History-Assisted Online User Allocation in Mobile Edge Computing

Xin He[†], Jiaqi Zheng[‡], Haipeng Dai[‡], Bowen Liu[‡], Wanchun Dou[‡], Guihai Chen[‡], Fu Xiao[†]
[†]School of Computer Science, Nanjing University of Posts and Telecommunications, China
[‡]State Key Laboratory for Novel Software Technology, Nanjing University, China
Email: xhe@njupt.edu.cn, jzheng@nju.edu.cn, haipengdai@nju.edu.cn, liubw@smail.nju.edu.cn
douwc@nju.edu.cn, gchen@nju.edu.cn, xiaof@njupt.edu.cn

2022 IEEE International Conference on Web Services (ICWS) | 978-1-6654-8143-4/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICWS55610.2022.00034

***Abstract*—Mobile edge computing (MEC) is emerging as a novel computing paradigm that pushes network resources (such as computation and storage resources) away from the centralized data center to distributed edge servers. By hiring various resources of nearby edge servers, the MEC provides high-bandwidth and low-latency network services for mobile users. As numerous mobile users may compete for limited edge servers' resources, to improve the resource utilization of the MEC system, it is very critical to investigate an effective user allocation policy. Previous studies mainly focus on investigating offline user allocation policies. However, mobile users may arrive online, and the MEC should be able to allocate these users online too. In a real-world MEC environment, online allocation decisions should not be made entirely in the dark. The historical user requests which may contain powerful hints about future user requests, can be adopted to assist in making allocation decisions. In this paper, we take the historical data into account and study the history-assisted online user allocation strategy. Specifically, we formulate the user allocation problem with a comprehensive model and show its hardness. Then, we present an online algorithm named HOUA to allocate mobile users according to both the online arrived user requests and the historical user requests. The competitive ratio of HOUA is proved. To further verify the effectiveness of HOUA, we conduct experiments on a widely-used real-world dataset. We show that HOUA can allocate more mobile users and achieve high resource rental revenue compared with the other approaches.**

***Index Terms*—Mobile edge computing, online user allocation, history-assisted online algorithm.**

I. Introduction

With the increasing popularity of smartphones and IoT devices, more and more mobile users are attracted by emerging applications such as VR [1], AR [2], [3], and interactive gaming [4]. These applications are usually computation-intensive and latency-sensitive, and require high energy consumption [5]. Limited by mobile devices' computation capabilities and battery power, it has become increasingly impractical to run such applications locally. Although mobile users' computation tasks can be offloaded to the powerful cloud, it suffers from high network latency, which cannot meet the strict delay requirements of the emerging applications [6].

In order to cope with the above problems, mobile edge computing (MEC) is proposed. MEC is a novel and attractive paradigm that provides cloud-like computation capabilities and low-latency network services at the edge of network [7], [8]. In MEC, base stations equipped with edge servers are distributedly located nearby mobile users, and each of them has its specific coverage area [9]. To eliminate the existence of no-signal areas, the coverage areas of neighbor base stations often partially overlap. Mobile users located in the overlap areas can communicate with one of the base stations via wireless connections. By renting various resources of edge servers near base stations, mobile users can access application services and complete their computing tasks in close proximity [10], [11]. Nevertheless, edge servers' resources are still limited [12], [13]. It is difficult for edge servers to fulfill the dense user resource requests within a short period [14]. Besides, mobile users may have different resource demands and willingness to pay for the resources leased from the infrastructure provider, *i.e.*, the manager of edge servers in MEC. According to resource requests of mobile users and capacities of edge servers, the infrastructure provider needs to make allocation decisions, *i.e.*, determining which user requests can be accepted and which can be denied. This problem is referred to as the user allocation problem. Although there are many ways to allocate mobile users' resource requests in MEC environment, the inappropriate allocation strategy can significantly downgrade the users' quality of experience and lower the revenue of the infrastructure provider. To ensure low-latency network services and the resource rental revenue, it is crucial to investigate the effective and efficient user allocation policy.

During the past years, the user allocation problem in MEC has attracted much attention, and a variety of policies with different optimization objections have been proposed, such as optimizing the number of users allocated to hired edge servers [15], the users' QoE [16], the overall system cost [17], *etc.* Most existing solutions allocate mobile users' requests in an offline manner. That is, the decision-maker knows all the user requests at once in advance. However, since the arrival and departure of mobile users are often random in a real-world MEC environment, the user requests are thus also random. Thus, the above solutions cannot handle such randomly arrived user requests generally, resulting in poor allocation performance. Although some online policies have been proposed [18], [19], they assume the user requests arrive in an adversarial order. However, the adversarial model may be too powerful for the user allocation problem, and it is more

978-1-6654-8143-4/22/$31.00 ©2022 IEEE
DOI 10.1109/ICWS55610.2022.00034

realistic to assume that the user requests arrive in random order. Besides, the above online solutions ignore referring to the knowledge of past allocation information when making decisions, *i.e.*, online allocation decisions are made in the dark. In fact, the historical requests are valuable and readily available. They may contain strong indications regarding future user requests, which can be used to assist in making allocation decisions [20].

This paper investigates the history-assisted online user allocation policy in MEC environment. To take the historical data into account, we adopt the random-order model with a sample (ROS) to describe the online user allocation problem. Specifically, ROS randomly samples the input instance and leverages the sampled data for the purpose of learning [21]. In practice, historical data can be treated as a random sample [22]. An intuitive example is that during a specific time period (such as 8:00 PM to 9:00 PM), mobile users in a specific geographic area request edge servers' resources to run their online games. On two consecutive days, requests of mobile users during the same time period in the same geographic area may be similar. Therefore, the historical user requests of the first day can be treated as a random sample from the request set of the second day. Besides, the ROS model requires the remaining part of the input instance to arrive one by one in random order, which replaces the adversarial model with a more realistic one. Since each mobile user has different resource demand and the ability to pay for the resource, allocating different user requests to edge servers will generate different amounts of resource rental revenue. We attempt to make immediately and irrevocably allocation decisions when a user request arrives. The objective is to maximize the overall resource rental revenue. We summarize the main contributions of this paper as follows:

- We introduce a novel theoretical model, *i.e.*, the ROS model, to characterize the online user allocation problem. With the help of the ROS model, the historical user allocation data is properly considered when making allocation decisions. Furthermore, we formulate the user allocation problem in MEC and prove its hardness.
- We propose a History-assisted Online User Allocation algorithm, called HOUA, to solve the online user allocation problem. Specifically, once a user request arrives, HOUA integrates historical data and leverages optimal matching and randomized rounding technologies to allocate the request. We prove the competitive ratio of HOUA.
- We conduct extensive experiments on a widely-used dataset to evaluate the performance of HOUA. The simulation results show that HOUA achieves superior performance in terms of the resource rental revenue and the number of allocated users.

The rest of the paper is organized as follows. Section II motivates the online user allocation problem with an example. Section III introduces our system model and formulates the online user allocation problem. Section IV presents a history-assisted online user allocation algorithm HOUA and analyzes its competitive ratio in detail. Section V evaluates the effectiveness of HOUA by conducting extensive experiments. Section VI reviews the related work and Section VII concludes this paper.

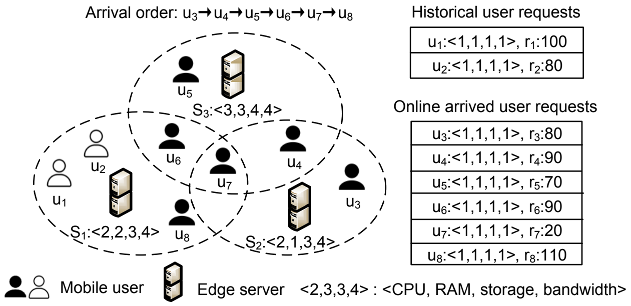


Fig. 1: An online user allocation example

II. Motivating Example

We illustrate the online user allocation problem by using a motivating example.

