QoS-aware Task Offloading with NOMA-based Resource Allocation for Mobile Edge Computing

Luyuan Zeng, Wushao Wen, Chongwu Dong*
School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou, China
{zengly33}@mail2.sysu.edu.cn,{wenwsh,dongchw3}@mail.sysu.edu.cn

2022 IEEE Wireless Communications and Networking Conference (WCNC) | 978-1-6654-4266-4/22/$31.00 ©2022 IEEE | DOI: 10.1109/WCNC51071.2022.9771629

***Abstract*—Task offloading can scale the service capacity of IoT devices. However, IoT devices should go through wireless networks to connect with edge computing servers. The wireless network's performance can not be guaranteed in the dynamic scenario of the mobile environment. Devices would obtain diverse channel quality in frequency, time, and space, which is affected by many factors, such as selective channel fading and path loss fading. The network experience varies significantly between devices, even allocating the same amount of resources for all devices. Besides, too many tasks offloaded to one edge server simultaneously could exhaust the network resources between devices and base station and computing resources in the edge server. So, allocating the communication resource and determining task offloading among devices is a critical issue that should be considered appropriately and comprehensively. Aiming at this problem, we propose a QoS-aware task offloading strategy by decomposing the original problem into two sub-problems: bandwidth resource block allocation and task offloading scheduling. In our approach, the bandwidth resource allocation from the 5G network and task offloading scheduling between multiple edge servers in one edge cloud are jointly considered in two successive phases. Our strategy enables the acceleration of task computation by fine-grained management of network resources in real-time. Simulation results show that our algorithm significantly improves task offloading utility and improves the utilization of network symbol resource.**

***Index Terms*—Task Offloading, Resource Allocation, Game theory, Mobile Edge Computing, NOMA.**

I. Introduction

In the general mobile edge platform, offloading tasks should go through the 5G network. And, high efficiency of data transmission network utilization can improve the performance of task offloading, which can also reduce the operational cost for App vendors. So, bandwidth resource allocation, which is an effective tool to balance the wireless channel quality for UE, should obtain an optimal solution and be considered in detail. In the OFDM-based technology, the resource block (RB) is the smallest unit of network resource allocation, which is mainly determined by two indicators: subcarrier frequency and time slot. Bandwidth resources are often allocated in a resource block group (RBG), consisting of several RBs. Frequency division technology based on non-orthogonal multiple access (NOMA) is the core technology of spectrum sharing in 5g networks. In the NOMA scheme, one RB can be allocated to more than one user in a multiplexing mode. When the power domain is used to multiplex different users, it is called the power domain NOMA(PD-NOMA) [1], which would introduce internal interference inevitably by the power domain multiplex technology among users in the same RBG. So, optimizing RBG resource allocation in the uplink of NOMA is a critical issue to improve network utilization efficiency for task offloading. Practically, the arrival of tasks from different user equipment (UE) can not be predicted and may pour into an edge server in a small time interval. These tasks would accumulate in the backlog queue of an edge server. Due to the limited computing resource in edge servers from an edge cloud, task offloading scheduling among different edge servers should also be considered to ensure the QoS requirements for UEs.

To address the above challenges, we propose a QoS-aware Task Offloading (QTO) strategy to improve network resource utilization and ensure the QoS requirements including the metrics of time delay and energy consumption for all users. In our approach, the allocation of bandwidth resource block groups(RBGs) for the access network of UE and adaptive task offloading scheduling from UEs to a MEC server are jointly considered. The task completion time and energy consumption incurred by offloading can be satisfied in our strategy, while the network utilization could be significantly improved. The main contributions of this article can be outlined as follows.

- The QoS-aware task offloading problem is a combinatorial optimization problem with NP computational complexity. To reduce the complexity of this problem, our work decomposes it into two subproblems. These two sub-problems are RBG allocation for the access network of different UEs and dynamic task offloading between UEs and multiple edge servers, respectively.
- To handle two subproblems, we propose an optimization strategy considering multiplexing in the same RBG and task offloading scheduling among multiple edge servers, to obtain a tradeoff benefit between each other for the acceleration of task computation.

