumption constraints, the evaluation results can help the a UCMEC provider determine the deployment density and coverage radius of APs, the amount of MEC computing resource deployed in the service area, the maximum number of users allowed access, and so on.

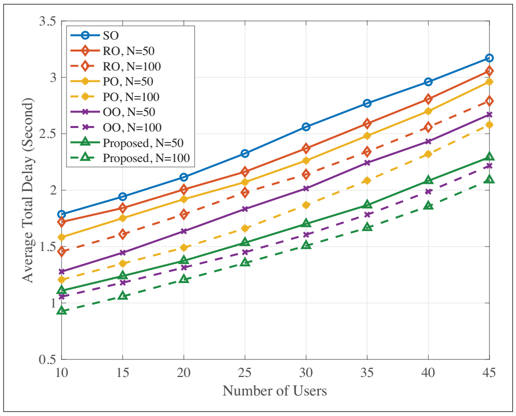

FIGURE 5. Average total delay vs. number of users for different AP numbers.

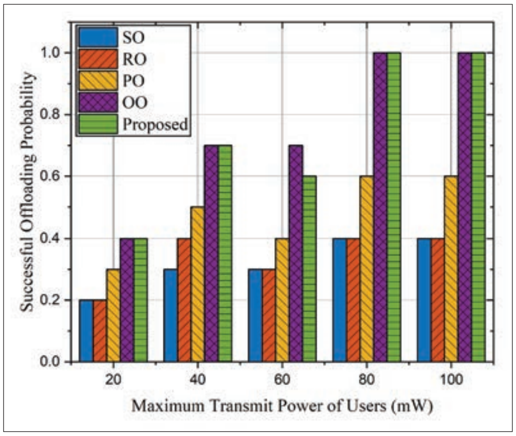

FIGURE 6. Successful offloading probability vs. maximum Transmit Power of users.

In UCMEC, by deploying a large number of APs, each user can transmit data through a specific AP cluster and offload tasks to the MEC-enabled CPU. To combat the challenges in UCMEC, a DRL-based optimization scheme is proposed to jointly optimize the offloading task partition, transmit power control, and computing resource allocation in UCMEC.

Conclusion and Future Work

In this article, we integrate the cooperative transmission method of UCN into MEC and propose UCMEC to overcome the lack of transmission resource in traditional MEC for users, especially those at the edge of the cell. In UCMEC, by deploying a large number of APs, each user can transmit data through a specific AP cluster and offload tasks to the MEC-enabled CPU. To combat the challenges in UCMEC, a DRL-based optimization scheme is proposed to jointly optimize the offloading task partition, transmit power control, and computing resource allocation in UCMEC. The simulation results verify the effectiveness of the proposed scheme. In particular, the proposed scheme can obtain a much lower energy consumption and delay, and ensure a higher successful offloading probability compared to traditional cellular-based MEC. In addition, from the simulation results, we can see that the gain brought by the deployment density of APs will gradually decrease. Considering the cost of deploying APs, UCMEC operators need to carefully determine the number and location of APs. Moreover, in the case that considers the overhead of the bandwidth-limited fronthaul, the size of the AP cluster needs to be limited, and the CPU deployment location and computing capability should be optimized to relieve the fronthaul pressure.

As an emergent architecture, there are still many open issues that should be further studied in UCMEC.

AP Deployment in UCMEC: According to the simulation results in the previous section, deploying more APs can bring a certain gain to energy consumption and delay, but the gain is decreasing with increasing AP density. In addition, deploying a larger number of APs will also bring more power and energy consumption. Therefore, it is necessary to investigate the optimal number of APs to maximize the system utility.

Fronthaul Overhead in UCMEC: In this article, we assume that the capacities of fronthaul links are sufficient, and ignore the transmission overhead of the fronthaul links. However, APs also need to receive the calculation results processed by the edge server from the CPU through the fronthaul links. With the increase of AP density, the communication between APs and the CPU will be more frequent, leading to a non-negligible fronthaul transmission overhead. Therefore, it is necessary to model and optimize the fronthaul communication process between CPU and APs to further ensure the low cost of UCMEC.

Security in UCMEC: In UCMEC, users need to transmit signals to multiple APs through wireless channels, which increases the risk of being attacked compared to that in traditional wireless networks due to the openness of the wireless channel. To ensure the security and privacy of users, effective security mechanisms need to be designed. In addition, the centralized architecture of deploying a single CPU in UCMEC is prone to a single point of failure. If the CPU is maliciously attacked, the entire network will be paralyzed. Therefore, it is necessary to design a new security framework or mechanism to ensure the stability and reliability of signal processing and decoding in the CPU.

Acknowledgments

This work was supported by the National Key R&D Program of China under Grant 2020YFA0711400, and the National Science Foundation of China under Grants U21A20452 and U19B2044.