Fig. 1 shows the user allocation example in MEC environment where there ate three edge servers and a series of user requests. Edge servers $S_1 \sim S_3$ have their geographical coverage areas and are equipped with various resources. Similar to [17], [23], we use a 4-vector $\langle CPU, RAM, ROM, BW \rangle$ to represent the resources of each edge server. As shown in Fig. 1, the capacities of S_1, S_2, and S_3 are $\langle 2,2,3,4 \rangle$, $\langle 2,1,3,4 \rangle$, and $\langle 3,3,4,4 \rangle$, respectively. Mobile users that are inside the coverage area of an edge server can rent the resources of that edge server. Using the resources provided by edge servers, mobile users can complete their computing tasks and use low-latency application services in close proximity. In Fig. 1, there are eight mobile users $u_1 \sim u_8$. $u_1 \sim u_2$ are historical users who have requested resources of edge servers and no longer occupy those resources. $u_3 \sim u_8$ are mobile users who arrive in order. They request the resources of edge servers one by one. For simplicity of illustration, we set the resource demand of each mobile user as $\langle 1,1,1,1 \rangle$. Since each edge server has limited resources, it is impossible to fulfill all of the resource requests of mobile users. To compete for limited resources, each mobile user has a different willingness to pay for the resources. We denote the payment of resource rental for each mobile user u_i as r_i. An online user allocation problem requires the decision-maker to determine whether to accept or deny the online arrived user request in time. The objective is to maximize the resource rental revenue of the infrastructure provider.

Now, we illustrate how the existing solutions solve the online user allocation problem. Without considering the historical user requests, the existing solutions greedily allocate user requests to edge servers that have sufficient resources and cover the users. In Fig. 1, mobile user u_3 appears first and is allocated to edge server S_2. Then, mobile user u_4 occurs. Due to the resource limitation of edge server S_2, u_4 is allocated to edge server S_3. Similarly, the subsequent users u_5, u_6, u_7 are allocated to edge servers S_3, S_1, S_1, respectively. This way, no proximity and capacity constraints are violated. Since the resources of S_1 are occupied by u_6 and u_7, the resource request of u_8 is denied. Therefore, the overall resource rental revenue is $\sum_{i=1}^{7} r_i = 350$.

However, the above allocation policy may not be optimal since it ignores the historical user requests which may contain strong indications regarding future user requests. The history-assisted solution refers to the resource rental payment of historical users u_1 and u_2, and may reject the request of user u_7 since the resource rental payment of user u_7 is low. The resources of edge server S_1 are kept for better future use. When mobile user u_8 occurs, edge server S_1 accepts its resource request to ensure the high resource rental revenue. Therefore, the overall resource rental revenue produced by the history-assisted solution is $\sum_{i=1}^{6} r_i + r_8 = 440$.

In the practical MEC environment, the scale of the online user allocation problem can be much larger than the example discussed. Therefore, it is very challenging to leverage the historical data to find a solution with a theoretical performance guarantee to solve the large-scale online user allocation problem.

III. System Model and Problem Formulation

In this section, we first introduce the system model. Then, we formulate the user allocation problem. The objective is to maximize the resource rental revenue. Finally, we show the NP-hardness of the problem. We summarize the notations of this paper in Table I.

A. System Model

In a specific geographical area, there are n mobile and m edge servers. The user allocation problem refers to how to allocate these n mobile users to m edge servers. We define $cov(s_j)$ and $\boldsymbol{c_{s_j}}$ to denote the coverage area and resource capacity of edge server s_j, where $\boldsymbol{c_{s_j}}$ is a $|t|$-dimensional vector. Each dimension $p \in t$ represents a type of resource, such as CPU, RAM, *etc.* If mobile user u_i is covered by edge server s_j, it can request the resources of s_j. The demand of resource request is denoted by $\boldsymbol{d_{u_i}}$, which is also a $|t|$-dimensional vector. $d_{u_i}^p \in \boldsymbol{d_{u_i}}$ is the p-th dimensional resource demand of u_i. When the resource request of u_i is accepted by edge server s_j, the p-th dimensional resource $c_{s_j}^p$ of s_j is consumed by $d_{u_i}^p$. We denote the revenue of s_j that accepts the resource request of u_i (*i.e.*, the resource rental fees of u_i) as r_{u_i,s_j}.

We adopt the ROS model [21] to introduce the historical data and describe the online user allocation problem. The definition of the ROS model is as follows.

Definition 1. ***ROS Model:*** The ROS model is a model that can simulate the experience that the decision-maker may have. The ROS model gets a random sample with a limited size from an input instance in advance and treats it as the historical data. The remaining part of an input instance is treated as the online set. The ROS model stipulates that the user requests arrive in a uniformly random order.

TABLE I: Key Notations

Notation	Meaning
O	Set of online arrived users
n	Number of online arrived users
H	Set of historical users
k	Number of historical users
U	Set of total users
σ	Number of total users
S	Set of edge servers
m	Number of edge servers
$cov(s_j)$	Coverage area of edge server s_j
t	Set of resource types
p	A type of resource
$\boldsymbol{c_{s_j}}$=$\{c_{s_j}^1, c_{s_j}^2, \ldots, c_{s_j}^{\lvert t \rvert}\}$	Resource capacity of edge server s_j
$c_{s_j}^p$	p-th dimensional resource capacity of edge server s_j
$\boldsymbol{d_{u_i}}$= $\{d_{u_i}^1, d_{u_i}^2, \ldots, d_{u_i}^{\lvert t \rvert}\}$	Resource demand of user u_i
$d_{u_i}^p$	p-th dimensional resource demand of user u_i
r_{u_i,s_j}	Revenue of edge server s_j accepting the resource request of user u_i
$\mathcal{I} = (U, S)$	Input instance of the MEC system

Now we describe the online user allocation problem in the ROS model. We define $\mathcal{I} = (U, S)$ as an input instance in a particular area of MEC system. U represents the set of mobile users, and $|U| = \sigma$. S represents the set of edge servers and $|S| = m$. U can be uniformly and randomly partitioned into a history set H and an online set $O = U \backslash H$. A mobile user in the online set arrives sequentially in a uniformly random order. Once mobile user $u_i \in O$ arrives, according to the historical and current information (including the historical allocation set (H, S), the proximity and capacity constraints of edge servers, the resource demand and resource rental fees of mobile user $u_i \in O$), the decision-maker should decide whether to select one of the edge servers to accept or reject the request of mobile user $u_i \in O$. The decision must be made timely without knowing any information about future user requests. The objective of the online user allocation problem is to find a subset $\tilde{O} \subseteq O$ that maximizes $\sum_{u_i \in \tilde{O}} \sum_{s_j \in S} r_{u_i,s_j}$.

B. Problem Formulation

According to the aforementioned description, we model the above problem as an Integer Linear Program (ILP). We formulate the problem as follows.

$$\text{maximize} \quad \sum_{u_i \in O} \sum_{s_j \in S} r_{u_i,s_j} x_{u_i,s_j} \tag{1}$$

$$\text{subject to:} \quad \sum_{u_i \in O} d_{u_i}^p x_{u_i,s_j} \leq c_{s_j}^p, \quad \forall s_j \in S, p \in t, \tag{1a}$$

$$\sum_{s_j \in S} x_{u_i,s_j} \leq 1, \quad \forall u_i \in O, \tag{1b}$$

$$x_{u_i,s_j} = 0, \quad \forall u_i \notin cov(s_j), \tag{1c}$$

$$x_{u_i,s_j} \in \{0, 1\}, \quad \forall u_i \in O, s_j \in S. \tag{1d}$$

We aim to maximize the total resource rental revenue. We define a zero-one integer variable $x_{u_i,s_j} \in \{0,1\}$ as the indicator of whether the resource request of mobile user u_i is allocated to server s_j. x_{u_i,s_j} equals to one means accepting the resource request and equals to zero means rejecting the resource request. Constraint (1a) ensures that the total resource demands cannot exceed the edge server's capacity in any resource dimension. Constraint (1b) implies that one user request can only be allocated to one edge server. Constraint (1c) reflects that if mobile user u_i is outside of the coverage area of edge server s_j, the request of u_i cannot be allocated to s_j.