The remainder of this paper is organized as follows. We summarize related works in Section II. The system model and problem formulation is introduced in Section III. The QTO strategy is proposed in Section IV. Section V shows our simulation results, and finally, Section VI concludes this paper.

This work was supported by the National Natural Science Foundation of China (NSFC) under Grant U1711264. The corresponding author is Chongwu Dong (dongchw3@mail.sysu.edu.cn).

II. Related Work

With the fast deployment of the 5G network, task offloading through the wireless network has attracted significant attention in academia and industry. Some typical applications, such as IoT service [2], are considered to offload their tasks to MEC servers through the 5G network. So, wireless resource allocation is one of the critical issues that should be designed appropriately. Zhang, Qi et al. [3] considered subchannel and transmit power allocation in the process of task offloading to achieve energy efficiency optimization. Li, Yun et al. [4] proposed a joint strategy by combining the allocation of radio resources and the virtual machine resource together to ensure the quality of service (QoS) of a multimedia application. Fang, Fang et al. [5] proposed the optimal resource allocation for delay minimization and partial task offloading in NOMA-MEC Networks. Kuang, Zhufang et al. [6] concentrated on the joint optimization by partial offloading scheduling and power allocation in mobile edge computing. Huang et al. [7] proposed a joint strategy by task offloading and resource block (RB) allocation scheduler. But, they have not considered the character of RB multiplexing in the NOMA. Although some studies [8] have considered the feature of multiplexing resource allocation into the bandwidth allocation, these studies have not combined it with task offloading optimization.

As for task offloading optimization, recent studies have proposed lots of solutions based on a different theory, such as convex optimization theory [9], analytic hierarchy process (AHP) [7], reinforcement learning method [10] and GBD-based algorithm [11]. Unlike these work, a strategy based on the game theory with the queue tool is proposed in our work to solve the task offloading problem, which can be more practical and efficient.

To sum up, our approach can maximize the offloading utility of UEs and improve the network utilization for App vendors by jointly combining the bandwidth resource allocation and adaptive task offloading scheduling between UEs and MEC servers.

III. System Model and Problem Formulation

We consider a MEC system, shown in Fig.1, consisting of multiple UEs and one base station(BS) connected to an edge cloud consisting of various MEC servers. Each UE can access one MEC server through the wireless channel of the BS. We denote the set of UE as $\mathcal{N} = \{n_1, n_2, n_i..., n_n\}$. $\mathcal{M} = \{m_1, m_2, m_j..., m_m\}$ denotes the MEC servers.

A. Network and Communication Model

In the 5G network, NOMA is the core radio access technology in the uplink, utilized for network resource allocation. In the NOMA, the bandwidth frequency W can be divided into several RBGs. One RBG can be multiplexed by a limited number of UEs which are connected to the same BS. The allocation status of each RBG is denoted as a set $B = \{b_1, b_2, ..., b_r\}$, $1 \leq b_r \leq \xi$. $y_{i,r} \in \{0, 1\}$ is denoted as the indicator of RBG allocation for UE i. If RBG r is allocated to UE i, $y_{i,r} = 1$, otherwise, $y_{i,r} = 0$. According to the Shannon bound, the

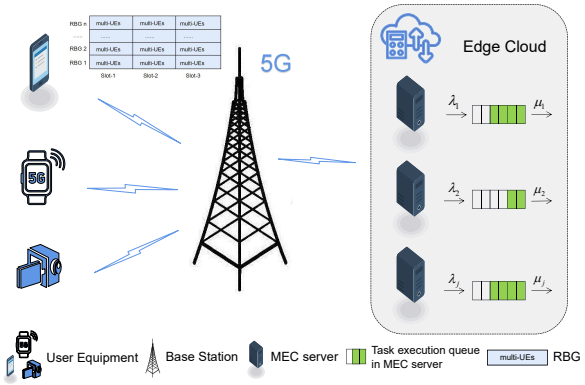

Fig. 1. MEC system.

