Energy-Time Efficient Task Offloading for Mobile Edge Computing in Hot-Spot Scenarios

Fanfan Wu[1], Xiuhua Li[1,*], Hui Li[1], Qilin Fan[1], Linquan Zhu[2], Xiaofei Wang[3], and Victor C. M. Leung[4, 5]
[1]School of Big Data & Software Engineering, Chongqing University, Chongqing, China
[2]Chongqing Innovation Center of Industrial Big-Data Co. Ltd, Chongqing, China
[3]TKLAN, College of Intelligence & Computing, Tianjin University, Tianjin, China
[4] College of Computer Science & Software Engineering, Shenzhen University, Shenzhen, China
[5]Department of Electrical & Computer Engineering, The University of British Columbia, Vancouver, Canada
*Corresponding author: Xiuhua Li
Email:{ff.w,lixiuhua, h.li, fanqilin}@cqu.edu.cn, zhulinquan-casic@qq.com, xiaofeiwang@tju.edu.cn, vleung@ieee.org

ICC 2021 - IEEE International Conference on Communications | 978-1-7281-7122-7/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICC42927.2021.9500358

***Abstract*—Mobile edge computing (MEC) provides a new ecosystem that enables cloud computing capabilities at the edge of mobile networks, which is characterized by ultra-low latency and high bandwidth as well as real-time access to radio network information leveraged by applications. Nevertheless, various challenges, especially the decision-making issues for task offloading, are yet to be properly addressed. In this paper, leveraging the insight from the relative evaluation method, we propose a metric to quantify the benefit on users' service experience enhancement by task offloading. Meanwhile, by comprehensively considering the energy cost, time cost and users' service experience enhancement throughout the task offloading process, we formulate the task offloading decision-making problem as a two-dimensional knapsack loading problem to maximize the cost efficiency of task offloading. To solve the optimization problem more efficiently, we propose a suboptimal heuristic algorithm with polynomial-time complexity. Compared with four baseline algorithms, simulation results demonstrate the cost efficiency improvement of our proposed scheme.**

I. Introduction

Driven by the rapid development of new generation communication technologies (i.e., 5G and beyond 5G) and services, mobile network traffic is explosively growing [1], [2]. In addition, some application scenarios, such as unmanned driving and intelligent manufacturing [3], have higher requirements for the quality of service/experience (QoS/QoE) of mobile users such as lower delay, less energy consumption and so on. Consequently, to address the above challenges, mobile edge computing (MEC), a new network architecture that pushes cloud computing capabilities from the core network to the edge network, has received extensive attentions [4]. Compared with the traditional centralized network architecture, MEC can significantly improve users' QoS/QoE by providing services at the edges (e.g., base stations (BSs)) of mobile networks [5].

In recent years, plentiful studies focusing on MEC have emerged. Specifically, the scheme design for task offloading efficiently and profitably in MEC networks has been extensively investigated. For instance, to minimize the service time, the authors in [6] proposed an application workload allocation scheme to minimize the total response time of user equipment (UE) request, where both the network time and computing time were taken into account. The studies of [7] and [8] investigated the minimization of the energy consumption for task offloading in the MEC network, where a policy-based approach and the Lyaponuv optimization approach were used, respectively. In [9], the authors sought to maximize the number of the UEs served by the MEC network while satisfying the delay requirements, and proposed a game-theoretic approach. Besides, to achieve efficient resource utilization, the authors in [10] investigated the corresponding resource allocation issue and proposed a double-matching strategy based on deferred acceptance. However, these above studies have some limitations on the optimization design for task offloading, as they unilaterally tend to improve the interests of some participants in the MEC network, while ignoring the generated costs borne by other participants. In the MEC network, the participants mainly involve mobile operators, 3rd-party service providers and users. By offloading tasks to the edge network, users can obtain a better service experience, while it will incur system costs, most of which are borne by the operation developers and 3rd-party service providers. Particularly, the system costs mainly involve the energy cost for data transmission and task execution, and the time cost due to the occupation of edge servers (ESs) and communication links. Thus, it appears particularly important to comprehensively consider the interests of different participants and to ensure the high cost efficiency of the MEC network, which need to be further explored.

Moreover, there are few studies focusing on the optimization of task offloading for MEC in hot-spot scenarios with high concurrency task requests, where the dense task requests require a more efficient offloading scheme. Besides, due to the task heterogeneity, users' different emphasis on the energy consumption and service delay should be considered. More specifically, for these real-time services, e.g., cloud-based games and virtual reality [11], users are more concerned about getting lower service delay; meanwhile, in some service scenarios, e.g., model training and big data analysis [12], insufficient computing power and excessive energy consumption are the key issues.

In this paper, we are motivated to propose an energy-

time efficient task offloading framework for MEC in hot-spot scenarios. Particularly, the main contributions of this paper can be summarized as follows:

- We integrate the issues of the resources allocation of the MEC network and the heterogeneity of dense tasks in hot-spot scenarios, for practically offloading tasks and improving the cost efficiency, towards energy-time efficient MEC networks.
- We formulate the task offloading optimization problem as a constrained loading problem based on the comprehensive consideration of the costs of energy and time as well as the users' service experience enhancement. Then we propose a suboptimal scheme with polynomial-time complexity for engineering implementation.
- Simulation results demonstrate that our proposed scheme can improve the cost efficiency of task offloading significantly.

The remainder of this paper is organized as follows. Section II introduces the system model and problem formulation. Section III presents the scheme design and complexity analysis. Section IV evaluates the proposed scheme through simulation results. Finally, Section V concludes this paper.

II. System Model and Problem Formulation

A. Network Architecture

In hot-spot scenarios, the MEC network architecture is illustrated in Fig. 1. Of particular note is each BS is equipped with an ES, forming a BS-ES pair, which may be selected to serve a UE via a wireless link. We consider a set of $\mathcal{N} = \{1, 2, ..., N\}$ UEs and a set of $\mathcal{M} = \{1, 2, ..., M\}$ BS-ES pairs. The resource capacity of the BS-ES pair j is described by $R_j = \{R_j^B, R_j^C\}$, where R_j^B and R_j^C denote the corresponding communication resource capacity and the computing resource capacity, respectively. In the considered MEC network, we assume that the communication resource for wireless transmissions is quantized as the bandwidth, and the computing resource for executing tasks is quantized as the number of CPU cycles. Besides, the heterogeneous task of UE i can be described by $T_i = (s_i, w_i, \alpha_i, \beta_i)$, $i \in \mathcal{N}$. Therein s_i means the size of the data required to be transmitted, and w_i denotes the computing resource required to accomplish the task; α_i and β_i, satisfying $\alpha_i + \beta_i = 1$, denote the user's attention weights to energy consumption and service delay, respectively. To be specific, If a UE sends a task to the MEC network primarily to reduce the local energy consumption, the weights satisfy $\alpha > \beta$; otherwise, $\alpha < \beta$ holds.

In the considered MEC network, each task is indivisible and can only be offloaded to one ES and executed remotely. If the MEC network does not have enough resources for task offloading, the task can only be executed locally. Furthermore, for the UE that is only within the service coverage area of one BS, its task can be directly offloaded to the local ES, which is out of the scope of this paper. The real challenge to be solved in this paper, is the task offloading for UEs in the cross-coverage area (i.e., hot spot) of multiple BSs, which is highlighted in gray in Fig. 1. Thus, let $d_i \in \{0\} \cup \mathcal{M}$ denote the offloading decision of task T_i, $i \in \mathcal{N}$. Here, if $d_i = j$, it indicates that task T_i is offloaded to the BS-ES pair j; if $d_i = 0$, it means task T_i is forced to execute locally by UE i. Additionally, similar to [9], we consider the task offloading process to be quasi-static, where no new tasks arrive and no current tasks leave midway, and the network status does not change during the task offloading process.

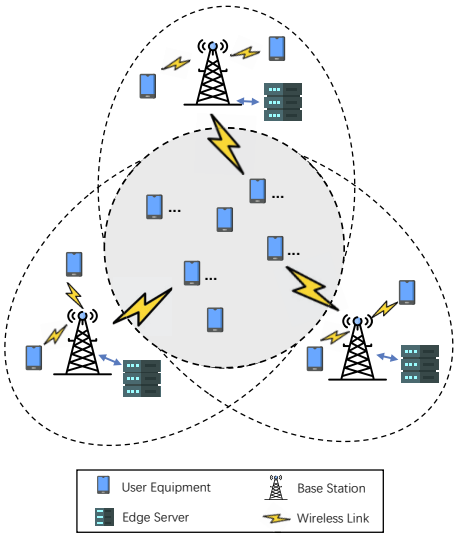

Fig. 1. An illustration of the MEC network architecture in hot-spot scenarios.