C. Problem Hardness

The offline user allocation problem is an NP-hard problem by a reduction from the generalized assignment problem (GAP). The online user allocation problem is substantially harder than the offline user allocation problem because allocation decisions should be made in time without knowing any information about the future user requests. Therefore, the online user allocation problem is NP-hard as well. The detailed proof is omitted due to space constraints.

IV. Online Algorithm Design

To solve the above problem, we present a history-assisted online algorithm named HOUA to solve the user allocation problem.

A. Algorithm Description

In MEC environment, the capacity $\boldsymbol{c_{s_j}}$ of each edge server s_j, as well as the resource demand $\boldsymbol{d_{u_i}}$ of each mobile user u_i may be different. To fully make use of edge servers' resources, we divide the resource requests of mobile users into two categories. Specifically, we define a resource request of mobile user u_i to be $light$ if u_i requests at most half of resources in every resource dimension of edge server s_j, *i.e.*, $d_{u_i}^p \leq \frac{1}{2}c_{s_j}^p, \forall p \in t$. Otherwise, we define the resource request of mobile user u_i to be $heavy$. Let U^{light} and U^{heavy} denote the set of mobile users that request $light$ resources and $heavy$ resources, respectively. $\mathcal{I}^{light} = (U^{light}, S)$ and $\mathcal{I}^{heavy} = (U^{heavy}, S)$ are sub-instances of $\mathcal{I} = (U, S)$ that contain U^{light} and U^{heavy}.

Now, we describe our history-assisted online algorithm, *i.e.*, HOUA, which makes the allocation decisions on $\mathcal{I}$ in time. Algorithm 1 shows the pseudo-code of HOUA. In Algorithm 1, the input H is the set of historical mobile users that is randomly sampled from the set of total mobile users U. The size of historical mobile users is set to $k = |H| \geq \lfloor n/e \rfloor$, where e is a mathematical constant. As each mobile user $u_i \in O$ arrives online, we use $u_i(l) \in O$ to denote the mobile user that arrives at round l. We define V_l to denote the set of historical mobile users at round l, where $V_0 = H$ and $V_l = V_{l-1} \cup u_i(l)$. $A_l = \{(u_i, s_j)\}_l$ is the set of feasible allocated pairs at round l, that is, each element $(u_i, s_j) \in A_l$ denotes that the request of mobile user $u_i \in O$ is allocated to edge server s_j. Since there are n mobile users arriving online, *i.e.*, $|O| = n$, l is set to a number in the range of $[1, n]$. In Algorithm 1, we divide the online allocation process into two phases. In the first phase, we adopt the optimal weighted matching to allocate $heavy$ requests in time. This phase is also named as the $heavy$ phase, which lasts for μn rounds, where μ is a constant that less than 1. In the second phase, we solve problem (1) on the sub-instance $\mathcal{I}^{light}$ and allocate $light$ requests online. This phase is also named as the $light$ phase, which lasts for $n - \mu n$ rounds. At each online round l, HOUA needs to determine whether there is a feasible allocated pair $(u_i(l), s_j(l))$. If a feasible allocated pair $(u_i(l), s_j(l))$ exists, the resource rental revenue will increase r_{u_i,s_j}. Otherwise, the request of mobile user $u_i(l)$ is rejected and the total resource rental revenue is unchanged.

Algorithm 1: HOUA

```
   Input: $H$, $u_i(l) \in O$
   Output: $A_n$
1  $V_0 \leftarrow H$, $A_0 \leftarrow \emptyset$;
2  foreach resource request of mobile user $u_i(l) \in O$ that
     arrives at online round $l$ do
3      $r_{u_i(l),s_j} \leftarrow 0, \forall u_i(l) \notin cov(s_j)$ ;
4      if $1 \leq l \leq \mu n$ then
5          $V_l^{heavy} \leftarrow V_{l-1}^{heavy} \cup \{u_i(l)\}$;
6          $\mathcal{U}_l^{heavy} \leftarrow (V_l^{heavy}, S)$;
7          Obtain the optimal weighted matching $\boldsymbol{x^{(l)}}$ on
             graph $G(\mathcal{U}_l^{heavy}) = (\mathcal{L} \cup \mathcal{R}, E)$;
8          if $(u_i(l), s_j(l))$ is a matched edge in $\boldsymbol{x^{(l)}}$ and
             $s_j(l)$'s resources are sufficient for
             $A_{l-1} \cup \{(u_i(l), s_j(l))\}$ then
9              $A_l \leftarrow A_{l-1} \cup \{(u_i(l), s_j(l))\}$;
10     else
11         $V_l^{light} \leftarrow V_{l-1}^{light} \cup \{u_i(l)\}$;
12         Select a subset $SV_l^{light} \subseteq V_{l-1}^{light}$ of cardinality
             $\lfloor n/e \rfloor$ uniformly at random;
13         $U_l^{light} \leftarrow SV_l^{light} \cup \{u_i(l)\}$;
14         $\mathcal{I}_l^{light} \leftarrow (U_l^{light}, S)$;
15         Solve the LP relaxation of (1) with constraints
             (1a), (1b), (1d) on the sub-instance $\mathcal{I}_l^{light}$ and
             obtain the optimal fraction solution $\boldsymbol{x^{(l)}}$;
16         Select edge server $s_j(l) = s_j$ to allocate mobile
             user randomly with probability $x_{u_i(l),s_j}^{(l)}$;
17         $Pr[s_j(l) = 0] = 1 - \sum_{s_j \in S} x_{u_i(l),s_j}^{(l)}$;
18         if $s_j(l) \neq 0$ and $s_j(l)$'s resources are sufficient
             for $A_{l-1} \cup \{(u_i(l), s_j(l))\}$ then
19             $A_l \leftarrow A_{l-1} \cup \{(u_i(l), s_j(l))\}$;
20 return $A_n$
```

Initially, we have $V_0 = H$ and $A_0 = \emptyset$ (line 1). For each mobile user $u_i(l)$, we set $r_{u_i(l),s_j} = 0$ if $u_i(l)$ is outside of s_j's coverage area, which makes constraint (1c) always hold (line 3). Then, Algorithm 1 starts to allocate $heavy$

requests (lines 4-9). Specifically, we consider the sub-instance $\mathcal{U}_l^{heavy} = (V_l^{heavy}, S)$ and transform $\mathcal{U}_l^{heavy}$ to an equivalent weighted bipartite graph $G(\mathcal{U}_l^{heavy}) = (\mathcal{L} \cup \mathcal{R}, E)$, where $\mathcal{L} = V_l^{heavy}$, $\mathcal{R} = S$, E is the set of edges. If there is an edge $(u_i, s_j) \in E$, it denotes the request of mobile user u_i is allocated to edge server s_j. We set the weight of an edge $(u_i, s_j) \in E$ to r_{u_i,s_j}. At each online round l, we obtain the solution of the optimal weighted matching $\boldsymbol{x^{(l)}}$ on graph $G(\mathcal{U}_l^{heavy})$ (line 7). If there exists a match $(u_i(l),\ s_j(l))$ in $\boldsymbol{x^{(l)}}$, it indicates that allocating mobile user u_i to edge server s_j at round l can achieve high resource rental revenue. Since u_i is $heavy$, we need to check whether the available resources of $s_j(l)$ are sufficient for $u_i(l)$, *i.e.*, the resources of $s_j(l)$ should not be occupied in A_{l-1}. After the $heavy$ phase, Algorithm 1 allocates $light$ requests one by one (lines 10-19). In the $light$ phase, we add $u_i(l)$ into a historical data set and construct a new historical data set V_l^{light} (line 11). To leverage the historical data set to assist in making allocation decisions, we define SV_l^{light} as a subset of V_{l-1}^{light}, which includes $\lfloor n/e \rfloor$ historical user requests randomly sampled from V_l (line 12). Let U_l^{light} be a set that contains SV_l^{light} and $u_i(l)$. The size of U_l^{light} is $\lceil n/e \rceil$. For the input instance $\mathcal{I}_l^{light} = (U_l^{light}, S)$, we solve the problem (1) with constraints (1a), (1b), (1d). Then, we obtain the optimal fraction solution $\boldsymbol{x^{(l)}}$ based on the LP relaxation and randomized rounding technologies (lines 13-15). When there exists a designated edge server (*i.e.*, $s_j(l) \neq 0$) and the residual resources of $s_j(l)$ are sufficient for accepting the resource request of mobile user $u_i(l)$, we allocate $u_i(l)$ to $s_j(l)$ and add the allocated pair $(u_i(l), s_j(l))$ into A_l (lines 16-19).

