Dynamic Adaptive User Allocation in Mobile Edge Computing

1st Jiajia Li
College of Computer Science and Software Engineering
Hohai University
Nanjing, China
jiajiali@hhu.edu.cn

2nd Shunhui Ji
College of Computer Science and Software Engineering
Hohai University
Nanjing, China
Shunhuiji@hhu.edu.cn

3rd Huiying Jin
School of Computer Science
Nanjing University of Posts and Telecommunications
Nanjing, China
hyjin@njupt.edu.cn

4th Hai Dong
School of Computing Technologies
RMIT University
Melbourne, Australia
hai.dong@rmit.edu.au

5th Zhiyuan Ge
College of Computer Science and Software Engineering
Hohai University
Nanjing, China
zhiyuange@hhu.edu.cn

6th Pengcheng Zhang *
College of Computer Science and Software Engineering
Hohai University
Nanjing, China
pchzhang@hhu.edu.cn

***Abstract*—In mobile edge computing (MEC), mobile users can offload tasks to edge nodes to alleviate local computational loads, leveraging the computing capabilities of edge nodes. However, users' high mobility and temporal variability pose challenges in dynamically allocating mobile users to optimize perceived Quality of Service (QoS). To address this challenge, this paper proposes an adaptive ant colony algorithm for user allocation decisions. This method constructs hidden mobility fitness relationships between users and servers based on user movement trajectories. It utilizes an improved adaptive ant colony algorithm to adjust fitness values automatically and optimize user allocation. The goal is to maximize overall user satisfaction under resource constraints while minimizing user allocation costs. Experimental analysis demonstrates that the proposed method achieves higher user allocation rates and effectively utilizes available resources on edge servers.**

***Index Terms*—Mobile Edge Computing, User Allocation, QoS Optimizing, Ant Colony Algorithm, Knapsack Problem**

I. Introduction

With the rapid development of Internet of Things (IoT) devices, accompanied by the emergence of scenarios such as smart cities, smart healthcare, and intelligent transportation, the total number of Internet-connected devices continues to increase [1]. