Spatial-Temporal Edge User Allocation: An Expectation Confirmation Theory Approach

Guobing Zou†, Zhiwei Xu†, Xiaoyu Xia, Ya Liu, Yanglan Gan, Bofeng Zhang, Min Zhou*, Qiang He*

Abstract—As the 5th generation (5G) network develops and rolls out rapidly, user requests can be offloaded to nearby edge servers for processing. This alleviates the pressure on the network backhaul and the remote cloud. Nevertheless, the edge user allocation (EUA) problem, as one of the new and main research challenges in the 5G era, has become a major obstacle to ensuring users' Quality of Experience (QoE) in the edge computing environment. Conventional EUA approaches, ranging from static global allocation model to online decision-making model, have ignored the long-term impact of the changes in users' expectations on user-perceived Quality of Service (QoS). Additionally, most existing approaches have not taken into account the distance between an edge user and an edge server, which impacts the user's data rate profoundly. To tackle these challenges, in this paper, we first formulate the spatial-temporal EUA (ST-EUA) problem by (1) modeling distance-aware QoS based on the wireless transmission attenuation process; and (2) modeling users' QoE based on the Expectation Confirmation Theory (ECT). Then, we transform the formulated problem as an optimization problem with multiple objectives and constraints. Finally, we develop two fuzzy control-based approaches, FC and BFC, to solve ST-EUA problem with the consideration of user consolidation and server load balance in the long run simultaneously. FC is designed for finding a sub-optimal allocation for a singleton user on demand in the scenarios where user requests are sparse, while BFC provides a global optimal allocation for a batch of users in the scenarios where user requests are intensive. We conduct extensive experiments based on two widely-used real-world datasets and show the superiority of our two proposed FC and BFC in effectiveness and efficiency over the baselines and the state-of-the-art approaches.

Index Terms—Edge User Allocation, Expectation Confirmation Theory, Quality of Experience, Quality of Service, Edge Service

1 Introduction

NOWADAYS, the world is witnessing a rapid growth of mobile and Internet-of-Things (IoT) devices, e.g., mobile phones, tablets, wearables, etc. According to Cisco's annual internet report (2018-2023) [1], the number of machine-to-machine connections between these end devices will grow to reach 14.7 billion by 2023. Nevertheless, many mobile and IoT devices suffer from constrained limited computation and storage capacities, which poses an obstacle to the deployment of resource-guzzling applications on these devices. Thanks to the advances in wireless communication such as 5G and Wi-Fi in recent years, computation tasks can be offloaded from end-devices to the cloud to take advantage of its configurable and abundant computing and storage resources [2, 3]. However, the cloud computing paradigm is struggling to fulfill the ever-increasing user demands because of expensive bandwidth and unpredictable network traffic over the internet [4]. Besides, an evident weakness of cloud computing is that users are acutely sensitive to network delays which is very difficult to reduce over the wide area network [5]. Service vendors often find it hard to ensure low service latency for their users.

- *G. Zou, Y. Liu, B. Zhang are with the School of Computer Engineering and Science, Shanghai University, Shanghai, China.*
- *Z. Xu, M. Zhou are with the School of Software, Tsinghua University, Beijing, China.*
- *X. Xia is with the School of Information Technology, Deakin University, Australia.*
- *Y. Gan is with the School of Computer Science and Technology, Donghua University, Shanghai, China.*
- *Q. He is with the Department of Computing Technologies, Swinburne University of Technology, Australia.*
- †*These authors contribute equally to this study and share first authorship.*
- **Corresponding authors.*

To minimize end-to-end network latency, the *edge computing* paradigm is included as part of 5G technology stack to provide resources such as CPU, memory and storage at the network edge [6]. In the edge computing environment, edge servers are deployed at cellular base stations to serve users within their coverage areas with real-time responses [7]. In this way, the central cloud is not always required to process users' requests. Computation-intensive or memory-hungry applications such as augmented reality, interactive gaming and autonomous vehicles can be deployed on edge servers to minimize service latency at the network edge and communication overheads over the network backhaul.

Offering unique advantages and opportunities, edge computing also raises many new and crucial challenges. Edge User Allocation (EUA), as one of these challenges, has attracted a lot of attention very recently [8–19]. In an area powered by 5G, edge servers are often densely and unevenly distributed to collectively cover the entire area. Adjacent edge servers' coverage areas often intersect. Under the proximity and capacity constraints, EUA aims to allocate users to their nearby edge servers appropriately to cost-effectively utilize the resources hired on edge servers. In different EUA scenarios, various optimization goals may be pursued, e.g., maximizing user allocation rate [8, 10, 11, 13, 16, 18], minimizing resource cost [8, 13, 16], minimizing communication interference [15, 19], minimizing reallocation rate [10], and maximizing user satisfaction [9, 12, 14, 17]. Unfortunately, these approaches either do not consider the long-term impact of the changes in users' expectations on QoS, or do not consider the different user-perceived QoS caused by wireless transmission loss. This will bring unstable resource allocation to the users, leading

to serious user dissatisfaction.

More specifically, the limitations of existing EUA approaches are twofold. First, they do not consider the temporal feature of the EUA problem properly, which involves the users' QoS expectation and the overall server load profile. For example, a user may maintain a service session for a period of time, and develop an increasing QoS expectation over a series of service invocations during the service session. An inappropriate allocation that produces unstable QoS will fail to fulfil the user's QoS expectation. From the service vendor's perspective, a straightforward objective is to monitor and fulfil the user's QoS expectation over time. Second, it was observed from our previous work [12] that the distance between an edge user and edge server plays an extraordinarily important role in the EUA problem due to its impact on the user's data rate. In general, the data rate of a user connected to an edge server declines gradually as it moves away from the edge server. Existing EUA approaches have not considered this spatial feature properly and suffers from poor performance in real-world EUA scenarios.

This new problem is referred to as the *Spatial-Temporal Edge User Allocation* (ST-EUA), the solution to which must consider the effect of time (*temporal perspective*) and distance (*spatial perspective*). To the best of our knowledge, this paper is the first attempt to investigate the ST-EUA problem. To solve this problem, we propose two approaches based on fuzzy control, namely FC and BFC, for finding an EUA strategy by incorporating user consolidation and server load balance in the long run. In the scenarios where user requests are sparse, we design FC to allocate a singleton user on demand by only using the information of nearby edge servers (*decentralized*). In the scenarios where edge servers face with intensive user requests, we present BFC, which utilizes the overall server load profile (*centralized*) and thus produces a global optimal allocation for a batch of users. It is noteworthy that both FC and BFC balance the allocation preference in user side and service provider side, i.e., consolidating users when edge servers are full of computing resources, and adjusting server capacities when it comes to excessive user demands. The main contributions of this paper are summarized as follows.

- We formulate the ST-EUA problem by modeling distance-aware QoS based on the wireless transmission attenuation process, and modeling users' QoE based on the expectation confirmation theory. The ST-EUA problem can then be transformed into an optimization problem with multiple global constraints.
- We develop two fuzzy control-based approaches, FC and BFC, by coupling the long-term user consolidation and server load balance. FC is designed for find a sub-optimal allocation on demand for a singleton user in the scenarios where user requests are sparse, while BFC produces global optimal allocation for a batch of users in the scenarios where user requests are intensive.
- Extensive experiments conducted on on two widely-used real-world datasets from Melbourne and Shanghai are carried out to demonstrate the effectiveness and efficiency of FC and BFC. The results show that the two proposed approaches significantly outperform state-of-the-art and those baseline approaches.

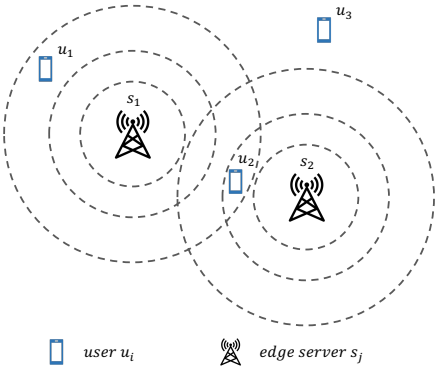


Fig. 1. An example of distance-aware edge user allocation problem.

The remainder of the paper is organized as follows. Section 2 motivates this research with an example. Section 3 formulates and models the ST-EUA problem formally. Section 4 transforms the ST-EUA problem into an optimization problem, and presents FC and BFC in detail. Section 5 presents and discusses the experimental results. Section 6 reviews the related work. Finally, Section 7 concludes the paper and points out the future work.

2 Motivating Example

To illustrate the temporal feature in the EUA problem, let us consider a typical game streaming service at the edge, where gaming video is rendered on the edge server before it is streamed to edge users. Gaming videos can be streamed at different quality levels, e.g., 360p, 720p, 1080p, 1440p and UHD, as long as the edge server has adequate computing resources to process these gaming videos for all the users allocated to it. Many possible allocation strategies can be formulated to maximize the allocation rate [8, 10, 11, 13, 16, 18] or the overall user satisfaction [9, 12, 14, 17]. However, existing approaches have ignored the influence of temporal feature and edge users' QoS expectations. For example, when an active user playing the game with 1080p or UHD, a downgrade to 360p is very likely to may cause a bad user experience. Similarly, if we always try to maximize every user's QoE at the current time slice, they may suffer significant QoE downgrade in subsequent time slices if edge servers' computing resources do not suffice to serve all the users at their expected QoS levels. Therefore, edge user allocation in a time slice requires the proper use of edge services' computing resources. Thus, users' historical and future requests must be taken into account to formulate a sustainable EUA strategy in the long term.