B. Competitive Analysis

We evaluate the performance of HOUA theoretically via competitive analysis. The competitive ratio is defined as the ratio of the solution obtained by an online algorithm and the solution obtained by an offline optimal algorithm. Specifically, given an input instance $\mathcal{I}$, $ALG(\mathcal{I})$ is an output of the online algorithm ALG and $OPT(\mathcal{I})$ is an output of the offline optimal algorithm OPT. An online algorithm is defined as a c-competitive algorithm if the condition $E[ALG(\mathcal{I})] \geq c \cdot E[OPT(\mathcal{I})]$ always holds for any input instance, where $c \in (0, 1]$. The expectation here reflects the randomness of the sampling of the historical data set and the online arrival order of user requests [21], [24]. For the convenience of illustration, we will replace $OPT(\mathcal{I}^{light})$ and $OPT(\mathcal{I}^{heavy})$ with OPT^{light} and OPT^{heavy} below.

Let $r(A_l)$ denote the total resource rental revenue of allocated pairs in A_l, *i.e.*, $r(A_l) = \sum_{(u_i,s_j)\in A_l} r_{u_i,s_j}$. We define δ_l to denote the revenue improvement from round $l-1$ to round l, *i.e.*, $\delta_l = r(A_l) - r(A_{l-1})$. Specifically, if the resource request of $u_i(l)$ is allocated to s_j at round l, $\delta_l = r_{u_i,s_j}$. On the contrary, in the case that the resource request of mobile user $u_i(l)$ is rejected by all edge servers at round l, $\delta_l = 0$.

Firstly, we bound the expected value of revenue improvement in the heavy phase.

Lemma IV.1. *In the heavy phase, the expected value of revenue improvement at each round is*

$$E[\delta_l] \geq \frac{\lfloor n/e \rfloor}{\lfloor n/e \rfloor + l - 1} \cdot \frac{1}{\mu n} E[OPT^{heavy}], \forall 1 \leq l \leq \mu n$$

Proof. We first prove the expected weight of matched edge $(u_i(l), s_j(l))$, *i.e.*, $E[w(u_i(l), s_j(l))]$. Then, we prove the probability that matched edge $(u_i(l), s_j(l))$ is feasible, *i.e.*, the probability that the resources of $s_l(l)$ are sufficient for $A_{l-1} \cup \{(u_i(l), s_j(l))\}$.

Since H and O are uniformly random subsets of U, $V_l^{heavy} \in U$ is a uniformly random subset with size $\lfloor n/e \rfloor + l$ at each online round l, where $1 \leq l \leq \mu n$. To break down the randomness and determine the arrival order of mobile users until round l, the selection of elements in V_l^{heavy} can be modeled as a series of independent random experiments: First, we select $\lfloor n/e \rfloor + l$ mobile users randomly from U. Then, we determine the order of these $\lfloor n/e \rfloor + l$ elements by randomly choosing an element and removing it. At online round l, Algorithm 1 calculates an optimal weighted matching on $G(\mathcal{U}_l^{heavy})$. Since $u_i(l)$ can be treated as an element that is uniformly and randomly selected from the set V_l^{heavy}, the expected weight of edge $(u_i(l), s_j(l))$ in $\boldsymbol{x^{(l)}}$ is $E[w(u_i(l), s_j(l))] = \frac{E[w(\boldsymbol{x^{(l)}})]}{\lfloor n/e \rfloor + l}$. Since V_l^{heavy} is a uniformly random subset of U, we have $E[w(\boldsymbol{x^{(l)}})] = \frac{\lfloor n/e \rfloor + l}{\sigma} E[OPT(\mathcal{I})]$, where $\sigma = |U|$ and $\mathcal{I} = (U, S)$. We get $E[w(u_i(l), s_j(l))] = \frac{1}{\sigma} E[OPT(\mathcal{I})]$. Since μn $heavy$ mobile users in the online set can be modeled as a random sample from U, $E[OPT^{heavy}] = \frac{\mu n}{\sigma} E[OPT(\mathcal{I})]$ holds. Therefore, we have $E[w(u_i(l), s_j(l))] = \frac{1}{\mu n} E[OPT^{heavy}]$.

Since $u_i(l)$ is a $heavy$ user request, to ensure the available resources of $s_j(l)$ are sufficient for $u_i(l)$, resources of $s_j(l)$ should not be occupied by any mobile users. In the weighted bipartite graph $G(\mathcal{U}_l^{heavy})$, $s_j(l)$ should not be matched at each round before l. For each round $1 \leq \lambda \leq \mu n$, s_j is matched if the vertex s_j is assigned to the left-hand side vertex $u_i(\lambda)$. The λ-th arrived vertex can be seen as a vertex being chosen uniformly at random from the $\frac{1}{\lfloor n/e \rfloor + \lambda}$ vertices on the left-hand side. Therefore, the probability of s_j being matched at round λ is at most $\frac{1}{\lfloor n/e \rfloor + \lambda}$. Since the arrival order of the left-hand vertices is irrelevant, the event that $u_i(\lambda)$ is matched to s_j is independent. We define the event that s_j is not matched before round l as A_l. Therefore, we have

$$Pr[A_l] \geq \prod_{\lambda=1}^{l-1} \left(1 - \frac{1}{\lfloor n/e \rfloor + \lambda}\right) = \frac{\lfloor n/e \rfloor}{\lfloor n/e \rfloor + l - 1}.$$

The probability that resources of $s_j(l)$ is not occupied by any mobile users before round l is $Pr[A_l]$. By combining $E[w(u_i(l), s_j(l))]$, we have

$$E[\delta_l] \geq \frac{\lfloor n/e \rfloor}{\lfloor n/e \rfloor + l - 1} \cdot \frac{1}{\mu n} E[OPT^{heavy}].$$

Therefore, Lemma IV.1 holds. □

Then, we bound the expected value of revenue improvement in the light phase.