upper limit of effective data transmission rate per Hz for UE i on RBG r can be represented as

$$R_{i,r} = log_2(1 + \frac{P_i|h_i|^2}{\sigma_2 + \sum_{j=1,h_j<h_i}^{\mathcal{N}} y_{j,r} P_j |h_j|^2}) \quad (1)$$

where P_i is the transmission power of UE i, h_i is the channel gain between BS and UE i, σ_2 is the background noise variance. The channel gain represents the UE's channel quality, which is influenced by several factors, e.g., the distance between UE and BS, the path loss, and small-scale fading in the wireless channel. In the uplink of NOMA, the transmission power is determined by UE. With the use of SIC technology, in the uplink, the UE signal with higher channel gain would be decoded first [12]. So, only the UE signal which is decoded last would have no interference. In short time intervals, the transmission power and channel gain of UE in the uplink are set consistently in all RBGs.

B. Computation Model

The offloading task of UE i is denoted as $\Psi_i = \{D_i, F_i\}$, where D_i is the data packet size, and F_i is the required number of CPU cycles for the task. The computation capacity of MEC servers can be denoted as $\mathcal{C} = \{c_1, c_2, c_3..., c_m\}$, which represents the number of clock cycles per second. We introduce the binary variables $x_{i,j}$ as offloading decision for UE i. $x_{i,j} = 1$, when UE i is associated with MEC server j for task execution. $\sum_j^{\mathcal{M}} x_{i,j} = 0$ can represent that UE i executes its task locally.

1) Local Execution Cost. The computation capability of UE i can be denoted as f_i^l, so the local execution time can be calculated as

$$T_i^l = \frac{F_i}{f_i^l} \quad (2)$$

The local execution energy comsuption of UE i can be calculated as

$$E_i^{exe,l} = \kappa (f_i^l)^2 F_i \quad (3)$$

where $\kappa = 10^{-27}$ is the effective switched capacitance depending on the chip architecture. The power consumption of the CPU is approximately proportional to the CPU frequency [13]. So, the energy comsumption per CPU cycle can be denoted as $\kappa(f_i^l)^2$.

Therefore, the cost of UE i from local task execution can be represented as

$$W_i^l = \omega_i T_i^l + (1-\omega_i) E_i^{exe,l} \tag{4}$$

where ω_i denotes the weight of energy comsumption and execution delay.

2) Offloading Execution Cost. Since a large number of tasks would be offloaded at the same time. MEC server needs to maintain a service queue for arrival tasks. According to the previous study, the task execution problem in the MEC server could be modeled as the M/M/1 queueing model.

Assume the offloading task arrival at MEC server j follows the Poisson process with the average arrival rate of λ_j (packets /second). Since the attribute of each task is different, we set a benckmark value F_{sta} for the required CPU cycles per packet. Based on the value of F_{sta}, the service rate of the queue μ_j (packets /second) can be calculated as $\mu_j = c_j / F_{sta}$. Then, the task execution time for UE i in MEC server j can be calculated as

$$T_{i,j}^{exe} = \frac{1}{\mu_j - \lambda_j} \tag{5}$$

where $\lambda_j = \frac{\sum_{i=0}^{\mathcal{N}} x_{i,j} F_i}{F_{sta}}$, which represents the number of tasks arrived at MEC server j per second after converting based on the F_{sta}. Therefore, the task execution time depends on the congestion level of the MEC server j. In addition, to ensure the stability of the MEC server, the utilization of the queue must maintain at $(\lambda_j/\mu_j < 1)$.

We assume UE i would offload the task to MEC server j. The offloading time delay for UE i can be calculated as

$$T_{i,j}^{off} = Ta_{i,j}^{trans} + T_{i,j}^{exe} + T_{i,j}^{re} \tag{6}$$

where $Ta_{i,j}^{trans}$ denotes the total transmission time from UE i to MEC server j. $Ta_{i,j}^{trans}$ includes waiting time for bandwidth RBG resource, which is determined by the complexity of RBG resource allocation algorithm. $T_{i,j}^{re}$ represents the transmission time for task execution result from MEC server j to UE i, and $T_{i,j}^{re}$ is tiny enough to be neglected.