To motivate this study, let us take Figure 1 as an example for distance-aware edge user allocation. There are three edge users $\{u_1, u_2, u_3\}$ and two edge servers $\{s_1, s_2\}$ in a particular area. Based on the pay-as-you-go price model, when the users pay for a service deployed on the edge servers, service vendor make full use of the computing resources on edge servers to serve them. The users that is not covered by any base stations, e.g., u_3, can be served by a

TABLE 1
Notations

Notation	Discription
u_i	The i-th user
s_j	The j-th edge server
c_l	The l-th computing resource level
$w_i^{(t)}$	Allocated QoS of user u_i at time t
$\hat{w_i}^{(t)}$	Expected QoS of u_i at t
$e_i^{(t)}$	Attained QoE of u_i at t
$d_{ij}^{(t)}$	Geographical distance between u_i and s_j at t
$r_i^{(t)}$	Allocated computing resource level for u_i at t
$r_j^{(t)}$	The number of computing resources available to s_j at t
$\gamma(d_{ij}^{(t)})$	Distance-aware QoS attenuation coefficient
β	Strategy parameter for fuzzy control-based EUA approach
$x_{ijl}^{(t)}$	Binary variable to indicate whether u_i is allocated to s_j with c_l at t
$U = \{u_1, u_2, ..., u_{\|U\|}\}$	A set of users
$S = \{s_1, s_2, ..., s_{\|S\|}\}$	A set of edge servers
$T = \{t_1, t_2, ..., t_{\|T\|}\}$	A set of moments
$C = \{c_1, c_2, ..., c_{\|C\|}\}$	A set of computing resource levels

remote cloud server. Since u_1 is closer to s_1, it can receive a stronger wireless connection signal, leading to a higher data transmission rate. Therefore, allocating u_1 to s_1 is a proper solution. Being covered by two overlapping base stations, u_2 can be allocated to either s_1 or s_2. To maximize the overall user satisfaction, existing approaches tend to allocate u_2 to s_2, including the one proposed in our previous work [12]. As a matter of fact, u_3 might also be allocated to s_1 to balance the load on edge servers, or to s_2 to consolidate edge users.

3 System Model

3.1 Problem Definition

In edge computing, edge servers are equipped with diverse computing resources (such as CPU, RAM, storage and bandwidth), and are often deployed around base stations [7]. Edge users can access required services from nearby edge servers instead of the remote cloud. This ensures a reliable and low-latency wireless communication connection. The key notations used in this paper and their descriptions are summarized in Table 1.

From the service providers' view, when they response to their users' service requests, the QoS can be measured, which is defined as follows:

Definition 1. ***(Quality of Service)*** *Given the allocated computing resources $r_i^{(t)}$ to an edge user u_i and the distance between the corresponding edge server $d_{ij}^{(t)}$, the QoS can be calculated to measure the service quality perceived by u_i.*

According to the Expectation Confirmation Theory (ECT) [20], a user's actual QoE is not solely correlated with QoS. It depends on whether its QoS expectation is fulfilled, defined as follows:

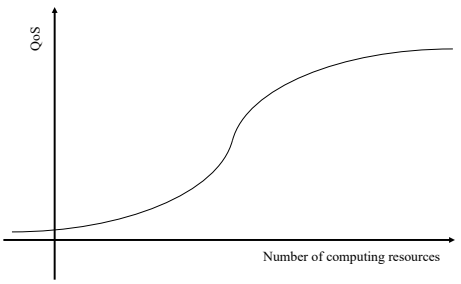


Fig. 2. Quantitative correlation between QoS and computing resources.

Definition 2. ***(Quality of Experience)*** *Given the QoS provided by the service provider and the QoS expected by a user, the user's QoE can be calculated to measure the its satisfaction with the service.*

By integrating the above definitions of QoS and QoE, the spatial-temporal edge user allocation problem can be defined as follows:

Definition 3. ***(Spatial-Temporal Edge User Allocation)*** *Given a set of users $U = \{u_1, u_2, ..., u_{|U|}\}$ and a set of edge servers $S = \{s_1, s_2, ..., s_{|S|}\}$ in a particular area, the spatial-temporal edge user allocation (ST-EUA) problem aims to allocate these users in U to appropriate edge servers in S to maximize their overall QoE, with the consideration of their varying QoS expectation across multiple time slices and their distance from edge servers.*

3.2 Distance-aware QoS Model

3.2.1 Quantitative Correlation between QoS and Computing Resources

The QoS provided to a user by an edge server depends on the computing resources allocated to the user. However, QoS is not linearly correlated with the amount of the computing resources allocated to serve the user [9, 12, 14, 17]. As illustrated in Figure 2, as the amount of computing resources increases, the corresponding QoS raises slowly at first, speeds up after that, and finally converges. Same as in [9, 12, 14, 17, 21], we adopt the sigmoid function to model the correlation between QoE and QoS. For simplicity, the amount of computing resources allocated to a user is discretized into several levels. Formally, it can be expressed as follows:

$$w_l = \frac{A}{1 + e^{B(\bar{c}_l - C)}} \tag{1}$$

where w_l is a discretized QoS level, $\bar{c}_l$ is the mean value of the amount of computing resources, A is the maximum value of QoS, B is the growth rate of the curve and C is the value at the middle of the curve.

3.2.2 QoS Attenuation of Wireless Transmission

Beside computing resources like CPU and RAM, a user's QoS relies heavily on its data transmission rate that is influenced by its wireless signal strength. In general, it decreases when the distance between the user and the edge

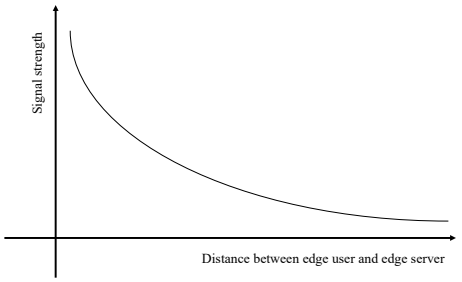


Fig. 3. Correlation between signal strength and distance.

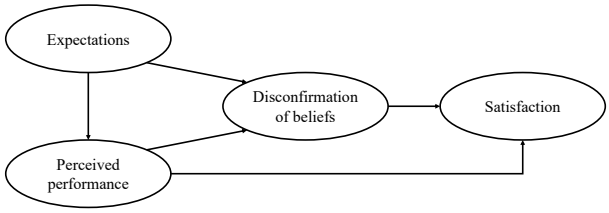


Fig. 4. Illustration of expectation confirmation theory.

server increases. As discussed in [12], the Free Space Path Loss (FSPL) model [22] can be used to model the process of wireless transmission attenuation. It is part of IEEE 802.11 standard [23], where wireless transmission is considered as the spherical diffusion in free space. Thus, the receiver power depends on the occupied surface area on the ball. Specifically, the attenuation of signal strength in a free space can be calculated as follows:

$$\gamma(d) = \frac{P_r}{P_t} = G_t G_r (\frac{\lambda}{4\pi d})^2 \tag{2}$$

where P_r and P_t are the receiver power and transmission power, respectively, G_t and G_r are the transmission antenna gain and receiver antenna gain, which are generally set to 1, respectively, d is the distance between transmitter and receiver and λ is the radio wavelength.

In edge computing, edge servers are attached to base stations and provide computing resources for user requests. Powered by 5G or 4G technologies, user-perceived QoS is attenuated during the wireless transmission process. Therefore, the key influence factor in a user's data rate is its distance from the edge server. As illustrated in Figure 3, the signal strength decreases as this distance increases. The attenuation coefficient can be expressed as:

$$\gamma(d_{ij}^{(t)}) = (\frac{\xi}{d_{ij}^{(t)}})^2 \tag{3}$$

where $\gamma(d_{ij}^{(t)})$ is the attenuation coefficient for the communication, ξ is an parameter that adjusts the variations in the FSPL model.

Given a computing resource level and the distance at time t, an edge user u_i's perceived QoS can be calculated as follows:

$$w_i^{(t)} = w_l \times \gamma(d_{ij}^{(t)}) \tag{4}$$

3.3 ECT-based QoE Model

To accurately reflect edge users' QoE over time, the expectation confirmation theory (ECT) is applied to build a QoE model. ECT consists of four components, including expectations, perceived performance, disconfirmation of beliefs and satisfaction. As shown in Figure 4, expectations refer to the characteristics or attributes that a person anticipates. It is posited to directly influence both perceived performance and disconfirmation of beliefs. Perceived performance refers to the quality of service provided by service providers, measured by QoS. By combining the amount of computing resources and distance between an edge user and an edge server, we can directly apply Equation 4 to represent the edge user's perceived performance. Disconfirmation of beliefs refers to the evaluations or judgements a user makes in regard to a service, compared with its original expectations. When the perceived performance is below an edge user's expectations, the disconfirmation is negative with a unsatisfactory mental state. Conversely, when the actually perceived performance is above its expectations, the disconfirmation of beliefs becomes positive with user satisfaction. Satisfaction refers to the extent that a user satisfy the provided QoS, measured by QoE. It is worthy noting that we assume that the influence of expectations on perceived performance is negligible in that most edge users are more sensitive to provided unstable QoS. In the ST-EUA problem, since perceived performance is represented by the QoS provided from service provider to an edge user u_i at a time t, satisfaction can be calculated based on QoE as follows:

$$e_i^{(t)} = w_i^{(t)} - \hat{w_i}^{(t)} \tag{5}$$

where $w_i^{(t)}$ represents u_i's perceived QoS at time t and $\hat{w_i}^{(t)}$ represents its expected QoS at time slice t.

According to the above calculation in 5, QoE is 0 if an edge user's expected QoS is confirmed. However, if an edge user's disconfirmation is positive or negative, the expected QoS varies. That is, it increases with the positive disconfirmation of beliefs, and decreases with negative disconfirmation of beliefs. Hence, the dynamic expected QoS of an edge user over time can be modeled as follows:

$$\hat{w_i}^{(t+1)} = \hat{w_i}^{(t)} + (1 - \frac{1}{1 + e^{|e_i^{(t)}|}}) \times e_i^{(t)} \tag{6}$$

where $\frac{1}{1+e^{|e_i^{(t)}|}}$ is a standard sigmoid activation function, ranging from 0 to 1. The greater the gap between the expected QoS and actually allocated one becomes, the smaller the activation function is, yielding a bigger change value. However, the the expected QoS does not change indefinitely as the gap increases.

4 Approach

In this section, we first transform the ST-EUA problem into an optimization problem that aims to maximize users' mean QoE across multiple time slices under multiple constrains. To solve the optimization problem, we then present two fuzzy control based approaches, one for generating an immediate allocation strategy with local optimization for a single edge user in decentralized manner, and the other for finding an approximate solution with global information for a batch of edge users in centralized manner.

4.1 ST-EUA Optimization Modeling

We consider an ST-EUA scenario with a set of edge servers $S = \{s_1, s_2, ..., s_{|S|}\}$ in a particular area and an incoming set of edge users $U = \{u_1, u_2, ..., u_{|U|}\}$, The objective of the ST-EUA problem is to find an allocation strategy $f : U \rightarrow S$ that maximizes the sum of user' mean QoE across multiple time slices. The objective function is modeled as:

$$\max : \sum_{t=1}^{|T|} \bar{e^{(t)}} = \sum_{t=1}^{|T|} \frac{\sum_{i=1}^{|U|} e_i^{(t)}}{|U|} \tag{7}$$

Here, users' mean QoE at each t is calculated individually because the numbers of users may vary across multiple time slices. The solution to an ST-EUA problem must fulfil multiple constraints, including capacity constraint, proximity constraint and allocation constraint:

$$s.t. :$$

$$\sum_{i=1}^{|U|} \sum_{l=1}^{|C|} c_l x_{ijl}^{(t)} \leq r_j^{(t)}, \forall s_j \in S, \forall t \in T \tag{8}$$

$$d_{ij}^{(t)} x_{ijl}^{(t)} \leq cov(s_j), \forall u_i \in U, \forall s_j \in S, \forall c_l \in C, \forall t \in T \tag{9}$$

$$\sum_{j=1}^{|S|} \sum_{l=1}^{|C|} x_{ijl}^{(t)} \leq 1, \forall u_i \in U, \forall t \in T \tag{10}$$

$$x_{ijl}^{(t)} = \begin{cases} 1, & \text{if } u_i \text{ is allocated to } s_j \text{ with } c_l \text{ at } t \\ 0, & \text{otherwise.} \end{cases} \tag{11}$$

where $x_{ijl}^{(t)}$ is an binary variable that indicates whether an edge user u_i can be allocated to an edge server s_j with computing resource level c_l in time slot t. Constraint (8) makes sure that any edge server s_j cannot provide the computing resources that exceed their corresponding upper bound capacity at any time t. Constraint (9) illustrates that any edge user u_i can only be allocated to a candidate edge server s_j that covers the u_i at any time t. Constraint (10) indicates that any edge user u_i can be assigned to zero or one edge server at the same time. Constraint (11) ensures that each task can be allocated to at most one edge server.

4.2 Decentralized Fuzzy Allocation of Singleton Edge User

In many time-sensitive application scenarios, service requests submitted by edge users are random. Edge users must be assigned to edge servers immediately so that their requests can be processed timely. To accommodate the randomness in edge users and their requests, we propose an approach named FC based on **F**uzzy **C**ontrol for allocating edge users individually. It is an online approach employed by edge servers to perform real-time user allocation in decentralized manner, pursuing **user consolidation** and **load balance** at the same time.

4.2.1 Overview of FC

Algorithm 1 shows the pseudo code of FC. It starts with an edge user that submits a service request. First, it obtains a set A of all the candidate edge servers satisfying the proximity constraint (line 1). If there are no candidates, the edge user is allocated to the cloud (lines 2-3); otherwise, the most suitable edge server and the corresponding resource level is allocated to the user (lines 5-7).

Algorithm 1 Fuzzy Control (FC)

Input: an edge user u_i.
Output: a decentralized allocation strategy $f : u_i \rightarrow s_j$.

1: $A \leftarrow S(u_i)$ # get the candidate edge servers for u_i, satisfying capacity and proximity constraints
2: **if** $|A| = \varnothing$ **then**
3: allocate u_i directly to the cloud
4: **else**
5: $j \leftarrow SelectServer(u_i, A)$ # Algorithm 2
6: $c_l \leftarrow GetResourceLevel(\beta)$
7: allocate u_i to s_j with resource level c_l
8: **end if**
9: **return** the generated decentralized allocation strategy f

Algorithm 2 SelectServer

Input: an edge user u_i; candidate edge servers A.
Output: selected edge server s_j.

1: $V \leftarrow \{v_j = \frac{r_j^0 - r_j^{(t)}}{r_j^0}, \forall s_j \in S\}$ # calculate the resource utilization rate of all available edge servers
2: $\rho \leftarrow \frac{\sum_{j=1}^{|V|} v_j}{|V|}$ # calculate the mean of the resource utilization rate
3: $\delta \leftarrow \sqrt{\frac{\sum_{j=1}^{|V|} (v_j - \rho)^2}{|V|}}$ # calculate the standard of the resource utilization rate
4: $\beta \leftarrow FuzzyInference(\rho, \delta)$ # calculate strategy parameter
5: $M \leftarrow \varnothing$ # user consolidation scores
6: $N \leftarrow \varnothing$ # load balance scores
7: **for** $j = 1$ to $|A|$ **do**
8: $m_j \leftarrow r_j^{(t)} / d_{ij}^{(t)}$
9: $n_j \leftarrow v_j - \rho$
10: $M \leftarrow M \cup m_j$
11: $N \leftarrow N \cup n_j$
12: **end for**
13: $T \leftarrow \varnothing$ # overall scores
14: **for** $j = 1$ to $|A|$ **do**
15: $x_j = (m_j - min(M))/(max(M) - min(M))$
16: $y_j = (n_j - min(N))/(max(N) - min(N))$
17: $t_j = \beta \times x_j + (1 - \beta) \times y_j$
18: $T \leftarrow T \cup t_j$
19: **end for**
20: $j \leftarrow \arg\max_j \{t_j : t_j \in T\}$
21: **return** s_j

The $SelectServer$ algorithm employed by Algorithm 1 is a fuzzy control based algorithm. Its pseudo code is shown in Algorithm 2. Based on the candidate servers, it calculates the resource utilization rate of all the candidate edge servers (line 1). Then, the mean and standard of resource utilization rate are calculated (lines 2-3). The result is fed into the fuzzy control inference system to obtain a *strategy parameter* (line 4). This strategy parameter represents the tendency to consolidate user preferences or load balance among edge servers, which is also used in $GetResourceLevel$ to obtain the resource level to be allocated. The higher the strategy parameter is, the more likely it will increase user satis-

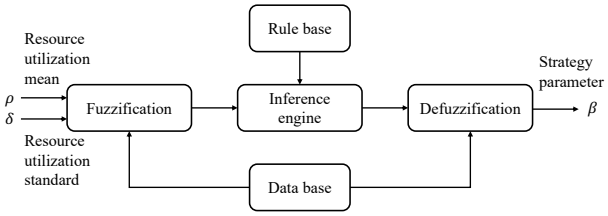


Fig. 5. Illustration of fuzzy inference of strategy parameter.

faction. Once the strategy parameter is deduced, we can calculate the scores for individual candidate edge servers (lines 5-12). It consists of two parts, including user consolidation score and load balance score. Specifically, the user consolidation score of an edge server m_j indicates the extent that it can offer excellent QoS, measured by the ratio between the available computing resources available to s_j and the geographical distance d_{ij} at time t. As for the load balance score of each edge server n_j, it is evaluated by using its current resource utilization rate and candidate edge servers' mean value at time slice t. After the calculation of user consolidation and load balance scores, M and N are standardized and combined to calculate an overall score T for each edge server with the strategy parameter as the regulating coefficient (lines 13-19). Finally, the maximum variable function is applied to find a solution that allocates the user to the edge server with the highest overall score (line 20).

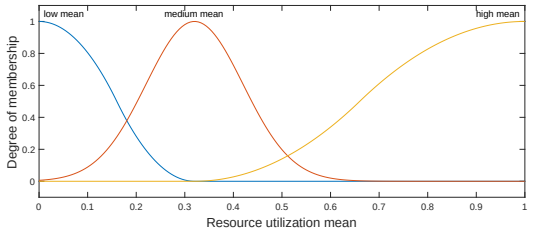
Since the global information is inaccessible, a straightforward method for allocating one edge user with different resource levels is applied by partitioning the strategy parameter into several adjacent discrete intervals. In general, as the interval of the strategy parameter becomes larger, it tends to increase user satisfaction, leading to a higher resource level. It can be formulated as follows:

$$GetResourceLevel(\beta) = \begin{cases} c_1 & 0 \leq \beta < 0.2 \\ c_2 & 0.2 \leq \beta < 0.4 \\ c_3 & 0.4 \leq \beta < 0.6 \\ c_4 & 0.6 \leq \beta < 0.8 \\ c_5 & 0.8 \leq \beta \leq 1.0 \end{cases} \quad (12)$$

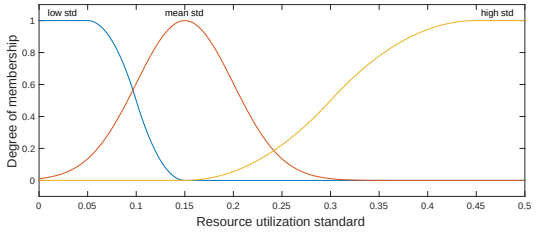
4.2.2 Fuzzy Inference of Strategy Parameter

Figure 5 shows a crucial component of our proposed BFC and FC - $FuzzyInference$. It takes candidate edge servers' mean and standard resource utilization rate as inputs, and outputs the strategy parameter that can balance the supply of edge servers and the demand of users. The specific calculation of $FuzzyInference$ is explained as follows.

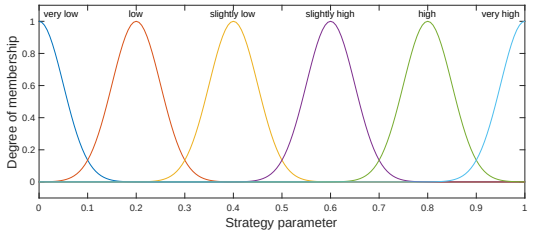
At first, we fuzzify the input and output variables by dividing the mean value $\rho \in [0, 1]$ into three fuzzy sets: **low mean** (LM), **medium mean** (MM) and **high mean** (HM); the standard value $\delta \in [0, 0.5]$ into tree fuzzy sets: **low standard** (LS), **medium standard** (MS) and **high standard** (HS); the strategy parameter $\beta \in [0, 1]$ into six fuzzy sets: **very low** (VL), **low** (L), **slightly low** (SL), **slightly high** (SH), **high** (H) and **very high** (VH). Figure 6 illustrates the membership functions of these fuzzy sets, which apply the commonly-used bell curve and Z-shaped curve to fit the



(a) Resource utilization mean.



(b) Resource utilization standard.



(c) Strategy parameter.

Fig. 6. Membership functions of ρ, δ and β.

membership degree of ρ, δ and β. The process of converting the actual values of ρ and δ into fuzzy categories as the inputs of the inference engine is called fuzzification, while transforming the output of inference engine into the strategy parameter β is called defuzzification. These membership functions are used in the fuzzification and defuzzification processes, indicating that how much of ρ, δ or β belongs to a certain fuzzy category. For example, by the fuzzification of the input parameters, their actual values can be converted into corresponding fuzzified categories. Assuming that the mean resource utilization rate is 0.05, it should be in a low average state, and more resources should be allocated to users to improve their overall QoE.

Given the fuzzy categories and their corresponding membership functions, the empirical rule base can be introduced into the inference engine to generate the mapping relationship from the mean resource utilization rate ρ and the standard resource utilization δ to the strategy parameter β. Here, Table 2 shows inference regulations among ρ, δ and β, which is also visualized in Figure 7. There are four cases:

(1) When ρ and δ are both low, β is very high, the aim is to increase the user consolidation score and to maximize the QoE;
(2) When ρ is high and δ is low, β is slightly high, so as to slightly increase the user consolidation score on the premise of ensuring load balance;
(3) When ρ is low and δ is high, β is slightly low, the allocation scheme prioritizes load balance and

TABLE 2
Rule base of ρ, δ and β for inference engine.

β		δ		
		LS	MS	HS
ρ	LM	VH	SH	SL
	MM	H	SL	L
	HM	SH	L	VL

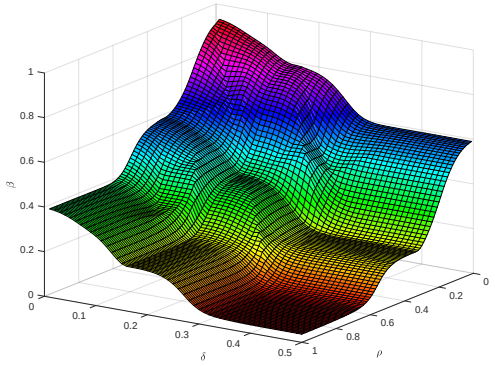


Fig. 7. The visualization of mapping relationship among ρ, δ and β.

temporarily ignores user consolidation;

(4) When ρ is high and δ is high, β is very low, indicating that the edge servers are under high pressure, and the candidate edge servers' workloads are unblanced. In such cases, it mainly focuses on the optimization of load balance to ensure the availability of computing resources for processing service requests in subsequent time slices.

Now we analyze the time complexity of the FC approach. Since the mapping relationship can be calculated offline, the time complexity of $FuzzyInference$ is $O(1)$. Thus, the total time computational complexity is $O(|S|)$ for allocating an edge user to an edge server, where $|S|$ is the number of edge servers. More specifically, it consists of finding the set of candidate edge servers, calculating edge servers' mean and standard resource utilizaiton rate, transforming and integrating the standardized scores as an overall one of each edge server and finding the edge server with the maximum score. To conclude, FC can efficiently obtain allocate users to edge servers in polynomial time, making it a highly efficient approach.

4.3 Centralized Fuzzy Allocation of Batch Edge Users

Although FC can allocate individual edge users efficiently, it is not always a suitable approach because it does not optimize the use of resources across multiple edge servers, especially when users do not have to be allocated in real time. For example, when a user wants to join a multiplayer game online, it is usually acceptable for users to wait for the game to initialize. In such cases, as the demonstrated in [8, 9, 11, 12, 14, 16, 17], edge users can be allocated by batch to optimize the utilization of edge servers' requests. To accommodate such EUA scenarios, we propose an approach named BFC that allocates edge users periodically based on fuzzy control.

Algorithm 3 Batch Fuzzy Control (BFC)

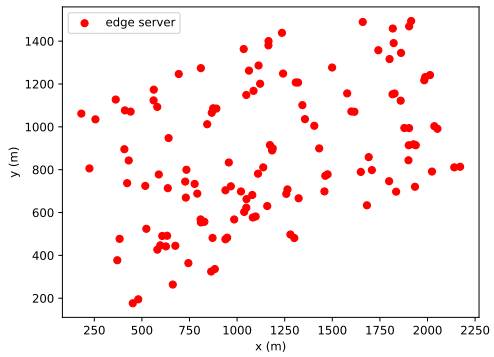
Input: a batch of edge users U.
Output: a centralized allocation strategy $f : U \rightarrow S$.