B. Energy and Time Model

Here, we will introduce the energy and time models in the two cases of task execution locally and remotely.

1) Local Execution Model: For UE i, c_i^L and e_i^L denote the local computing ability and the energy consumption for implementing a CPU cycle, respectively. Hence, the time consumption for the local execution of task T_i is

$$D_i^L = w_i / c_i^L. \tag{1}$$

Besides, the energy consumption of the local execution can be calculated as

$$E_i^L = w_i e_i^L. \tag{2}$$

2) Remote Execution Model: In the remote execution, the communication resource and the computing resource assigned to task T_i are denoted as ξ_i and σ_i, respectively. In different application scenarios, the communication resource of BSs and the computing resource of ESs are allocated among the offloaded tasks in different ways [9], [13], [14]. In this paper, considering the heterogeneity and fairness of tasks, the MEC network allocates resources proportionally according to the task type. More specifically, more communication and computing resources will be allocated to tasks with larger data sizes and higher computational requirements. Thus, in the case of $d_i = j$, the allocated resources of BS-ES pair j for task T_i can be calculated as

$$\xi_i = R_j^B \frac{s_i}{\sum\limits_{\forall i \in \mathcal{N} \wedge d_i = j} s_i}, \tag{3}$$

$$\sigma_i = R_j^C \frac{w_i}{\sum\limits_{\forall i \in \mathcal{N} \wedge d_i = j} w_i}. \tag{4}$$

Hence, in the case that task T_i offloaded to ES j, the uplink data rate $r_{i,j}$ can be calculated as

$$r_{i,j} = \xi_i \log_2(1 + \frac{p_i g_{i,j}}{N_0}), \tag{5}$$

where p_i and N_0 are the transmit power of UE i and the noise power, respectively; $g_{i,j}$ is the corresponding channel gain from UE i to BS j, which is constant during the offloading process. Similar to [15], it is assumed that the task execution output is in a small size so that the time consumption of feedback transmission can be negligible. Therefore, the time consumption consists of the elapsed time for data uploading and task execution. For the task T_i offloaded to ES j, the time consumption for task offloading is

$$D_{i,j}^R = s_i / r_{i,j} + w_i / \sigma_i, \tag{6}$$

meanwhile, the energy for the data transmission of UE i, denoted by $E_{i,j}^T$, is calculated as

$$E_{i,j}^T = p_i s_i / r_{i,j}. \tag{7}$$

Similar to (2), the energy consumption for task execution generated on ES j is $E_{i,j}^S = w_i e_j^S$, where e_j^S is the energy consumption for implementing a CPU cycle on ESs. Here, $e_j^S < e_i^L$ holds since ESs generally have higher computation energy efficiency than UEs.

As a result, the total energy consumption for accomplishing task T_i is calculated as

$$E_{i,j}^R = E_{i,j}^T + E_{i,j}^S. \tag{8}$$

C. Service Experience Enhancement Model

To quantify the service experience enhancement (hereinafter referred as to "buff") by task offloading, we propose a reasonable measurement metric leveraging the insight from the relative evaluation method [16].

Note that, to simplify the model, we only consider two important factors that affect the users' service experience: the energy consumption and the service delay. When evaluating the user's service experience in the case of $d \neq 0$, the service delay considered is the entire consumed time to obtain the task execution result, and the energy consumption considered is the energy consumption generated by the UE for data transmission, since users only care about themselves and don't care about the energy cost of task execution borne by ESs. Therefore, for task T_i with the allocation decision d_i, the buff value denoted by B_{i,d_i} is calculated as

$$B_{i,d_i} = \begin{cases} 1, & d_i = 0 \\ (\frac{E_i^L}{E_{i,d_i}^T})^{\alpha_i} \cdot (\frac{D_i^L}{D_{i,d_i}^R})^{\beta_i}, & d_i \neq 0. \end{cases} \tag{9}$$

Remark 1: Based on (9), when the UE executes its task T_i locally, i.e., $d_i = 0$, the UE obtains the value of buff as 1. Therefore, the tasks, offloaded to any ES with a buff value that is not greater than 1, should be executed locally for more efficient utilization of network resources.