And, the transmission power for UE i can be calculated as

$$E_{i,j}^{trans} = p_i T_{i,j}^{trans} \tag{7}$$

where $T_{i,j}^{trans}$ denotes transmission time from UE i to MEC server j, except waiting time for bandwidth RBG resource.

Therefore, the cost of UE i from task offloading can be represented as

$$W_{i,j}^{off} = \omega_i T_{i,j}^{off} + (1-\omega_i) E_{i,j}^{trans} \tag{8}$$

3) Utility Function of User. The utility function for UE i can be denoted as

$$U_i = \sum_{j=1}^{\mathcal{M}} x_{i,j} (W_i^l - W_{i,j}^{off}) \tag{9}$$

If $W_i^l - W_{i,j}^{off} < 0$, UE i would be no willing to offload its task to MEC server j, and $x_{i,j} = 0$. For UE i, $U_i > 0$ represents that offloading task to MEC server is more efficient, UE i would offload its task to MEC server when $U_i > 0$.

C. Problem Formulation

In the scenario, with the joint participation of UEs, MEC servers and one BS, we aim to maximize the whole utility of the system users, including energy consumption utility and time delay utility based on the QoS of UEs. Thus, the optimization problem can be modeled as

$$\max_{X,Y} \sum_{i=1}^{\mathcal{N}} U_i \tag{10}$$

$$s.t.\, C1 : x_{i,j} \in \{0,1\}, \forall i \in \mathcal{N}, \forall j \in \mathcal{M} \tag{11}$$

$$C2 : y_{i,r} \in \{0,1\}, \forall i \in \mathcal{N}, \forall r \in B \tag{12}$$

$$C3 : b_r \leqslant \xi, \forall b_r \in B \tag{13}$$

$$C4 : \lambda_j / \mu_j < 1, \forall j \in \mathcal{M} \tag{14}$$

$$C5 : \sum_{j}^{\mathcal{M}} x_{i,j} <= 1, \forall j \in \mathcal{M} \tag{15}$$

where $X \in R^{\mathcal{N} \times \mathcal{M}}$ is the offloading decision matrix. $Y \in R^{\mathcal{N} \times B}$ is the RBG resource allocation decision matrix. Generally, constraint C1 and C2 are offloading decision indicator and RBG allocation indicator, respectively. Constraint C3 guarantees the maximum multiplexing number per RBG. Constraint C4 ensures the stability of the queue. Constraint C5 ensures that each UE can select only one server for offloading or execute its task locally.

IV. Joint QoS-Aware Task Offloading and Resource Allocation Algorithm

According to the above analysis, the optimization problem in (10) has two optimization decisions X and Y. However, this optimization is difficult to be solved polynomially for the following reasons: first, the optimization problem is not convex due to that Y is a binary variable parameter. second, the computation complexity of the optimization problem is $O(2^{\mathcal{M} \times \mathcal{N} \times B})$, which would increase exponentially with $\mathcal{M}$, $\mathcal{N}$ and B. In this case, to reduce the computation complexity, we decompose the optimization problem into two subproblems and propose a two-phase strategy to approach the optimal solution for user offloading decisions.

A. Bandwidth Resource Allocation Subproblem

To reduce the transmission time for UEs and improve the utilization of network symbol resources. We propose an optimization strategy considering multiplexing in the same RBG for bandwidth resource allocation. UEs assigned to the same RBG would introduce the internal interfere, which could

affect the data transmission rate. Meanwhile, the channel gain between UEs is diverse. These characteristics will result in different RBG allocation policies with different data transmission rate for UEs and throughput for RBGs. In this case, we introduce the UE combination to represent a group of UEs allocated to the same RBG in our strategy. Thus, the goal of our strategy is to get all UE combinations that can maximize the network utilization. For example, UE combination allocated with RBG $r1$ and RBG $r2$ can be denoted as $Rc_{r1} = \{n_1, n_3, n_5\}$ and $Rc_{r2} = \{n_2, n_4, n_6\}$ respectively. n_1 represents UE 1. After UE combination adjustment between RBG $r1$ and RBG $r2$, Rc_{r1} and Rc_{r2} may be adjusted as $Rc_{r1} = \{n_1, n_4, n_5\}$, $Rc_{r2} = \{n_2, n_3, n_6\}$ to obtain higher effective transmission rate for RBG $r1$ and RBG $r2$. The object of bandwidth resource allocation subproblem can be represented as follows.