1: **repeat**
2: **for** $i = 1$ to $|U|$ **do**
3: $A \leftarrow S(u_i)$ # get get the available edge servers for u_i, satisfying capacity and proximity constraints
4: **if** $|A| = \varnothing$ **then**
5: pre-allocate u_i directly to the cloud
6: **else**
7: $j \leftarrow SelectServer(u_i, A)$
8: **if** u_i is not pre-allocated **then**
9: pre-allocate u_i to s_j with c_1
10: **else if** u_i is pre-allocated with c_l and s_j can provide u_i with c_{l+1} **then**
11: pre-allocate u_i to s_j with c_{l+1}
12: **end if**
13: update f with pre-allocation of u_i
14: **end if**
15: **end for**
16: **until** U converges to their allocated resource levels
17: **return** the generated centralized allocation strategy f

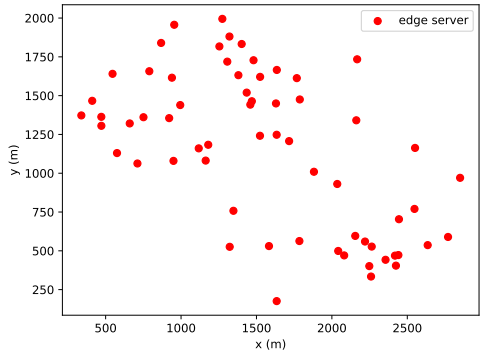
Algorithm 3 presents BFC, a centralized process of allocating edge users by batch. Unlike the FC approach that allocates only one edge user at a time, BFC takes a batch of edge users that arrived in the current time slice as inputs and outputs an allocation strategy for all these edge users. It iteratively carries out pre-allocation to gradually increase the resource level of each edge user to dynamically coordinate the available computing resources. BFC starts with a until-loop (line 1) and terminates upon convergence, i.e., when no edge users' resource levels can be improved further (line 16). For each edge user in U (line 2), the algorithm finds the candidate edge servers and selects a pre-allocated one s_j with the maximum overall score through fuzzy inference (lines 3 - 7). If u_i has not been allocated in previous iterations, it is pre-allocated to s_j with resource level c_1 (lines 8-9); otherwise, if there are computing resources to promote u_i's current resource level, u_i is pre-allocated to s_j with an updated resource level c_{l+1} (lines 10-11). At the end of each iteration, the algorithm updates the pre-allocation state of u_i under different conditions (line 13). When the convergence condition is met, f is returned as the allocation strategy (line 17).

The time complexity of BFC is $O(|U||S||C|)$, where $|U|$, $|S|$ and $|C|$ are the number of edge users, edge servers and resource levels, respectively. Without considering the the number of users, its complexity is linear to the number of resource levels and is only a little higher than FC. This indicates the high efficiency of BFC.

Since $|C|$ is much smaller than both $|U|$ and $|S|$, it can be omitted in the evaluation of the practical time complexity of the BFC approach. Moreover, $|U|$ is also much smaller than $|S|$ in real-world EUA scenarios. Thus, BFC can find a solution to allocate edge users in polynomial time, which is linear to the number of edge users to be allocated.



(a) CBD area in Melbourne



(b) Lujiazui area in Shanghai

Fig. 8. The distribution of edge servers on Melbourne dataset and Shanghai dataset.

5 Experiments

5.1 Datasets and Experimental Setup

To verify the effectiveness and efficiency of FC and BFC, a series of experiments are conducted on two widely-used real-world datasets, including the Melbourne dataset[1] and the Shanghai dataset[2]. In the experiments, we select the base stations from Central Business District (CBD) area of Melbourne and Shanghai Lujiazui area as the benchmarking datasets as shown in Figure 8, which are distributed and scattered unevenly in the areas because of the influence of urban architecture and topography. Specifically, they contain the geographical locations of 125 base stations from Melbourne in Australia and 63 base stations from Shanghai in China respectively.

In the experiments, the parameters of all the approaches, including our approaches and the competing ones, are tuned to achieve the optimal performance. For the settings of the distance-aware QoS model, we set $A = 2$, $B = 1.5$ and $C = 3.5$. The dimension of computing resources is set to be 4, including CPU, RAM, storage and bandwidth. The possible resource levels are partitioned into five cases: $c_1 =< 1, 2, 1, 2 >$, $c_2 =< 2, 4, 1, 3 >$, $c_3 =< 3, 2, 4, 5 >$, $c_4 =< 4, 5, 3, 6 >$ and $c_5 =< 5, 6, 6, 5 >$. For the attenuation coefficient calculation, we set $\xi = 100m$, and $\gamma(d_{ij}^{(t)}) = 1$ if $d_i j^{(t)} < 100m$.

Under each setting, the experiment runs for 150 time slices. Considering that edge users' QoS expectations in the previous time slices are not stable yet, we set the initial expected QoS of all users to be 0 in each run and omit the first 50 time slices to obtain a reliable experiment result. In addition, we repeat each experiments 50 times under each parameter setting and take the average values as the final results. All the experiments are carried out on a machine equipped with Intel(R) Xeon(R) Platinum 8260 CPU @ 2.40GHz.

5.2 Competing Approaches and Evaluation Metrics

To demonstrate the performance of FC and BFC, we compare them with six competing approaches, including three approaches based on integer linear programming (ILP) [8, 9, 12], one state-of-the-art approach [14], and two greedy-based approaches. For ILP-based approaches, we adopt a well-recognized optimizer Gurobi[3] to find the global optimal solution.

- **VSVBP** [8]: This approach solves the EUA problem based on integer linear programming (ILP), aiming to find the optimal solution to maximize the number of allocated edge users and the number of hired edge servers needed.
- **DQoS** [9]: This approach considers the quantitative correlation between QoS and computing resources, and aims to find the optimal solution that maximizes users' overall QoS based on ILP.
- **DEUA** [12]: It is our previous approach that considers the distance between edge user and edge server into the EUA problem. It leverages ILP to optimize edge users' overall QoS.
- **QoEUA** [14]: It is a greedy-based approach based on global information, which gradually increases the amount of computing resources allocated to edge users, and generates a solution that approximates the optimal QoS greedily.
- **MaxQoSGreedy**: It is a greedy-based approach designed for ST-EUA problem. It assigns each edge user to the nearest candidate edge servers with the largest resource level.
- **ProperQoSGreedy**: It is another greedy-based approach designed for ST-EUA problem. It greedily finds a edge server that can accommodates the incoming user with the QoS closest to the QoS expected by that user.
- **FC**: It is our proposed decentralized approach based on fuzzy control inference, which can find a solution in real-time response to an ST-EUA problem for one edge user, balancing user request and server load.
- **BFC**: It is our proposed batch-enhanced fuzzy control approach for an ST-EUA problem, which takes into account global information and finds a solution for a batch of edge users within a period of time intervals.

1. https://github.com/swinedge/eua-dataset
2. http://www.sguangwang.com/TelecomDataset.html
3. https://www.gurobi.com

TABLE 3
Experiment results of edge user allocation among competing approaches on Melbourne dataset.

Methods	Mean QoE	Mean QoS	Allocation Rate	Elapsed CPU Time
MaxQoSGreedy	0.59	128.97	80.31%	38.68
ProperQoSGreedy	-1.66	28.82	90.97%	46.64
VSVBP	0.10	3.57	91.97%	1,314.12
DQoS	7.43	59.26	84.54%	1,340.98
DEUA	3.95	152.52	81.76%	1,252.99
QoEUA	13.28	39.29	99.98%	152.00
FC	11.99	85.12	87.94%	40.88
BFC	24.69	48.59	99.99%	467.41

TABLE 4
Experiment results of edge user allocation among competing approaches on Shanghai dataset.

Methods	Mean QoE	Mean QoS	Allocation Rate	Elapsed CPU Time
MaxQoSGreedy	1.36	91.99	36.91%	29.69
ProperQoSGreedy	-4.12	32.62	43.00%	31.99
VSVBP	0.27	3.93	65.85%	1,334.22
DQoS	3.65	57.73	39.36%	1,188.03
DEUA	-6.39	133.00	37.87%	1,218.52
QoEUA	8.87	19.23	69.60%	130.09
FC	5.91	53.11	42.85%	38.58
BFC	11.37	20.91	70.09%	377.87

In the experiments, we employ four widely-used evaluation metrics to compare and analyze the experiment results, three for effectiveness and one for efficiency.

- **Mean QoE**: It is measured by the sum of mean value of QoE across all the time slices.
- **Mean QoS**: It is measured by the sum of mean value of QoS across all the time slices.
- **Allocation Rate**: It is measured by the percentage of edge users allocated to edge servers.
- **Elapsed CPU Time**: It is measured by the computational time taken to find a solution.

5.3 Experiment Results and Analyses

Table 3 and Table 4 summarize and compare results achieved by the competing approaches on the Melbourne dataset and the Shanghai dataset. The cells marked with dark gray, gray and light gray denote the highest value, the second highest value and the third highest value in the corresponding column, respectively. The total number of edge users in Melbourne and Shanghai are set to 800 and 1,500. It can be observed that BFC always achieves the highest mean QoE and allocation rate among all the approaches. In terms of QoE, it outperforms FC by 105.92%, QoEUA by 85.91%, DEUA by 525.06% and DQoS by 232.30% on Melbourne dataset. The reason is that it considers both spatial and temporal features when allocating edge users with the consideration of users' dynamic QoS expectations over time. However, BFC does not receive superior performance in terms of mean QoS, particularly in scenarios where edge users submit excessive requests in experiments conducted on the Shanghai dataset. The underlying reason may lie in that BFC, similar to QoEUA, takes a batch of edge users into consideration across multiple time slices. This also leads to BFC and QoEUA achieving the highest allocation rates on both datasets. In terms of elapsed CPU time, since BFC and QoEUA incurs extra computation with its consideration of resource levels, their performace is worse than MaxQoSGreedy, ProperGreedy and FC algorithms, whose performance is linear to the number of edge servers. However, they can still obtain competitive computational costs, nearly 3 times lower than the state-of-the-art benchmarks, such as VSVBP, DQoS and DEUA. The main reason is that VSVBP, DQoS and DEUA are designed based on ILP, taking a lot of time to find the optimal solution in each time slice.