D. Problem Formulation

In this paper, by comprehensively considering the interests of different participating roles in the MEC network, we take the cost efficiency of task offloading as the optimization goal. More specifically, we regard the energy and time consumed by task offloading and the generated buff for users as the costs and benefit, respectively. Thus, for task T_i with the allocation decision d_i, the cost efficiency of task offloading denoted by Ω_{i,d_i} can be calculated as

$$\Omega_{i,d_i} = \begin{cases} \frac{B_{i,d_i}}{\eta E_i^L + D_i^L}, & d_i = 0 \\ \frac{B_{i,d_i}}{\eta E_{i,d_i}^R + D_{i,d_i}^R}, & d_i \neq 0, \end{cases} \tag{10}$$

where η is a positive weight parameter. As a result, a higher Ω_{i,d_i} means more energy-time efficient.

Based on the above analysis, in the considered hot-spot scenario, we aim to maximize the average Ω for all tasks when making a task offloading decision $\mathcal{D} = \{d_i | i \in \mathcal{N}\}$ for tasks $\mathcal{T} = \{T_i | i \in \mathcal{N}\}$, and the corresponding optimization problem can be formulated as

$$\max_{\{d_i\}} \ \Omega = \frac{1}{N} \sum_{i \in \mathcal{N}} \Omega_{i,d_i} \tag{11a}$$

$$s.t. \quad \sum_{i \in \mathcal{N} \wedge d_i = j} \xi_i \leq R_j^B, \forall j \in \mathcal{M}, \tag{11b}$$

$$\sum_{i \in \mathcal{N} \wedge d_i = j} \sigma_i \leq R_j^C, \forall j \in \mathcal{M}, \tag{11c}$$

$$B_{i,d_i} > 1, \forall i \in \mathcal{N} \wedge d_i \neq 0, \tag{11d}$$

$$d_i \in \{0\} \cup \mathcal{M}, \forall i \in \mathcal{N}. \tag{11e}$$

Here, the constraints in (11b) and (11c) ensure that the resource requirements of the allocated tasks do not exceed the communication and computing resources that the BS-ES pair can provide, respectively. Besides, according to *remark 1*, the constraint in (11d) indicates that only the tasks, executed by ESs with a better service experience than the local execution, can be successfully offloaded.

III. Scheme Design and Complexity Analysis

Under the condition of limited communication and computing resources, the problem in (11) of finding an optimal task offloading decision is a special case of the two-dimensional knapsack loading problem (2D-KLP), which is NP-hard [17]. For solving the 2D-KLP, some algorithms with high computation complexity generally require massive iterations, e.g., game-theoretic approach [9], [15], which may result in a decrease in network management efficiency and incur a huge cost in algorithmic computation and state synchronization. In this section, in order to solve the problem in (11) efficiently, we propose a sub-optimal solution with polynomial-time complexity.

A. Greedy Heuristic Scheme for Task Offloading

From the perspective of practical engineering implementation, we propose a low-complexity greedy heuristic scheme to

maximize the cost efficiency. Particularly, the main procedures of the proposed scheme are briefly described as follows:

- *Step* 1: BS-ES pair selection. In this step, the BS-ES pair with the lightest workload is selected for the task to be offloaded.
- *Step* 2: Task classification. The buff value will be calculated for all the unallocated tasks. According to *remark 1*, tasks will be divided into two categories: execution locally and execution remotely, which are denoted by Φ^L and Φ^R, respectively.
- *Step* 3: Task prioritization and allocation. For tasks that are requested to be executed in the MEC network, their priorities are determined synthetically by the energy cost, time cost and buff brought by task offloading. The task with the highest priority will be allocated to the BS-ES pair selected in *Step* 1. After the highest priority task and its offloading decision are determined, the network status information will be updated.

The above steps will be executed in a loop, and the loop will stop until the network has no idle resources or all tasks are allocated. Next, the details of each step are described as follows.

1) BS-ES pair selection: In the design of the scheme, in order to provide users with better service experience and ensure the load balance of the network, the BS-ES pair with the lightest workload is selected to run the offloaded task. Hence, the assigned BS-ES pair index is optimized as

$$\chi = \arg\min_{j\in\mathcal{M}}\{\frac{R_j^{C-load}}{R_j^C}\cdot\frac{R_j^{B-load}}{R_j^B}\}, \tag{12}$$

where $R_j^{C-load} = \sum_{i\in\mathcal{N}\wedge d_i=j} w_i, \forall i \in \mathcal{N}$, and $R_j^{B-load} = \sum_{i\in\mathcal{N}\wedge d_i=j} s_i, \forall i \in \mathcal{N}$, represent the computing load and the communication load of BS-ES pair j, respectively.