$$max \sum_{r=1}^{B} \sum_{i}^{\mathcal{N}} y_{i,r} R_{i,r} \tag{16}$$

Algorithm 1 RBG resource allocation combination optimization algorithm

Input: IterNum,
Initialization: $\mathbf{Y} \in \mathbb{R}^{\mathcal{N} \times B} = 0$
execute round-trip fair RB allocation to get initial RBG allocation result $\mathbf{Y}$
Get UE combination in each RBG: $RM \in \mathbb{R}^{B} \leftarrow Y$
while $iter \leqslant iterNum$ **do**
    Update $iter = iter + 1$.
    for all $i \in RM$ **do**
        for all $j \in RM$ **do**
            if $RM(i)', RM(j)'$ performs better **then**
                $candidate\_queue.append(RM(i)', RM(j)')$
            end if
        end for
    end for
    while $candidate\_queue \neq \emptyset$ **do**
        $pair = candidate\_queue.pop()$
        if pair does not contain RBG that already has been adjusted **then**
            execute the adjustment of the pair
        end if
    end while
end while

To maximize the above optimization, we proposed a strategy to adjust the UE combinations for each RBG. In our strategy, UE combination can first be initialized by a round-trip fair scheduler. Then, Each UE is allocated with a fixed number of RBGs. After that, the combination adjustment is executed in each iteration. In the process of UE combination adjustment, each UE combination would make a pair with each other. And, the pair of UE combinations that can obtain higher network utilization by mutually exchanging their items, would be put into the candidate queue. The candidate queue is sorted by the total effective transmission rate of each RBG pair in descending order. The item in the candidate queue is taken out for UE combination adjustment by orders. In our strategy, each UE combination in a RBG could be only adjusted once. If an item in the candidate queue includes the RBG which already has beed adjusted, the item would not to be exchanged. This process would be executed in several iterations to approach an optimal network utilization. Our strategy is described in Algorithm 1.

B. Task Offloading Based on Game Theory

Based on the bandwidth resource allocation subproblem, we can get the transmission time for each UE. Then, the task offloading subproblem can be written as

$$\max_{X} \sum_{i=1}^{\mathcal{N}} U_i \tag{17}$$

$$s.t. C1, C5 \tag{18}$$

It is a knapsack problem, which is NP-hard. To tackle this problem, we propose a QoS-aware game to solve the problem. In this game, UEs are acted as players to determine which MEC server they would choose for task offloading. In our QoS-aware game, the initial process of task offloading is first conducted to set the status of all UEs. Then, the gaming process between UEs is iteratively executed in several loops.

1) Initialization : Since service queue is considered in each MEC server, the average task execution time in each MEC server is determined by the number of offloading tasks. In the initial process, we assume the required CPU cycles of all tasks are the same, we can get the optimal number of offloading tasks per server to minimize the whole execution time in MEC servers. We formulate the initial process as a mixed-integer nonlinear programming, which can be represented as follows.

$$min \sum_{j=1}^{\mathcal{M}} \frac{z_j}{\mu_j - z_j} \tag{19}$$

$$s.t. V = \sum_{j=1}^{\mathcal{M}} z_j = min(\mathcal{N}, p \sum_{j=1}^{\mathcal{M}} \mu_j) \tag{20}$$

where $\mathcal{Z} = \{z_1, z_2, z_j ... z_m\}$ denotes the optimal number of offloading tasks per server. Since we have to maintain the stability of the queue, the total number of offloading tasks should not exceed $p \sum_{j=1}^{\mathcal{M}} \mu_j$. $p < 1$, which denotes a preferable queue capacity ratio for each MEC server. Due to the fixed number of MEC servers and limited number of UEs connected to the same BS, we can calculate $\mathcal{Z}$ in advance, and construct a mapping table, which contains user amount and corresponding result $\mathcal{Z}$.