Lemma IV.2. *In the light phase, the expected value of revenue improvement at each round is*

$$E[\delta_l] \geq \frac{\lfloor n/e \rfloor}{\lfloor n/e \rfloor + \mu n}(1 - \frac{2|t|(l-\mu n - 1)}{\lceil n/e \rceil})\frac{1}{\lceil n/e \rceil}E[OPT], \quad \forall \mu n + 1 \leq l \leq n$$

Proof. According to the ROS model, elements in the historical set H are obtained by uniform random sampling from U. The remaining part of U constructs the online set O and elements in O arrive in a uniformly random order. Therefore, $V_{l-1}^{light} \subseteq U^{light}$ is a uniformly random subset. Since SV_l^{light} of cardinality $\lfloor n/e \rfloor$ is a uniformly random subset of V_{l-1}^{light}, SV_l^{light} can also be treated as a uniformly random subset of U^{light}. For each $l \in (\mu n + 1, n]$, $U_l^{light} = SV_l^{light} \cup \{u_i(l)\}$ with $\lceil n/e \rceil$ mobile users is a uniformly random subset of U^{light}. Thus, we have $E[OPT(\mathcal{I}_l^{light})] = E[OPT^{light}]$. In Algorithm 1, we get the optimal fraction solution $\boldsymbol{x}^{(l)}$ for the sub-instance $\mathcal{I}_l^{light}$. Therefore, we have

$$\begin{aligned} E\left[\sum_{u_i(l) \in U_l^{light}} \sum_{s_j \in S} r_{u_i(l),s_j} x_{u_i(l),s_j}^{(l)}\right] &\geq E\left[OPT(\mathcal{I}_l^{light})\right] \\ &= E\left[OPT^{light}\right]. \end{aligned}$$

Since u_i is a random element of U_l^{light} and $U_l^{light} \subseteq U^{light}$ is a uniformly random subset of cardinality $\lceil n/e \rceil$, we have

$$\begin{aligned} E\left[r_{u_i(l),s_j(l)}\right] &= E\left[\sum_{s_j \in S} r_{u_i(l),s_j} x_{u_i(l),s_j}^{(l)}\right] \\ &= \frac{1}{\lceil n/e \rceil} E\left[\sum_{u_i(l) \in U_l^{light}} \sum_{s_j \in S} r_{u_i(l),s_j} x_{u_i(l),s_j}^{(l)}\right] \\ &\geq \frac{1}{\lceil n/e \rceil} E\left[OPT^{light}\right]. \end{aligned}$$

We define B_l as an event when edge server $s_j(l)$ has at least half of the available capacity in each resource dimension. Since each $light$ user request consumes at most half of resources in every resource dimension, the occurrence of B_l means that $u_i(l)$ can be allocated to $s_j(l)$. B_l occurs if and only if no mobile users are allocated to s_j in the $heavy$ phase. Besides, the total resource consumption of the $light$ phase before round l should be less than half of each resource of s_j. According to Lemma IV.1, the probability that edge server $s_j(l)$ is not matched in previous μn rounds is $\prod_{l=1}^{\mu n}(1 - \frac{1}{\lfloor n/e \rfloor + l}) = \frac{\lfloor n/e \rfloor}{\lfloor n/e \rfloor + \mu n}$

Let $w_{s_j(l)}^p(l)$ be the p-th dimensional total resource consumption of edge server $s_j(l)$ at online rounds $\mu n + 1, \mu n + 2, \ldots, n$. Now we bound $E[w_{s_j(l)}^p(l)]$ on each resource dimension. For each online round $\lambda \in (\mu n + 1, \ldots, l - 1)$, Algorithm 1 obtains the fraction optimal solution $\boldsymbol{x}^{(\lambda)}$. Since $\boldsymbol{x}^{(\lambda)}$ is a feasible solution, the overall resource consumption of mobile users' requests that are allocated to edge server $s_j(l)$ in dimension p is at most $c_{s_j}^p$. For a random element $u_i(\lambda) \in U_\lambda^{light}$, the expected value of consuming the resources of edge server $s_j(l)$ in any dimension p is at most $\frac{c_{s_j}^p}{\lceil n/e \rceil}$. Therefore, we have

$$E[w_{s_j(l)}^p(l)] \leq \sum_{\lambda = \mu n + 1}^{l-1} \frac{c_{s_j}^p}{\lceil n/e \rceil} = \frac{l - \mu n - 1}{\lceil n/e \rceil} c_{s_j}^p.$$

According to Markov's Inequality, we have

$$Pr[w_{s_j(l)}^p(l) \geq \frac{c_{s_j}^p}{2}] \leq \frac{l - \mu n - 1}{\lceil n/e \rceil} c_{s_j}^p \cdot \frac{2}{c_{s_j}^p} = \frac{2(l - \mu n - 1)}{\lceil n/e \rceil}.$$

Therefore, the probability that more than half of the resources of edge server $s_j(l)$ are consumed in some dimension is at most $\frac{2|t|(l-\mu n-1)}{\lceil n/e \rceil}$. Therefore, we have

$$P[B_l] \geq \frac{\lfloor n/e \rfloor}{\lfloor n/e \rfloor + \mu n}(1 - \frac{2|t|(l - \mu n - 1)}{\lceil n/e \rceil}).$$

According to the expected revenue of allocating $u_i(l)$ to $s_j(l)$, we get

$$E[\delta_l] \geq \frac{\lfloor n/e \rfloor}{\lfloor n/e \rfloor + \mu n}(1 - \frac{2|t|(l - \mu n - 1)}{\lceil n/e \rceil})\frac{1}{\lceil n/e \rceil}E[OPT^{light}].$$

Thus, Lemma IV.2 holds. □

Now, we analyze the competition ratio of HOUA.

Theorem 1. *For $k \geq \lfloor n/e \rfloor$, the competitive ratio of HOUA is $\frac{1}{4|t|}$.*

Proof. According to Lemma IV.1, the overall revenue of the $heavy$ phase is

$$\begin{aligned} \sum_{l=1}^{\mu n} E[\delta_l] &\geq \sum_{l=1}^{\mu n} \frac{\lfloor n/e \rfloor}{\lfloor n/e \rfloor + l - 1} \cdot \frac{1}{\mu n} E[OPT^{heavy}] \\ &= \frac{\lfloor n/e \rfloor}{\mu n} \sum_{l=1}^{\mu n} \frac{1}{\lfloor n/e \rfloor + l - 1} E[OPT^{heavy}] \\ &= \frac{\lfloor n/e \rfloor}{\mu n} \sum_{\tilde{l} = \lfloor n/e \rfloor}^{\mu n + \lfloor n/e \rfloor - 1} \frac{1}{\tilde{l}} E[OPT^{heavy}] \\ &\geq \frac{\lfloor n/e \rfloor}{\mu n} \ln(\frac{\mu n + \lfloor n/e \rfloor}{\lfloor n/e \rfloor}) E[OPT^{heavy}] \\ &\geq \frac{\lfloor n/e \rfloor}{\mu n} \ln(1 + \mu e) E[OPT^{heavy}] \end{aligned}$$

Since $\frac{\lfloor n/e \rfloor}{n} \geq \frac{1}{e} - \frac{1}{n}$ and $\frac{\ln(1+\mu e)}{\mu} > 1$, we have

$$\sum_{l=1}^{\mu n} E[\delta_l] \geq (\frac{1}{e} - \frac{1}{n}) E[OPT^{heavy}].$$

Since $n >> e > 1$, we get

$$\sum_{l=\mu n+1}^{n} E[\delta_l] \geq \frac{1}{e} E[OPT^{heavy}]. \tag{2}$$

For the $light$ phase, Lemma IV.2 gives a non-trivial bound on the expected resource rental revenue improvement for $l \leq \frac{\lceil n/e \rceil}{2|t|} + \mu n + 1$. Therefore, the overall revenue of the $light$ phase is