References

[1] W. Shi *et al.*, "Edge Computing: Vision and Challenges," *IEEE IoT J.*, vol. 3, Oct. 2016.

[2] Q. Luo *et al.*, "Resource Scheduling in Edge Computing: A Survey," *IEEE Commun. Surveys & Tutorials*, vol. 23, no. 4, Aug. 2021, pp. 2131–65.

[3] S. Mukherjee and J. Lee, "Edge Computing-Enabled Cell-Free Massive MIMO Systems," *IEEE Trans. Wireless Commun.*, vol. 19, no. 4, Jan. 2020, pp. 2884–99.

[4] A. Ammar *et al.*, "User-Centric Cell-Free Massive MIMO Networks: A Survey of Opportunities, Challenges and Solutions," *IEEE Commun. Surveys & Tutorials*, Dec. 2021.

[5] C. Pan *et al.*, "User-Centric C-RAN Architecture for Ultra-Dense 5G Networks: Challenges and Methodologies," *IEEE Commun. Mag.*, vol. 56, no. 6, June 2018, pp. 14–20.

[6] H. Q. Ngo *et al.*, "Cell-Free Massive MIMO versus Small Cells," *IEEE Trans. Wireless Commun.*, vol. 16, no. 3, Jan. 2017, pp. 1834–50.

[7] P. Zhao *et al.*, "Energy-Saving Offloading by Jointly Allocating Radio and Computational Resources for Mobile Edge Computing," *IEEE Access*, June 2017.

[8] Y. Wang *et al.*, "Cooperative Task Offloading in Three-Tier Mobile Computing Networks: An ADMM Framework," *IEEE Trans. Vehic. Tech.*, vol. 68, no. 3, Mar. 2019, pp. 2763–76.

[9] L. Huang, S. Bi, and Y.-J. A. Zhang, "Deep Reinforcement Learning for Online Computation Offloading in Wireless Powered Mobile-Edge Computing Networks," *IEEE Trans. Mobile Computing*, vol. 19, no. 11, Nov. 2020, pp. 2581–93.

[10] S. Buzzi *et al.*, "User-Centric 5G Cellular Networks: Resource Allocation and Comparison with the Cell-Free Massive MIMO Approach," *IEEE Trans. Wireless Commun.*, vol. 19, no. 2, Feb. 2020, pp. 1250–64.

[11] H. A. Ammar *et al.*, "Downlink Resource Allocation in Multiuser Cell-Free MIMO Networks with User-Centric Clustering," *IEEE Trans. Wireless Commun.*, vol. 21, no. 3, Mar. 2022, pp. 1482–97.

[12] Z. Chen, E. Björnson, and E. G. Larsson, "Dynamic Resource Allocation in Co-Located and Cell-Free Massive MIMO," *IEEE Trans. Green Commun. and Networking*, vol. 4, no. 1, Nov. 2020, pp. 209–20.

[13] X. Zhu *et al.*, "A Deep Reinforcement Learning-Based Resource Management Game in Vehicular Edge Computing," *IEEE Trans. Intelligent Transportation Systems*, Sept. 2021, pp. 1–12.

[14] J. Schulman *et al.*, "Proximal Policy Optimization Algorithms," arXiv preprint arXiv:1707.06347, Aug. 2017.

[15] R. Sutton and A. Barto, "Reinforcement Learning: An Introduction," *IEEE Trans. Neural Networks*, vol. 9, no. 5, Sept, 1998, pp. 1054–54.

Biographies

Langtian Qin received his B.S. degree in information engineering from Xidian University, Xi'an, China, in 2021. He is currently pursuing a Master's degree with the Department of Electronic Engineering and Information Science (EEIS), University of Science and Technology of China (USTC), Hefei. His research interests include mobile edge computing, resource allocation in wireless heterogeneous networks, and blockchain.

Hancheng Lu [SM'19] received his Ph.D. degree in communication and information systems from USTC in 2005. He is currently an associate professor with the Department of EEIS, USTC. His research interests include resource optimization in wireless communication systems, such as RIS, NOMA, and cell-free, and caching and service offloading at wireless network edges.

Feng Wu [M'00, SM'06, F'13] received his B.S. degree in electrical engineering from Xidian University in 1992, and his M.S. and Ph.D. degrees in computer science from the Harbin Institute of Technology, China, in 1996 and 1999, respectively. He was a principal researcher and research manager with Microsoft Research Asia. He is currently a professor at USTC, and the Dean of the School of Information Science and Technology. His research interests include image and video compression, media communication, and media analysis and synthesis. He has authored or coauthored more than 200 high-quality papers. He serves as an Associate Editor of *IEEE Transactions on Circuits and System for Video Technology* and *IEEE Transactions on Multimedia*.