According to $\mathcal{Z}$, the server selection in the initial process can be obtained in the following description. At first, we calculate each UE's utility when it offloads task to the MEC server with the least task execution time. Then sort $\mathcal{N}$ by the value of offloading utility in the ascending order, and select

V UEs to do task offloading. For V UEs, the user with less utility would be associated with a better MEC server.

2) Gaming Processing : In the Qos-based game, an initial offloading decision can be conducted in the previous method. After that, in each iteration process of the game, each UE would attempt to change decision alone to increase the whole utility. If a UE finds a better decision, it will send a decision update request to the competition pool. Then, the controller would choose a UE to update its decision in this iteration process. Eventually, the offloading decision profile would reach a Nash equilibrium when no UE can further increase the whole utility by unilaterally changing its association decision.

Algorithm 2 Qos-based Game for Task Offloading

Input: $\mathcal{N}, \mathcal{M}, T^{trans}, Ta^{trans}, \mathcal{C}, W^l \in \mathbb{R}^{\mathcal{N}}$,
get initial solution X for UEs' task offloading desicion
repeat
calculate the whole utility U under X using (10).
for all $i \; in \; \mathcal{N}$ **do**
    find the decision x_i' that incurs the highest U.
    if $x_i \neq x_i'$ **then**
        send x_i' to competition pool for decision update opportunity
    end if
end for
select UE i randomly from competition pool, then update its decision
until no more decision updates needed

C. Game Property

In this section, we investigate the existence of a Nash equilibrium in the QoS-aware game. The key is to prove that the QoS-aware game is potential. Since the object function (10) in our formulation is the sum of each UE's utility, which can be also represented as

$$\Phi = \sum_{i=1}^{\mathcal{N}} U_i \tag{21}$$

According to the study [14], we can get that (21) is the potential function. So, there must exist a Nash equilibrium for QoS-aware game. And, the Nash equilibrium of the QoS-aware game for UEs is $X^* = \{x_1^*, x_2^*, ...x_n^*\}$. At the Nash equilibrium, no user can further increase the whole utility by unilaterally changing its offloading decision, and the potential function should satisfy $\Phi_u(x_u, x_{-u}^*) \leq \Phi_u(x_u^*, x_{-u}^*)$.

V. Evaluation

In this section, we develop one event-based simulation based on the 5G toolbox in the matlab to evaluate the performance of the proposed QoS-aware Task Offloading (QTO) strategy.

A. experimental Setting

The channel model is based on the scheduling functionality of the medium access control (MAC) layer of the 5G New Radio (NR) stack proposed by 3GPP [15]. We consider a MEC system consisting of a BS, several UEs, and three MEC servers in the simulation. The coverage of the BS is 50m×50m. The background noise variance is -174 dBm/Hz, and the total bandwidth is 30 MHz. Other parameters used in the simulation are summarized in Table I.

TABLE I
SIMULATION PARAMETERS

Parameters	Values
Transmission power of UE(p_i)	{0.2,0.3,0.5,0.8}W
Task data size(D_i)	unif(7500,15000) bytes
Required CPU cycles of offloading task(f_i)	unif(1500,3000)
Computation capability of UE(c^{ue})	unif(0.8,3) Gcycles/s
Computation capability of MEC server(c^{mec})	{48,64,80} Gcycles/s
Distance between UE and BS(d_i)	unif(1,50) m

B. Network Utilization Comparison

To show the advantage of our strategy in network utilization, we introduce two bandwidth allocation strategies, the common RBG multiplexing without combination optimization strategy, and the RBG non-multiplexing strategy, to compare with our method called RBG multiplexing with combination optimization strategy. In the RGB non-multiplexing strategy, resource allocation would not consider the multiplexing of RBG.

Fig.2 shows the effective data transmission rate for each RBG in the above three transmission schemas. In the process of combination optimization, our strategy can differentiate the channel quality between different UEs allocated to the same RBG as much as possible. And especially, It would try to avoid UEs with poor channel quality to be allocated with the same RBG. As we can see, in our strategy, the data transmission rate in each RBG can be guaranteed with high stability and high efficiency compared with other strategies.