$$\begin{aligned}
\sum_{l=\mu n+1}^{n} E[\delta_l] &\geq \sum_{l=\mu n+1}^{\frac{\lceil n/e \rceil}{2|t|}+\mu n+1} E[\delta_l] \\
&= \sum_{l=\mu n+1}^{\frac{\lceil n/e \rceil}{2|t|}+\mu n+1} \frac{\lfloor n/e \rfloor}{\lfloor n/e \rfloor + \mu n} \\
&\quad (1 - \frac{2|t|(l-\mu n-1)}{\lceil n/e \rceil}) \frac{1}{\lceil n/e \rceil} E[OPT^{light}] \\
&= \frac{\lfloor n/e \rfloor}{\lfloor n/e \rfloor + \mu n} \frac{1}{\lceil n/e \rceil} E[OPT^{light}] \cdot \\
&\quad \sum_{l=\mu n+1}^{\frac{\lceil n/e \rceil}{2|t|}+\mu n+1} (1 - \frac{2|t|(l-\mu n-1)}{\lceil n/e \rceil}) \\
&= \frac{\lfloor n/e \rfloor}{\lfloor n/e \rfloor + \mu n} \frac{1}{\lceil n/e \rceil} E[OPT^{light}] \cdot \\
&\quad (\frac{\lceil n/e \rceil}{2|t|} + 1 - \frac{2|t|}{\lceil n/e \rceil} \frac{1}{2} \frac{\lceil n/e \rceil}{2|t|} (\frac{\lceil n/e \rceil}{2|t|} + 1)) \\
&= \frac{\lfloor n/e \rfloor}{\lfloor n/e \rfloor + \mu n} \frac{1}{\lceil n/e \rceil} E[OPT^{light}] \cdot (\frac{\lceil n/e \rceil}{4|t|} + \frac{1}{2}) \\
&= \frac{\lfloor n/e \rfloor}{\lfloor n/e \rfloor + \mu n} E[OPT^{light}] \cdot (\frac{1}{4|t|} + \frac{1}{2\lceil n/e \rceil}) \\
&= \frac{1}{1+e\mu} E[OPT^{light}] \cdot (\frac{1}{4|t|} + \frac{1}{2\lceil n/e \rceil}).
\end{aligned}$$

When $\mu \leq \frac{2|t|}{n}$, we have

$$\frac{1}{1+e\mu} (\frac{1}{4|t|} + \frac{1}{2\lceil n/e \rceil}) \geq \frac{1}{4|t|}.$$

Therefore, we get

$$\sum_{l=\mu n+1}^{n} E[\delta_l] \geq \frac{1}{4|t|} E[OPT^{light}]. \tag{3}$$

According to equation (2) and equation (3), the competition ratio of HOUA is

$$\begin{aligned}
E[ALG] &= \sum_{l=1}^{\mu n} E[\delta_l] + \sum_{l=\mu n+1}^{n} E[\delta_l] \\
&\geq \frac{1}{e} E[OPT^{heavy}] + \frac{1}{4|t|} E[OPT^{light}] \\
&\geq \frac{1}{4|t|} E[OPT].
\end{aligned}$$

Thus, Theorem 1 holds. □

V. Performance Evaluation

In this section, we conduct experiments on a real-world dataset. We compare the effectiveness of HOUA against five approaches. In the following, we introduce experiment settings and then present the experimental results. We conduct the experiments on a Windows PC equipped with an Intel Core i7 and 16G RAM.

A. Settings

In our experiments, we adopt the widely-used real-world EUA dataset [15]. The EUA collects the information of edge servers and mobile users in the CBD of Melbourne, Australia. It contains more than 120 edge servers and 800 mobile users. To be more specific, EUA extracts the radio base stations' location according to the data provided by Australian Communications and Media Authority (ACMA) and assumes that each base station is equipped with an edge server. Therefore, the locations of edge servers can be determined. Besides, EUA adopts IP-API [25] to look up the IP addresses of mobile users and convert it into the geographical locations of mobile users [15]. The coverage area of each edge server is randomly selected within the range of 200 $\sim$ 400m. Each edge server is equipped with four types of resources and each mobile user has one resource request. We randomly generate the resources of edge servers, the resource demands of user requests, and the resource rental fees of mobile users according to normal distributions.

In our experiments, the performance metrics are the number of allocated users, and the overall resource rental revenue. To comprehensively evaluate the effectiveness of HOUA, we simulate various user allocation scenarios in MEC by varying the number of online arrived mobile users, the number of edge servers, and the edge servers' capacities, respectively. The details are shown in Table II. Specifically, Set #1.1 and Set #1.2 are for small-scale experiments, and Set #2.1$\sim$Set #2.3 are for large-scale experiments. The last column of Table II indicates the average capacity in each capacity dimension on each edge server. Every time the parameter varies, we repeat the experiment for 10 times, and report the average results. We compare HOUA with the following five baseline user allocation approaches in our experiments:

- OPT [26]: It obtains the offline optimal solution by using the LP-solver, such as the CPLEX Optimizer.
- DRoEUA [18]: It employs a fuzzy control technology to allocate a user to an edge server in real-time.
- BestFit: It allocates a user to the edge server with the smallest amount of residual resources.
- WorstFit [27]: This approach allocates a mobile user to the edge server with the largest amount of residual resources to achieve load balancing.
- Random: It randomly allocates a user to the edge server that can accept the user.

For the large-scale experiments, it is difficult to obtain an optimal solution timely. Therefore, we ignore OPT and only compare HOUA against DRoEUA, BestFit, WorstFit,

TABLE II: Experiment Settings

	Mobile Users	Edge Servers	Capacity
Set #1.1	4, 8, 12, 16	6	1
Set #1.2	12	4, 6, 8, 10	1
Set #2.1	50, 100, ..., 250	60	40
Set #2.2	150	30, 60, ...,150	40
Set #2.3	150	60	20, 40, ...100

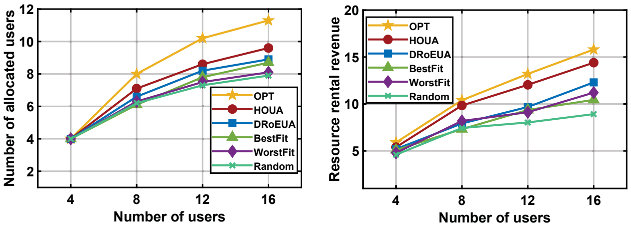
and Random in Set #2.1∼Set #2.3. Besides, we omit the comparison between HOUA and OL-EUA [19] since we mainly focus on leveraging the historical data to make online allocation decisions without considering the signal interference among mobile users.

B. Experimental Results

Fig. 2∼3 show the results in small-scale experiments. Overall, as the number of mobile users and edge servers vary, HOUA can allocate more mobile users and obtain high resource rental revenue compared with DRoEUA, BestFit, WorstFit, and Random.

In Set #1.1, we conduct a small-scale experiment, varying the number of mobile users from $4 \sim 16$. The results are shown in Fig. 2. In Fig. 2, we notice that when there are 4 mobile users, all the mobile users are allocated. It is because resources of edge servers are sufficient for all mobile users. The resource rental revenues generated by the above policies are similar. Then, with the increase of users, the number of allocated users and resource rental revenue increase. OPT can obtain the offline optimal solution so as to allocate the most users and generate the highest resource rental revenue. The performance of HOUA is marginally lower than OPT, but higher than others.

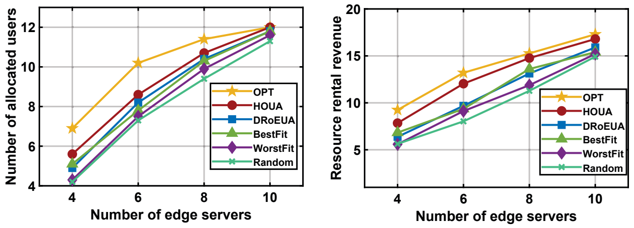
In Set #1.2, we vary the number of edge servers. The results are presented in Fig. 3. Fig. 3 shows that when 12 mobile users arrive online and the average capacity of the edge server is 1, OPT performs best. However, OPT needs too much time to find a feasible solution and cannot allocate mobile users in time. Second to OPT, HOUA can allocate mobile users online and outperforms DRoEUA, BestFit, WorstFit, and Random, respectively. The reason is that HOUA leverages the historical data and has more information to make online allocation decisions.