Compared with BFC and QoEUA, although FC receives much higher mean QoE than DQoS, DEUA, MaxQoSGreedy, VSVBP and ProperQoSGreedy, it is still lower than that of BFC. The reason is that FC is a decentralized approach, while BFC is designed based on batch-based optimizatio across a set of time slices. Unlike BFC, the main advantage of FC is that it can respond to an edge user's request immediately in time-sensitive application scenarios. Moreover, FC receives extremely high performance in terms of elapsed CPU time, 32.5 times less than VSVBP, DQoS and DEUA, 3.8 times less than QoEUA, and 11.7 times less than BFC. In particular, it is worth noting that QoEUA as a centralized algorithm, is only slightly better than FC in terms of mean QoE, but significantly worse than FC in terms of mean QoS and elapsed CPU time. Thus, taking advantage

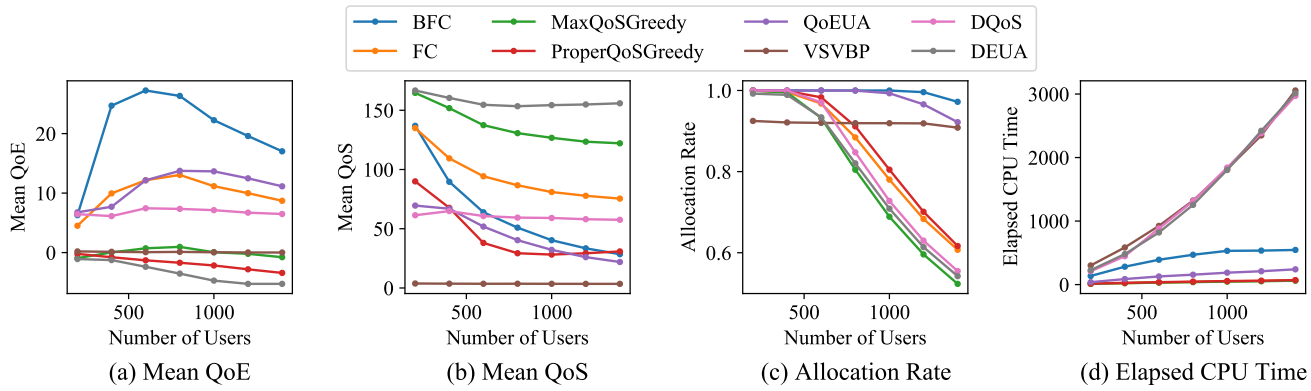


Fig. 9. Performance impact as the number of edge users changes on Melbourne dataset.

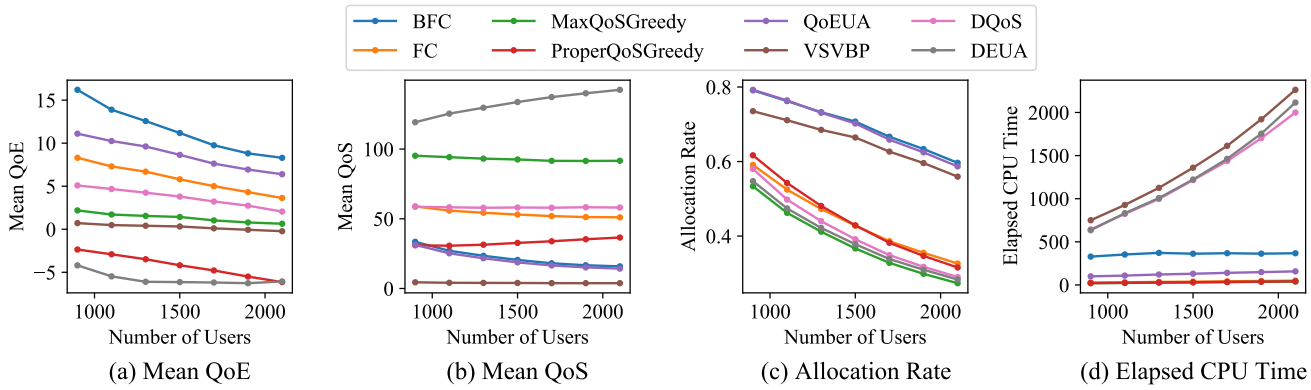


Fig. 10. Performance impact as the number of edge users changes on Shanghai dataset.

of decentralized allocation, FC is completely competitive in terms of multiple performance metrics. MaxQoSGreedy and DEUA pursue to maximize QoS without considering the long-term QoE across multiple time slices, which lead to their poor mean QoE. VSVBP aims at maximizing the allocation rate while minimizing the number of edge servers needed without considering the impact of distance, similar to DQoS. Since MaxQoSGreedy and ProperQoSGreedy pursue optimal QoS within each time slice, they cannot obtain satisfactory QoE for edge users, unlike BFC and FC. Consequently, we conclude that FC and BFC achieve superior performance against the competing approaches.

To evaluate the impacts of different parameters, we vary the number of edge users and compare the results are shown in Figures 9 and 10. In the experiments, the number of edge users in Melbourne dataset varies from 200 to 1,400, with an interval of 200. The number of edge users in Shanghai dataset varies from 1,000 to 2,000 with an interval of 200.

Figure 9(a) shows the change in the mean QoE among competing approaches as the number of users increases. Most of them increase at first before dropping down, because of edge servers' constrained capacity cannot accommodate the increasing workloads. An interesting phenomenon is that FC outperforms the centralized QoEUA approach when the number of edge users reaches 400. This validates that BFC and FC are capable of optimizing computing resource utilization and maintaining a balance between edge servers' resources and users' requests. Figure 9(b) shows the correlation between mean QoS and the number of edge users, where BFC and FC achieve a relatively high performance and ensure the balance between service requests and edge users' QoS expectations. As shown in Figure 9(c), when the number of edge users exceeds 400, the allocation rate for all the competing approaches starts to decline, especially ProperQoSGreedy, FC, DQoS, DEUA and MaxQoSGreedy. The rationale for that is BFC, QoEUA and VSVBP tend to allocate more edge users at all times. Figure 9(d) shows that BFC consumes an acceptable computational cost and FC is almost the best in comparison to all baselines, except MaxQoSGreedy and ProperQoSGreedy. This is because MaxQoSGreedy and ProperQoSGreedy are straightforward allocation approaches that only allocate computing resources greedily.

Figure 10(a) shows that the mean QoE for most of the competing approaches decreases as the number of edge users continues to increase. Specifically, BFC achieves the highest mean QoE followed by QoEUA, which also allocates edge users by batch in a centralized manner. Thanks to the ability of fuzzy control to allocate users with the consideration of user consolidation and load balance, FC achieves the third highest QoE over the other competing approaches in a decentralized manner. Figure 10(b) demonstrates that BFC is more likely to provide lower QoS when the number of edge users becomes larger. The reason is that BFC always tries to allocate more edge users, as shown in Figure 10(c). Figure 10(d) illustrates a similar tendency in elapsed CPU