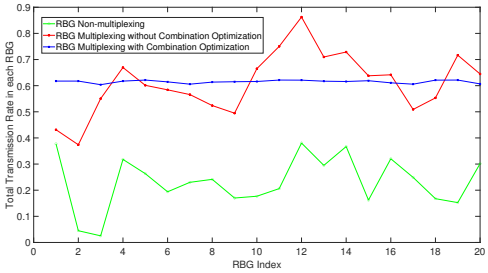

Fig. 2. Effective data transmission rate for each RBG.

As shown in Fig.3, the total effective transmission rate for all RBGs is slightly increased in our strategy, which would reduce the transmission time for UEs and improve the utilization of network symbol resource. In Fig.4, we compare the effective transmission rate for each UE in these three transmission schemas. And, we can find that, our strategy can ensure slightly better data transmission rate among all UEs, compared with the common RBG multiplexing strategy. To sum up, our strategy can ensure that each RBG can carry out the stable data volume and provide a reliable communication resource for UEs.

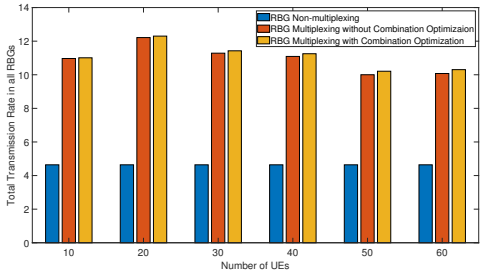

Fig. 3. The comparison of total data transmission rate for all RBGs.

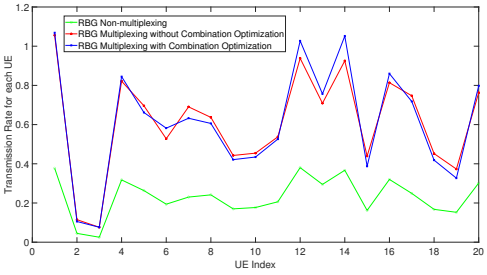

Fig. 4. Effective data transmission rate for each UE.

C. Utility Comparison

To verify the performance of the QoS-aware Task Offloading (QTO) strategy, we compare our proposed algorithm with three two-phase strategies. The random selection method and proportional selection method are both used to derive an initial solution for the Qos-based game phase. In the random strategy, UEs would be randomly scheduled to a MEC server or execute tasks locally. In the proportion-based selection method, the MEC server with higher computing capacity has a larger proportion for task offloading. That is to say, the MEC server with higher computing capacity is more likely to be chosen by UEs. RGB non-multiplexing and RGB Multiplexing are utilized to allocate RBG for UEs in another phase. RGB Multiplexing is applied in our strategy, while RGB non-multiplexing is used for comparison.

As shown in Fig.5, the utility gained by our strategy is always the highest among all strategies along with the number of users. In Fig.6, we can find that our strategy can make more UE offload their tasks to the edge server, which can reflect that our strategy can better utilize the network resource and computation resource.

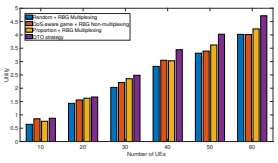

Fig. 5. The utility comparison.

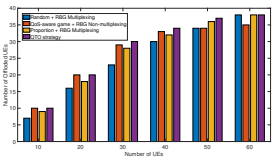

Fig. 6. The comparison of number of offloaded UEs.

VI. Conclusion

In this paper, our work proposed a QoS-aware task offloading strategy to maximize the utility of all UEs, which jointly considered the bandwidth resource allocation and adaptive task offloading decisions between UEs and multiple MEC servers. In our method, we first decompose the original NP-hard problem into two sub-problem: bandwidth RBG allocation and task offloading scheduling, respectively. Then, we propose a two-phase strategy based on combination optimization and Qos-based game for the bandwidth RBG allocation and task offloading scheduling. In our comparable simulations, the results verified the efficiency of the proposed scheme, which could ensure UEs' QoS requirements and maximize the utility of the user system.