2) Task classification: After the BS-ES pair is selected, according to its state information and (9), we can get the expected buff of each task if it is offloaded to the BS-ES pair χ. On the basis of *remark 1*, the tasks can be divided into two categories, execution locally and execution on the ES χ, denoted by Φ^L and Φ^R, which are defined as

$$\Phi^L = \{i|B_{i,\chi} \le 1, i \in \mathcal{T}^u\}, \tag{13}$$

$$\Phi^R = \{i|B_{i,\chi} > 1, i \in \mathcal{T}^u\}. \tag{14}$$

3) Task prioritization and allocation: In order to make the utilization of network resources more effective and valuable, those tasks that can achieve a good balance between task offloading costs and benefits should have higher allocation priorities. In this paper, the priority definition jointly takes the energy cost, time cost and buff into consideration. Ultimately, the priority of task T_i is calculated as

$$P_i = (r_i^E)^{\alpha_i}\frac{\Delta E_i}{\sum_{i\in\Phi^R}\Delta E_i} + (r_i^{\mathrm{D}})^{\beta_i}\frac{\Delta D_i}{\sum_{i\in\Phi^R}\Delta D_i}, \tag{15}$$

Algorithm 1 Iterative Greedy Heuristic Algorithm.

Input:
Task set: $\mathcal{T} = \{T_i|i\in\mathcal{N}\} = \{(s_i, w_i, \alpha_i, \beta_i)|i\in\mathcal{N}\}$;
Resource capacity set: $\mathcal{R} = \{R_j|j \in \mathcal{M}\} = \{(R_j^B, R_j^C)|j\in\mathcal{M}\}$.
Output: Decision set: $\mathcal{D} = \{d_i|i\in\mathcal{N}\}$.
Initialization:
Unallocated task set: $\mathcal{T}^u = \mathcal{T}$;
Offloading decision: $d_i = 0, \forall i \in \mathcal{N}$;
Resource load: $R_j^{C-load} = R_j^{B-load} = 0, \forall j \in \mathcal{M}$.

repeat
  —*Step* 1. [BS-ES pair selection]
  Select the BS-ES pair χ, where $\chi = \arg\min_{j\in\mathcal{M}}\{\frac{R_j^{C-load}}{R_j^C}\cdot\frac{R_j^{B-load}}{R_j^B}\}$.
  —*Step* 2. [Task classification]
  Set $\Phi^R = \emptyset$.
  Set $\Phi^L = \emptyset$.
  for each $T_i \in \mathcal{T}^u$ **do**
    Calculate the $B_{i,\chi}$ according to (9).
    if $B_{i,\chi} > 1$ **then**
      Set $\Phi^R = \Phi^R \cup \{i\}$.
    else
      Set $\Phi^L = \Phi^L \cup \{i\}$.
    end if
  end for
  if $\Phi^R = \emptyset$ **then**
    return $\mathcal{D}$.
  end if
  —*Step* 3. [Task prioritization and allocation]
  for each $T_i, i \in \Phi^R$ **do**
    Calculate the priority P_i according to (15).
  end for
  Select the task T_φ, where $\varphi = \arg\max_{i\in\mathcal{N}}\{P_i\}$.
  Set $d_\varphi = \chi$.
  Set $\mathcal{T}^u = \mathcal{T}^u - \Phi^L - \{d_\varphi\}$.
  Set $C_\chi^{load} = C_\chi^{load} + w_\varphi$.
  Set $B_\chi^{load} = B_\chi^{load} + s_\varphi$.
until $\mathcal{T}^u = \emptyset$.
return $\mathcal{D}$.

where, from the perspective of system benefits, $r_i^E = E_i^L/E_{i,\chi}^T$ and $r_i^D = D_i^L/D_{i,\chi}^R$, represent the ratio of the energy consumption of the UE and the service delay when the task is executed locally and remotely, respectively; from the perspective of system costs, ΔE_i and ΔD_i represent the difference in the energy cost and time cost between the local execution and remote execution, respectively, which can be calculated as

$$\Delta E_i = E_i^L - E_{i,\chi}^R, \tag{16}$$

$$\Delta D_i = D_i^L - D_{i,\chi}^R. \tag{17}$$

Then the task T_ψ with the highest priority can be offloaded

to the ES χ, where the ψ is expressed as

$$\psi = \arg\max_{i \in \mathcal{M}} \{P_i\}. \tag{18}$$

After the task offloading decision-making is completed, the network resource information will be updated. The algorithm will proceed to the next loop until each task has been allocated or the network has no idle resources. At last, the detailed process of our proposed scheme is given in **Algorithm 1**.