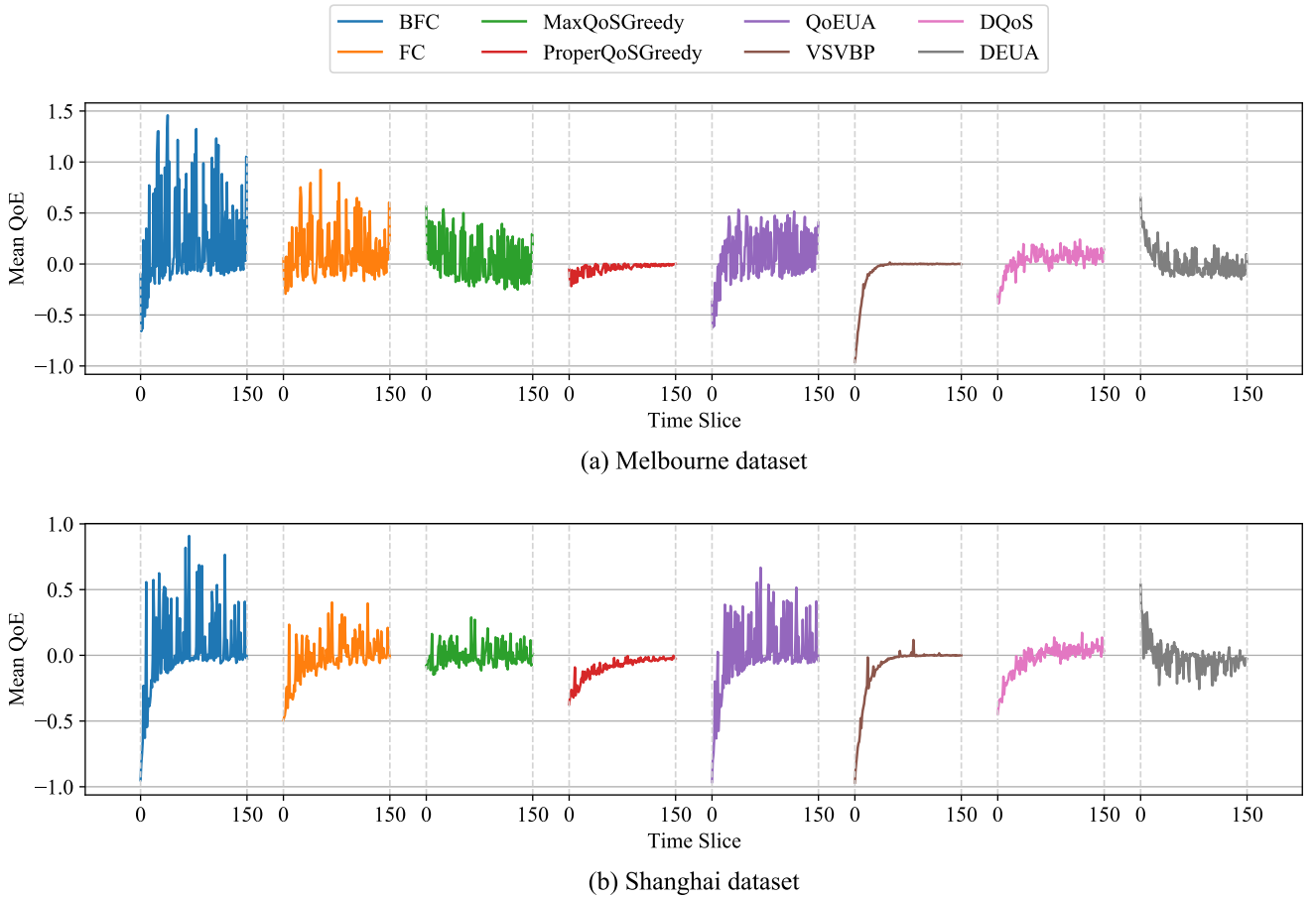


Fig. 11. Experiment results of tendency on mean QoE along with the variations of time slices.

time in experiments conducted on the Melbourne dataset.

5.4 Case Study

To further demonstrate the effectiveness of BFC and FC for ST-EUA problem, we perform a case study to illustrate the allocation process of edge users across different time slices on both Melbourne and Shanghai datasets. In the experiments, we compare the mean QoE obtained by each competing approach across all the 150 time slices and plotting the results on the x axis for better observation, as shown in Figure 11. Here, we assume that an edge user's expected QoS is 0 at the beginning. As time goes, it continues to increase and and finally reaches a stable state.

From the results demonstrated in Figure 11(a), we can see that BFC, FC, GreedyMaxQoS and QoEUA's mean QoE increases in certain time slices, leading to an overall high mean QoE, because they can make full use of the diverse computing resources when the number of service requests is small. The mean QoE achieved by BFC at the beginning is low since BFC always tends to allocate stable computing resources to the edge users. It prioritizes the balance between the number of service requests and edge servers' loads to maximize user experience across multiple time slices.

On the contrary, since GreedyProperQoS and VSVBP mainly aim at providing better QoS and allocating as many edge users as possible without the consideration of QoE, they do not optimize resource utilization well. Thus, their mean QoE stabilizes gradually over time. Furthermore, DEUA starts to achieve a higher mean QoE, since it maximizes the QoS in each time slice to improve users' QoE. In the long term, however, computing resources cannot be leveraged fully, resulting in a decline in mean QoE over time. Therefore, we conclude that since FC and BFC both both strike a balance between supply and demand by considering the long-term impact of edge users' expected QoS. In the meantime, they can both achieve high mean QoE in the long term.

6 Related Work

In recent years, edge computing is widely acknowledged for its unique advantages in offering low latency, cost saving, reliability and scalability. It allows the delivery of new applications and services for the future Internet. Services can be deployed on edge servers to respond to nearby users' requests rapidly. However, due to the proximity and capacity constraints, an edge server has limited computing resources for serving the users with its certain coverage area, which often makes it a challenging task to serve all the edge users with satisfactory QoE. As a novel computing paradigm, it raises many new research challenges in allocation-like problems, including edge user allocation [8, 9, 11, 17], edge

data caching [24, 25], edge server placement [26] and edge application deployment [27, 28].

The edge user allocation (EUA) problem, as one of the main challenges in edge computing, has attracted a lot of attentions recently. Lai et al. [8] firstly introduce and model this problem, and propose a variable sized vector bin packing approach to find a solution. It aims to maximize the number of allocated edge users and minimize the number of edge servers needed. Then, they take a step forward by considering the correlation between computing resources and user satisfaction [9], and further promote their approach with a QoE-aware greedy search [14]. Inheriting the core idea from [8], the authors [16] propose an approach named Most Capacity First (MCF) from service providers' perspective, which is a cost-greedy search that minimizes the number of edge servers needed for serving users. Peng et al. [10] model the EUA problem as a revolvable process. They propose a greedy algorithm based on the mobility of edge users to find an allocation solution.

Furthermore, one of our previous studies [12] attempts to study the distance-aware EUA problem, where the correlation between edge users' signal strength and their distances from edge servers are explored to precisely model the influence of user satisfaction by wireless signal attenuation. In our other study [18], we argue that a user's service request can be partitioned into multiple tasks to be performed by different edge servers, while conventional EUA approaches assume that a service request can either be fully fulfilled by a single edge server or cannot be satisfied at all. Thus, the EUA approach proposed in [18] can accommodate more sophisicated and general real-world application scenarios. In addition, some researchers argue that wireless communication interference occurr when multiple users communicate with the same edge server simultaneously [15, 19]. One one hand, some of the above approaches, such as those proposed in [8, 9, 12, 18], use Integer Linear Programming (ILP) model to search for an optimal solution. On the other hand, some studies [11, 15, 17, 19] transform an EUA problem into an a game problem and employ game-theoretical approaches to solve the game problem. Moreover, the authors of [13] point out that online allocation process should be decentralized to respond to edge users' requests as soon as possible. To achieve this objective, they tackle the EUA problem online with a-fuzzy control-based algorithm to balance user requests and server capacities, which also motivates the core idea of $FuzzyInference$ of our proposed FC and BFC.

However, all of the above approaches have ignored the long-term impact of the temporal feature on edge users. That is, an edge user's QoS expectation often changes continuously over time. Also, the spatial feature affects edge users' QoE significantly. Unlike conventional approaches that oversimplify the EUA problem by converting it as an one-time problem, we formulate the problem with both temporal and spatial features and propose two novel approaches named BFC and FC to tackle the ST-EUA problem in different scenarios. FC and BFC overcome the limitations of existing EUA approaches by modeling the changes in users' QoS expectations over time based on the ECT theory. They also resolve the challenge in how to strike a balance between user requests and edge servers' capacities in the long term, which is a realistic and critical concern from the service provider's perspective.

7 Conclusion and Future work

In this paper, we focus on the problem of edge user allocation with the consideration of spatial and temporal features. The distances between edge users and edge servers are taken into account to reveal the relationship of wireless signal strength and data transmission rate, which plays an important role in edge users' overall QoE. Moreover, dynamic variations on edge users' expected QoS are considered based on the expectation confirmation theory over time, which can effectively capture edge users' demands and improve users' QoE. To solve the ST-EUA problem, we propose two fuzzy logic based approaches FC and BFC, the former for allocating individual edge users in a decentralized manner and the latter by batch in a centralized manner. Through extensive experiments conducted on two real-world datasets, the effectiveness and efficiency of FC and BFC are evaluated against a series of baselines and the state-of-the-art approaches. The results demonstrate their superior performance.

In the future, we plan to explore the migration of edge services among edge servers to make better use of computing resources. We are also going to model the EUA problem as a deep reinforcement learning problem.

Acknowledgments

This work was supported by National Natural Science Foundation of China (No. 61772128, 62172088), and Shanghai Natural Science Foundation (No. 21ZR1400400).

References

[1] U. Cisco, "Cisco annual internet report (2018–2023) white paper," 2020.

[2] M. H. Chen, B. Liang, and M. Dong, "Multi-user multi-task offloading and resource allocation in mobile cloud systems," *IEEE Transactions on Wireless Communications*, vol. 17, no. 10, pp. 6790–6805, 2018.

[3] 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, 2018.

[4] 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.

[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. H. Luan, L. Gao, Z. Li, Y. Xiang, G. Wei, and L. Sun, "Fog computing: Focusing on mobile users at the edge," *arXiv preprint arXiv:1502.01815*, 2015.

[7] Y. C. Hu, M. Patel, D. Sabella, N. Sprecher, and V. Young, "Mobile edge computing—a key technology towards 5G," *ETSI white paper*, vol. 11, no. 11, pp. 1–16, 2015.

[8] 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 *International Conference on Service-Oriented Computing*, 2018, pp. 230–245.

[9] 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 *International Conference on Service-Oriented Computing*, 2019, pp. 86–101.