(a) Number of allocated users (b) Resource rental revenue

Fig. 2: Effectiveness vs. mobile users (Set #1.1)

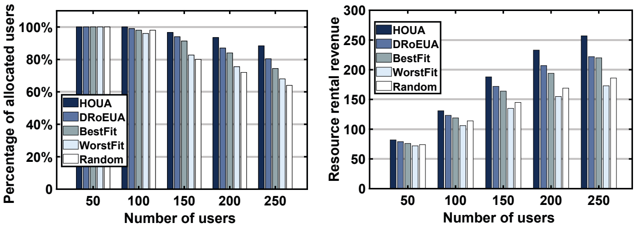
Fig. 4∼6 demonstrate the effectiveness of HOUA in large-scale experiments. In general, with the help of historical data, HOUA outperforms other baseline approaches in terms of the mobile users and the resource rental revenue when parameters vary.



(a) Number of allocated users (b) Resource rental revenue

Fig. 3: Effectiveness vs. edge servers (Set #1.2)

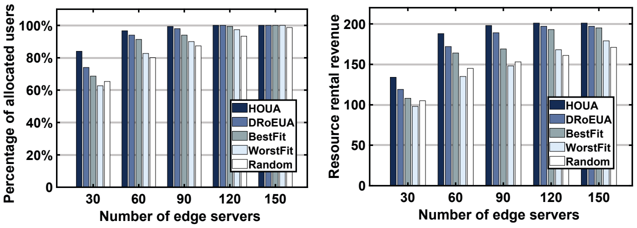
Set #2.1 varies the number of mobile users in large-scale experiments. As shown in Fig. 4, we observe that when the number of mobile users is 50, although all users can be allocated, the resource rental revenue generated by each approach is slightly different. It is because the resource rental fees of mobile users for edge servers are different. Fig. 4(a) shows that the percentage of allocated mobile users decreases when the number of mobile users increases. The reason is that edge servers' resources are not sufficient and mobile users need to compete for the limited resources. Fig. 4(b) shows the resource rental revenue first experiences a rapid increase, and then the growth trend slows down with the increase of mobile users since edge servers' resources are gradually exhausted. Specifically, HOUA outperforms DRoEUA, BestFit, WorstFit, and Random by 9.9%, 18.7%, 30.2%, and 38.1% in the number of allocated users and by 15.8%, 16.8%, 48.6%, and 38.2% in the resource rental revenue when there are 250 mobile users.



(a) Percentage of allocated users (b) Resource rental revenue

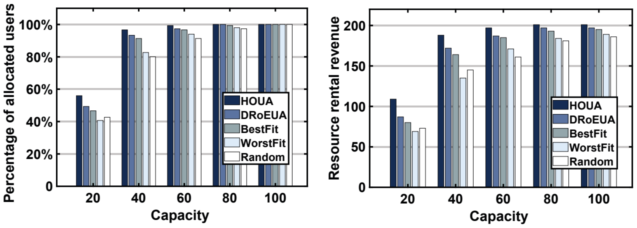
Fig. 4: Effectiveness vs. mobile users (Set #2.1)

Set #2.2 varies the number of edge servers. Fig. 5 shows the trends for the percentage of allocated users and the overall resource rental revenue. When edge servers increase, more available capacities occur and more mobile users can be allocated. We observe that HOUA can properly utilize these resources to serve mobile users. Specifically, in Fig. 5, we observe that HOUA increases the number of allocated users by 13.5% against DRoEUA, by 22.3% against BestFit, by 34.1% against WorstFit, and by 29.8% against Random and improves the resource rental revenue by 12.6% against DRoEUA, by



(a) Percentage of allocated users (b) Resource rental revenue

Fig. 5: Effectiveness vs. edge servers (Set #2.2)



(a) Percentage of allocated users (b) Resource rental revenue

Fig. 6: Effectiveness vs. capacity (Set #2.3)

24.1% against BestFit, by 36.7% against WorstFit, and by 26.6% against Random when there are 30 edge servers in MEC.

Set #2.3 varies the available capacities on each edge server. Fig. 6 shows the corresponding results. With the capacities of edge servers increasing, more mobile users can be allocated to edge servers, resulting in higher resource rental revenue. Although more capacities of edge servers decrease the performance gap among HOUA and other baseline approaches, HOUA is still better than others. Specifically, the average advantages of HOUA over DRoEUA, BestFit, WorstFit, and Random are 3.8%, 5.8%, 12.5%, and 12.7% in terms of allocated mobile users and 8.8%, 12.9%, 25.6%, and 24.1% in terms of resource rental revenue.

VI. Related Work

MEC emerges as an extension of cloud computing that deploys network services (*e.g.*, task computation, data caching) at edge servers [28]. By renting the resources of edge servers, mobile users can complete their computation tasks in close proximity [29]. In MEC, numerous mobile users may compete for limited resources of edge servers. To improve the resource utilization in MEC, it is critical to study the user allocation problem.

Lai et al. [15] first introduced the edge user allocation problem from the application vendor's perspective. They aimed to optimize the number of users allocated to specified edge servers. The LP solver is adopted to obtain the optimal solution. Then, Lai et al. [16] further explored the relationship between the user allocation and the user's quality of experience (QoE). A heuristic approach is designed to maximize the overall QoE. Moreover, Xu et al. [30] analyzed the affect of the distance between a mobile user and an edge server on the data transmission rate. The corresponding algorithm is proposed to allocate mobile users to edge servers with the highest QoS or QoE level. With the overall cost of resource renting taken into account, He et al. [17] presented a decentralized user allocation policy to achieve a Nash equilibrium. The above solutions allocate mobile users offline, *i.e.*, they assume that all of the user requests are known when making allocation decisions. However, the offline allocation policies fail to handle randomly arrived user requests. Based on this, Peng et al. [31] considered the mobility of users and presented a heuristic algorithm to optimize user allocation and migration decisions jointly. Wu et al. [18] built a comprehensive user allocation model, including the user allocation rate, the cost of resource renting, and the server's energy consumption. A method based on fuzzy control is employed to yield the user allocation decisions online. For the NOMA-based MEC system, Cui et al. [19] analyzed the intra-cell and inter-cell interference in detail that existed in the NOMA scheme. They proposed an online algorithm based on the primal-dual technology to allocate the user request and transmit power jointly. The competitive ratio is analyzed and the theoretical performance bound is proved. However, the existing online solutions make user allocation decisions totally in the dark. They ignore the valuable and readily available historical data, which can be leveraged to assist in making online allocation decisions.

In MEC, another popular problem is task offloading. It has some similarities to the user allocation problem. The task offloading mainly focuses on making offload decisions to complete computation tasks. Various works have studied the online algorithms for the task offloading problem. To name a few, Han et al. [32] divided the task offloading problem into two subproblems, *i.e.*, how to assign a task to an edge server and how to schedule tasks in each edge server. They presented an online algorithm named OnDics to minimize the task response time. Meng et al. [33] jointly considered the task dispatching and network. They proposed a deadline-aware greedy scheduling strategy to offload tasks online. Moreover, He et al. [34] proposed a series of online algorithms to offload tasks with different time sensitivities and analyzed the algorithmic competitive ratios. The task offloading problem requires edge servers or the cloud to process each latency-sensitive task rapidly. The occupation and release of resources are thus also frequent. It is very different from the user allocation problem. In our user allocation problem, mobile users rent a specific amount of resources within a certain period of time to use the services deployed on those servers or complete their computational tasks. Besides, the online task offloading algorithms mentioned above also do not consider the historical data.

To the best of our knowledge, our work is the first to solve the online user allocation problem by leveraging the historical data. With the help of historical data, we present an online algorithm with theoretical performance bound to improve the number of allocated mobile users.

VII. Conclusion

This paper investigated the online user allocation problem in MEC environment with the historical data taken into account. Specifically, we first modeled and formulated the online user allocation problem. We showed the NP-hardness of the problem. Next, we presented a history-assisted online algorithm to solve the user allocation problem. We analyzed the competitive ratio of HOUA theoretically. Finally, we conducted experiments on the real-world dataset and evaluated the performance of HOUA against other state-of-the-art approaches. The experimental results show that HOUA can achieve superior performance in terms of the resource rental revenue and the number of allocated mobile users. Our future work may consider a more general MEC environment where each edge server is connected via high-speed links. Under this situation, a mobile user's request can be allocated to other edge servers that do not necessarily cover a user. It requires joint consideration of the user allocation and the route selection.