B. Complexity Analysis

Theorem 1: The time complexity of implementing the proposed **Algorithm 1** is polynomial.

Proof: During each loop in **Algorithm 1**, for *Step* 1, the BS-ES pair selection consumes M iterations to iterate over the set $\mathcal{M}$. For *Step* 2, the classification process has not greater than N iterations for unallocated task set $\mathcal{T}^u$, which is a subset of $\mathcal{T}$. In the last *Step* 3, for each task in Φ^R, the priority calculation process requires $|\Phi^R|$ iterations, which is also not greater than N. In view of the fact that, in the *Step* 2 of each loop, part of the unallocated tasks will be assigned to Φ^L, which need not be processed in the next loop. Hence, there are no more than N loops are required throughout the task offloading process. Overall, the iterations consumed in the entire process of **Algorithm 1** is less than $(M+N+N)\cdot N$. Therefore, the time complexity of implementing the proposed **Algorithm 1** is polynomial as $O(2N^2 + N \cdot M)$.

IV. Numerical Results

In this section, we conduct simulation experiments to evaluate the performance of the proposed scheme. For simulation purpose, the experiment parameters are set as follows: M = 3, and each BS's bandwidth is 10 MHz where the deployed ESs have the computing ability of 3 GHz, 6 GHz, 9 GHz, respectively. For the tasks, the data size and required computing resource follow the random distribution with $s_i \in [300, 800]$KB and $w_i \in [0.1, 1]$GHz [13], respectively; and the weights α and β are randomly distributed in $[0, 1]$. Besides, the transmission power p_i of each UE is set as 0.5w and the ratio of $g_{i,j}$ to N_0 is fixed at 8Hz/w [18].

Moreover, we consider various scenarios by changing two experiment parameters: 1) the number of UEs; 2) the value of weight η. Each experiment is repeated for 50 times and the results are averaged. Furthermore, there are four baselines for performance comparison as follows:

- Greedy [9]: In *Step* 3, the task with the greatest Ω is allocated first.
- Random [9]: Randomly allocate tasks to a BS-ES pair that has adequate available resources.
- ED-Focus: In *Step* 3, tasks with less energy saving and time reduction have higher priorities.
- Buff-Focus: In *Step* 3, the task with the greatest buff is allocated first.

First, the average Ω performance with the above task offloading schemes in terms of different numbers of UEs is shown in Fig. 2. Here, we consider the cases of $\eta = 1$ and $\eta = 2$. From Fig. 2, all the average Ω of all schemes decrease as the number of UEs increases, and our proposed scheme consistently has the best performance in all cases, especially when the network workload is light (i.e., small number of UEs). In detail, the average advantages of our proposed scheme over ED-Focus, Buff-Focus, Random and Greedy are 8.41%, 115.17%, 118.15% and 65.34% in the case of $\eta = 1$, and 5.85%, 117.57%, 174.89% and 56.13% in the case of $\eta = 2$. Furthermore, as the number of UEs continues to rise, the performance gap among the schemes is getting increasingly smaller. Because in the state where overloaded UEs occurs, the communication resources and computing resources are no longer sufficient to sustain all task offloading requests. As the number of UEs increases, the proportion of unallocated tasks is increasing, and the effect of task offloading becomes less.

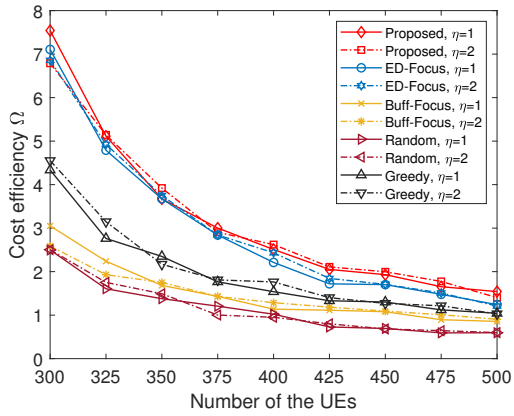

Fig. 2. Average Ω versus different numbers of UEs.