[10] Q. Peng, Y. Xia, Z. Feng, J. Lee, C. Wu, X. Luo, W. Zheng, H. Liu, Y. Qin, and P. Chen, "Mobility-aware and migration-enabled online edge user allocation in mobile edge computing," in *IEEE International Conference on Web Services*, 2019, pp. 91–98.

[11] 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.

[12] 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 International Conference on Web Services*, 2020, pp. 66–74.

[13] Q. Peng, Y. Xia, Y. Wang, C. Wu, W. Zheng, X. Luo, S. Panz, Y. Ma, and C. Jiang, "A decentralized collaborative approach to online edge user allocation in edge computing environments," in *IEEE International Conference on Web Services*, 2020, pp. 294–301.

[14] P. Lai, Q. He, G. Cui, X. Xia, M. Abdelrazek, F. Chen, J. Hosking, J. Grundy, and Y. Yang, "QoE-aware user allocation in edge computing systems with dynamic QoS," *Future Generation Computer Systems*, vol. 112, pp. 684–694, 2020.

[15] G. Cui, Q. He, X. Xia, P. Lai, F. Chen, T. Gu, and Y. Yang, "Interference-aware saas user allocation game for edge computing," *IEEE Transactions on Cloud Computing*, 2020, DOI: 10.1109/TCC.2020.3008448.

[16] 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*, 2020, DOI: 10.1109/TCC.2020.3001570.

[17] P. Lai, Q. He, G. Cui, F. Chen, M. Abdelrazek, J. Grundy, J. Hosking, and Y. Yang, "Quality of Experience-aware user allocation in edge computing systems: A potential game," in *IEEE International Conference on Distributed Computing Systems*, 2020, pp. 223–233.

[18] G. Zou, Y. Liu, Z. Qin, J. Chen, Z. Xu, Y. Gan, B. Zhang, and Q. He, "TD-EUA: Task-decomposable edge user allocation with QoE optimization," in *International Conference on Service-Oriented Computing*, 2020, pp. 215–231.

[19] G. Cui, Q. He, F. Chen, H. Jin, and Y. Yang, "Trading off between multi-tenancy and interference: A service user allocation game," *IEEE Transactions on Services Computing*, 2020, DOI: 10.1109/TSC.2020.3028760.

[20] R. L. Oliver, "Effect of expectation and disconfirmation on postexposure product evaluations: An alternative interpretation," *Journal of Applied Psychology*, vol. 62, no. 4, pp. 480–486, 1977.

[21] M. Hemmati, B. McCormick, and S. Shirmohammadi, "QoE-aware bandwidth allocation for video traffic using sigmoidal programming," *IEEE MultiMedia*, vol. 24, no. 4, pp. 80–90, 2017.

[22] T. S. Rappaport, *Wireless communications: principles and practice*. Prentice Hall PTR, 1996.

[23] I. L. S. Committee, "IEEE 802.11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications," *IEEE Std 802.11-2016 (Revision of IEEE Std 802.11-2012)*, pp. 1–3534, 2016.

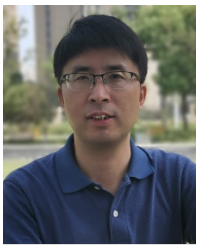
[24] Y. Liu, Q. He, D. Zheng, M. Zhang, F. Chen, and B. Zhang, "Data caching optimization in the edge computing environment," in *IEEE International Conference on Web Services*, 2019, pp. 99–106.

[25] X. Xia, F. Chen, Q. He, G. Cui, P. Lai, M. Abdelrazek, J. Grundy, and H. Jin, "Graph-based optimal data caching in edge computing," in *International Conference on Service-Oriented Computing*, 2019, pp. 477–493.

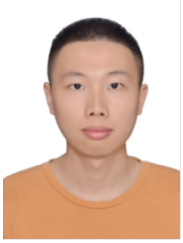
[26] G. Cui, Q. He, X. Xia, F. Chen, H. Jin, and Y. Yang, "Robustness-oriented k edge server placement," in *IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*, 2020.

[27] Y. Chen, S. Deng, H. Zhao, Q. He, Y. Li, and H. Gao, "Data-intensive application deployment at edge: A deep reinforcement learning approach," in *IEEE International Conference on Web Services*, 2019, pp. 355–359.

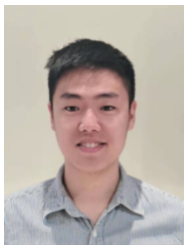
[28] S. Deng, Z. Xiang, J. Taheri, K. A. Mohammad, J. Yin, A. Zomaya, and S. Dustdar, "Optimal application deployment in resource constrained distributed edges," *IEEE Transactions on Mobile Computing*, 2020.



Guobing Zou is an Associate Professor and Dean of the Department of Computer Science and Technology, Shanghai University, China. He received his PhD in Computer Science from Tongji University, Shanghai, China, 2012. He has worked as a Visiting Scholar in the Department of Computer Science and Engineering at Washington University in St. Louis from 2009 to 2011, USA. His current research interests mainly focus on services computing, edge computing, data mining and intelligent algorithms, recommender systems. He has published more than 80 papers on premier international journals and conferences, including IEEE Transactions on Services Computing, IEEE Transactions on Network and Service Management, IEEE International Conference on Web Services, International Conference on Service-Oriented Computing, IEEE International Conference on Services Computing, International Journal of Web Services Research, International Journal of Web and Grid Services, AAAI, Information Sciences, Expert Systems with Applications, Knowledge-Based Systems, Applied Intelligence, etc.

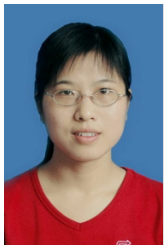


Zhiwei Xu is currently a master student in the School of Software, Tsinghua University, Beijing, China. He received his Bachelor degree in Computer Science from Shanghai University, 2021. His research interests include software engineering, services computing and program languages processing. He has published four papers on premier international journals and conferences, including IEEE International Conference on Web Services, International Conference on Service-Oriented Computing, IEEE Transactions on Knowledge and Data Engineering, and Frontiers in Psychiatry.



Xiaoyu Xia received his master degree from University of Melbourne, Australia in 2015. He is a PhD candidate at Deakin University. His research interests include edge computing, parallel and distributed computing, service computing, software engineering and cloud computing. He has published 23 papers on international premier journals and conferences, including IEEE Transactions on Services Computing, IEEE Transactions on Mobile Computing, IEEE Transactions on Parallel and Distributed Systems, IEEE Transactions on Cloud Computing, Future Generation Computer Systems, IEEE International on Web Services, and International Conference on Service Oriented Computing.

Ya Liu is currently a master student in the School of Computer Engineering and Science, Shanghai University, China. Before that, she received a Bachelor degree in Computer Science and Technology at Jiangxi University of Finance and Economics, 2019. Her research interests include edge computing, QoS management, and heuristic algorithms. She has published two papers on International Conference on Service-Oriented Computing and IEEE International Conference on Web Services, respectively.



Yanglan Gan is an Associate Professor in the school of Computer Science and Technology, Donghua University, Shanghai, China. She received her PhD in Computer Science from Tongji University, Shanghai, China, 2012. Her research interests include bioinformatics, service computing, and data mining. She has published more than 50 papers on premier international journals and conferences, including Bioinformatics, BMC Bioinformatics, IEEE/ACM Transactions on Computational Biology and Bioinformatics, IEEE Transactions on Services Computing, IEEE Transactions on Network and Service Management, IEEE International Conference on Web Services, IEEE International Conference on Service-Oriented Computing, Neurocomputing, and Knowledge-Based Systems. She served as a program committee member on varieties of international conferences.

Bofeng Zhang is a full professor in the School of Computer Engineering and Science at Shanghai University. He received his Ph.D. degree from the Northwestern Polytechnic University (NPU) in 1997, China. He experienced a Postdoctoral Research at Zhejiang University from 1997 to 1999, China. He worked as a visiting professor at the University of Aizu from 2006 to 2007, Japan. He worked as a visiting scholar at Purdue University from 2013 to 2014, US. His research interests include service recommendation, intelligent human-computer interaction, and data mining. He has published more than 150 papers on international journals and conferences. He worked as the program chair for UUMA and ICSS. He also served as a program committee member for multiple international conferences.

Min Zhou is an Associate Professor at School of Software, Tsinghua University, China. He received his PhD degree from Tsinghua University in 2014. His research interests include formal methods, system modelling, program static analysis. He has published more than 30 research papers in international premier journals and conferences, including IEEE Transactions on Software Engineering, Artificial Intelligence, IEEE Transactions on Industrial Electronics, IEEE Transactions on Evolutionary Computation, IEEE Transactions on Circuits and Systems, Journal of Automated Reasoning, Theory of Computing Systems.

Qiang He is an Associate Professor in the Department of Computing Technologies, Swinburne University of Technology, Australia. He received the first PhD degree from Swinburne University of Technology (SUT), Australia, in 2009 and the second PhD degree in computer science and engineering from Huazhong University of Science and Technology (HUST), China, in 2010. His research interests include edge computing, software engineering, cloud computing, services computing, big data analytics, and green computing. More details about his research can be found at https://sites.google.com/site/heqiang/.