Acknowledgement

This work was supported in part by the National Science Fund for Distinguished Young Scholars of China under Grant No.62125203, the Key Program of the National Natural Science Foundation of China under Grant No.61932013, the Open Fund of PDL under Grant No. WDZC20205500109, the Natural Science Foundation of Jiangsu Province under Grant No. BK20201248.

References

[1] P. Lin, Q. Song, D. Wang, F. R. Yu, L. Guo, and V. C. Leung, "Resource management for pervasive-edge-computing-assisted wireless vr streaming in industrial internet of things," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 11, pp. 7607–7617, 2021.

[2] Z. Xu, D. Liu, W. Liang, W. Xu, H. Dai, Q. Xia, and P. Zhou, "Online learning algorithms for offloading augmented reality requests with uncertain demands in mecs," in *IEEE ICDCS*, pp. 1064–1074, 2021.

[3] Y. Siriwardhana, P. Porambage, M. Liyanage, and M. Ylianttila, "A survey on mobile augmented reality with 5g mobile edge computing: architectures, applications, and technical aspects," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 2, pp. 1160–1192, 2021.

[4] X. Xia, F. Chen, J. Grundy, M. Abdelrazek, H. Jin, and Q. He, "Constrained app data caching over edge server graphs in edge computing environment," *IEEE Transactions on Services Computing*, 2021.

[5] 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, 2015.

[6] T. X. Tran and D. Pompili, "Joint task offloading and resource allocation for multi-server mobile-edge computing networks," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 1, pp. 856–868, 2018.

[7] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," *IEEE internet of things journal*, vol. 3, no. 5, pp. 637–646, 2016.

[8] R. Yu and P. Li, "Toward resource-efficient federated learning in mobile edge computing," *IEEE Network*, vol. 35, no. 1, pp. 148–155, 2021.

[9] X. Xia, F. Chen, Q. He, G. Cui, J. C. Grundy, M. Abdelrazek, X. Xu, and H. Jin, "Data, user and power allocations for caching in multi-access edge computing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 33, no. 5, pp. 1144–1155, 2021.

[10] T. Ouyang, R. Li, X. Chen, Z. Zhou, and X. Tang, "Adaptive user-managed service placement for mobile edge computing: An online learning approach," in *IEEE INFOCOM*, pp. 1468–1476, 2019.

[11] G. Zhao, H. Xu, Y. Zhao, C. Qiao, and L. Huang, "Offloading tasks with dependency and service caching in mobile edge computing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 32, no. 11, pp. 2777–2792, 2021.

[12] Z. Meng, H. Xu, M. Chen, Y. Xu, Y. Zhao, and C. Qiao, "Learning-driven decentralized machine learning in resource-constrained wireless edge computing," in *IEEE INFOCOM*, pp. 1–10, 2021.

[13] X. Ma, A. Zhou, S. Zhang, and S. Wang, "Cooperative service caching and workload scheduling in mobile edge computing," in *IEEE INFOCOM*, pp. 2076–2085, 2020.

[14] S. Pasteris, S. Wang, M. Herbster, and T. He, "Service placement with provable guarantees in heterogeneous edge computing systems," in *IEEE INFOCOM*, pp. 514–522, 2019.

[15] P. Lai, Q. He, M. Abdelrazek, F. Chen, J. Hosking, J. Grundy, and Y. Yang, "Optimal edge user allocation in edge computing with variable sized vector bin packing," in *Springer ICSOC*, pp. 230–245, 2018.

[16] P. Lai, Q. He, G. Cui, X. Xia, M. Abdelrazek, F. Chen, J. Hosking, J. Grundy, and Y. Yang, "Edge user allocation with dynamic quality of service," in *Springer ICSOC*, pp. 86–101, Springer, 2019.

[17] 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, 2019.

[18] C. Wu, Q. Peng, Y. Xia, Y. Ma, W. Zheng, H. Xie, S. Pang, F. Li, X. Fu, X. Li, *et al.*, "Online user allocation in mobile edge computing environments: A decentralized reactive approach," *Journal of Systems Architecture*, vol. 113, p. 101904, 2021.

[19] G. Cui, Q. He, X. Xia, F. Chen, F. Dong, H. Jin, and Y. Yang, "Ol-eua: Online user allocation for noma-based mobile edge computing," *IEEE Transactions on Mobile Computing*, pp. 1–1, 2021.

[20] Y. Zhang, G. Prekas, G. M. Fumarola, M. Fontoura, I. Goiri, and R. Bianchini, "History-based harvesting of spare cycles and storage in large-scale datacenters," in *USENIX OSDI*, pp. 755–770, 2016.

[21] H. Kaplan, D. Naori, and D. Raz, "Competitive analysis with a sample and the secretary problem," in *SODA*, pp. 2082–2095, SIAM, 2020.

[22] D. Naori and D. Raz, "Online placement of virtual machines with prior data," in *IEEE INFOCOM*, pp. 2539–2548, 2020.

[23] P. Lai, Q. He, J. Grundy, F. Chen, M. Abdelrazek, J. G. Hosking, and Y. Yang, "Cost-effective app user allocation in an edge computing environment," *IEEE Transactions on Cloud Computing*, pp. 1–1, 2020.

[24] M. Babaioff, N. Immorlica, and R. Kleinberg, "Matroids, secretary problems, and online mechanisms," in *SODA*, pp. 434–443, SIAM, 2007.

[25] "IP-API." https://ip-api.com/#103.232.212.48.

[26] "IBM ILOG CPLEX Optimization Studio." https://www.ibm.com/products/ilog-cplex-optimization-studio.

[27] M. Jia, W. Liang, Z. Xu, and M. Huang, "Cloudlet load balancing in wireless metropolitan area networks," in *IEEE INFOCOM*, pp. 1–9, 2016.

[28] N. Abbas, Y. Zhang, A. Taherkordi, and T. Skeie, "Mobile edge computing: A survey," *IEEE Internet of Things Journal*, vol. 5, no. 1, pp. 450–465, 2017.

[29] H. Trinh, P. Calyam, D. Chemodanov, S. Yao, Q. Lei, F. Gao, and K. Palaniappan, "Energy-aware mobile edge computing and routing for low-latency visual data processing," *IEEE Transactions on Multimedia*, vol. 20, no. 10, pp. 2562–2577, 2018.

[30] Z. Xu, G. Zou, X. Xia, Y. Liu, Y. Gan, B. Zhang, and Q. He, "Distance-aware edge user allocation with qoe optimization," in *IEEE ICWS*, pp. 66–74, 2020.

[31] Q. Peng, Y. Xia, Z. Feng, J. Lee, C. Wu, X. Luo, W. Zheng, S. Pang, H. Liu, Y. Qin, *et al.*, "Mobility-aware and migration-enabled online edge user allocation in mobile edge computing," in *IEEE ICWS*, pp. 91–98, 2019.

[32] Z. Han, H. Tan, X.-Y. Li, S. H.-C. Jiang, Y. Li, and F. C. Lau, "Ondisc: Online latency-sensitive job dispatching and scheduling in heterogeneous edge-clouds," *IEEE/ACM Transactions on Networking*, vol. 27, no. 6, pp. 2472–2485, 2019.

[33] J. Meng, H. Tan, X.-Y. Li, Z. Han, and B. Li, "Online deadline-aware task dispatching and scheduling in edge computing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 31, no. 6, pp. 1270–1286, 2019.

[34] X. He, J. Zheng, Q. He, H. Dai, B. Liu, W. Dou, and G. Chen, "Confect: Computation offloading for tasks with hard/soft deadlines in edge computing," in *IEEE ICWS*, pp. 262–271, 2021.