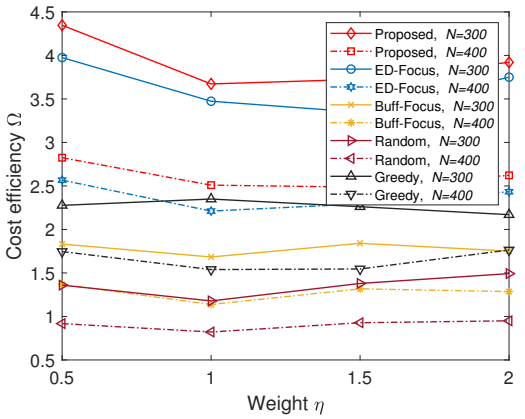

Fig. 3. Average Ω versus different η in the cases of $N = 300, 400$.

Fig. 3 shows the impact of different η on the average Ω in the considered schemes. Here, we consider the cases of $N = 300$ and $N = 400$. From Fig. 3, under different weight settings of η, our proposed scheme can still outperform other schemes. In detail, the average advantages of our proposed scheme over ED-Focus, Buff-Focus, Random and Greedy are 7.70%, 120.40%, 189.58% and 72.96% in the case of $N = 300$, and 9.86%, 104.41%, 188.72% and 58.37% in the case

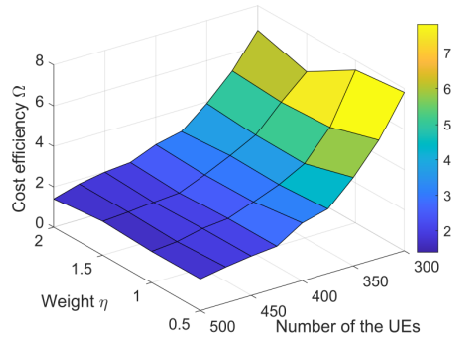

Fig. 4. Average Ω under different η for different numbers of UEs.

of $N = 400$. At the same time, we can see that in the case of $N = 300$, the performance of all schemes is better than that in the case of $N = 400$.

From the poor performance of ED-Focus and Buff-Focus shown in Fig. 2 and Fig. 3, it can be concluded that focusing only on the system costs or the system benefit will lead to a decrease in the cost efficiency of task offloading. Meanwhile, there is a big performance gap between the Greedy scheme and our proposed scheme. Because in the *Step* 3, the task prioritization design integrates the interests of the individual user and the overall users, rather than simply offloading the most cost-effective task first. Finally, Fig. 4 shows the impact of two parameters on the performance of our proposed scheme in the form of a three-dimensional graph. Under different weight settings of η, the scheme performance fluctuates. Besides, with the increase of the number of UEs, all the Ω of the proposed schemes with different η present a downward trend, wherein the reason is the same as the above explained.

V. Conclusion

In this paper, we have investigated the issue of energy-time efficient task offloading for the considered MEC network in hot-spot scenarios. From the novel point of view of cost efficiency of task offloading, we have formulated the task offloading decision-making problem as a 2D-KLP, comprehensively considering the energy cost, time cost and users' service experience enhancement. To solve the problem efficiently, we have proposed a low-complexity iterative greedy heuristic scheme. Simulation results have shown that the proposed scheme can greatly increase the cost efficiency and outperform the considered baselines. In the future work, more complex dynamic network scenarios will be considered.

Acknowledgment

This work is supported in part by National NSFC (Grants No. 61902044 and 62072060), Fundamental Research Funds for the Central Universities (Grant No. 2020CDJQY-A022), National Key R & D Program of China (Grants No. 2018YFB2100100 and 2018YFF0214700), Chongqing Research Program of Basic Research and Frontier Technology (Grant No. cstc2019jcyj-msxmX0589), Key Research Program of Chongqing Science & Technology Commission (Grants No. CSTC2017jcyjBX0025 and CSTC2019jscx-zdztzxX0031), Chinese National Engineering Laboratory for Big Data System Computing Technology, and Canadian NSERC.