References

[1] O. Maraqa, A. S. Rajasekaran, S. Al-Ahmadi, H. Yanikomeroglu, and S. M. Sait, "A survey of rate-optimal power domain noma with enabling technologies of future wireless networks," *IEEE Commun. Surv. Tutorials*, vol. 22, no. 4, pp. 2192–2235, 2020.

[2] Z. Wan, D. Xu, D. Xu, and I. Ahmad, "Joint computation offloading and resource allocation for noma-based multi-access mobile edge computing systems," *Computer Networks*, no. 2, p. 108256, 2021.

[3] Q. Zhang, L. Gui, F. Hou, J. Chen, S. Zhu, and F. Tian, "Dynamic task offloading and resource allocation for mobile-edge computing in dense cloud ran," *IEEE Internet Things J.*, vol. 7, no. 4, pp. 3282–3299, 2020.

[4] Y. Li, J. Liu, B. Cao, and C. Wang, "Joint optimization of radio and virtual machine resources with uncertain user demands in mobile cloud computing," *IEEE Trans. Multimedia*, vol. 20, no. 9, pp. 2427–2438, 2018.

[5] F. Fang, Y. Xu, Z. Ding, C. Shen, M. Peng, and G. K. Karagiannidis, "Optimal resource allocation for delay minimization in noma-mec networks," *IEEE Trans. Commun.*, vol. 68, no. 12, pp. 7867–7881, 2020.

[6] Z. Kuang, L. Li, J. Gao, L. Zhao, and A. Liu, "Partial offloading scheduling and power allocation for mobile edge computing systems," *IEEE Internet Things J.*, vol. 6, no. 4, pp. 6774–6785, 2019.

[7] X. Huang, Y. Cui, Q. Chen, and J. Zhang, "Joint task offloading and qos-aware resource allocation in fog-enabled internet-of-things networks," *IEEE Internet Things J.*, vol. 7, no. 8, pp. 7194–7206, 2020.

[8] B. Yang, L. Zhang, O. Onireti, P. Xiao, M. A. Imran, and R. Tafazolli, "Mixed-numerology signals transmission and interference cancellation for radio access network slicing," *IEEE Trans. Wireless Commun.*, vol. 19, no. 8, pp. 5132–5147, 2020.

[9] P. K. Korrai, E. Lagunas, A. Bandi, S. K. Sharma, and S. Chatzinotas, "Joint power and resource block allocation for mixed-numerology-based 5g downlink under imperfect csi," *IEEE Open J. Commun. Soc.*, vol. 1, pp. 1583–1601, 2020.

[10] 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. on Mobile Comput.*, vol. 19, no. 11, pp. 2581–2593, 2020.

[11] Y. Zhang, J.-H. Liu, C.-Y. Wang, and H.-Y. Wei, "Decomposable intelligence on cloud-edge iot framework for live video analytics," *IEEE Internet Things J.*, vol. 7, no. 9, pp. 8860–8873, 2020.

[12] C. Zhang, F. Jia, Z. Zhang, J. Ge, and F. Gong, "Physical layer security designs for 5g noma systems with a stronger near-end internal eavesdropper," *IEEE Trans. Veh. Technol.*, vol. 69, no. 11, pp. 13 005–13 017, 2020.

[13] W. Lin, T. Yu, C. Gao, F. Liu, T. Li, S. Fong, and Y. Wang, "A hardware-aware cpu power measurement based on the power-exponent function model for cloud servers," *Information Sciences*, vol. 547, pp. 1045–1065, 2021.

[14] D. Monderer and L. S. Shapley, "Potential games," *Games and economic behavior*, vol. 14, no. 1, pp. 124–143, 1996.

[15] T. Jiang, J. Zhang, P. Tang, L. Tian, Y. Zheng, J. Dou, H. Asplund, L. Raschkowski, R. D'Errico, and T. Jämsä, "3gpp standardized 5g channel model for iiot scenarios: A survey," *IEEE Internet Things J.*, vol. 8, no. 11, pp. 8799–8815, 2021.