References

[1] "Cisco annual internet report (2018-2023)," Tech. Rep., Cisco, Mar. 2020.
[2] X. Wang, X. Li, S. Pack, Z. Han, and V. C.M. Leung, "STCS: Spatial-temporal collaborative sampling in flow-aware Software Defined Networks," IEEE J. Sel. Areas Commun., vol. 38, no. 6, pp. 999-1013, Jun. 2020.
[3] V. Kotsiou, G. Z. Papadopoulos, P. Chatzimisios, and F. Theoleyre, "LDSF: Low-latency distributed scheduling function for industrial Internet of Things," *IEEE Internet of Things Journal*, vol. 7, no. 9, pp. 8688–8699, May 2020.
[4] X. Wang, C. Wang, X. Li, V. C. M. Leung, and T. Taleb, "Federated deep reinforcement learning for Internet of Things with decentralized cooperative edge caching," *IEEE Internet of Things Journal*, vol. 7, no. 10, pp. 9441-9455, Oct. 2020.
[5] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A survey on mobile edge computing: The communication perspective," *IEEE Communications Surveys Tutorials*, vol. 19, no. 4, pp. 2322–2358, Aug. 2017.
[6] Q. Fan and N. Ansari, "Application Aware Workload Allocation for Edge Computing-Based IoT," *IEEE Internet of Things Journal*, vol. 5, no. 3, pp. 2146–2153, Jun. 2018.
[7] C. You, K. Huang, H. Chae, and B. Kim, "Energy-efficient resource allocation for mobile-edge computation offloading," *IEEE Transactions on Wireless Communications*, vol. 16, no. 3, pp. 1397–1411, Mar. 2017.
[8] W. Chen, D. Wang and K. Li,"Multi-User Multi-Task Computation Offloading in Green Mobile Edge Cloud Computing," *IEEE Transactions on Services Computing*, vol. 12, no. 5, pp. 726-738, Oct. 2019.
[9] Q. He, G. Cui, X. Zhang, F. Chen, S. Deng, H. Jin, Y. Li, and Y. Yang, "A game-theoretical approach for user allocation in edge computing environment," *IEEE Transactions on Parallel and Distributed Systems*, vol. 31, no. 3, pp. 515–529, Mar. 2020.
[10] B. Jia, H. Hu, Y. Zeng, T. Xu, and Y. Yang, "Double-matching resource allocation strategy in fog computing networks based on cost efficiency," *Journal of Communications and Networks*, vol. 20, no. 3, pp. 237–246, Aug. 2018.
[11] J. Dai, Z. Zhang, S. Mao, and D. Liu, "A view synthesis-based 360 vr caching system over mec-enabled c-ran," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 10, pp. 3843–3855, Oct. 2020.
[12] R. Li, H. Song, J. Cao, P. Barnaghi, J. Li, and C. X. Mavromoustakis, "Big data intelligent networking," *IEEE Network*, vol. 34, no. 4, pp. 6–7, 2020.
[13] K. Zhang, Y. Mao, S. Leng, Q. Zhao, L. Li, X. Peng, L. Pan, S. Maharjan, and Y. Zhang, "Energy-efficient offloading for mobile edge computing in 5g heterogeneous networks," *IEEE Access*, vol. 4, pp. 5896–5907, Aug. 2016.
[14] X. Guo, R. Singh, T. Zhao, and Z. Niu, "An index based task assignment policy for achieving optimal power-delay tradeoff in edge cloud systems," in *2016 IEEE International Conference on Communications (ICC)*. IEEE, May. 2016, pp. 1–7.
[15] X. Chen, L. Jiao, W. Li, and X. Fu, "Efficient multi-user computation offloading for mobile-edge cloud computing," *IEEE/ACM Transactions on Networking*, vol. 24, no. 5, pp. 2795–2808, Oct. 2016.
[16] R. D. Cohen, "The relative valuation of an equity price index," *The Best of Wilmott*, p. 99, Feb. 2006.
[17] I. Ketykó, L. Kecskés, C. Nemes, and L. Farkas, "Multi-user computation offloading as multiple knapsack problem for 5G mobile edge computing," in *2016 European Conference on Networks and Communications (EuCNC)*, Jun. 2016, pp. 225–229.
[18] Y. Zhao, S. Zhou, T. Zhao, and Z. Niu, "Energy-efficient task offloading for multiuser mobile cloud computing," in *2015 IEEE/CIC International Conference on Communications in China (ICCC)*, Apr. 2015, pp. 1–5.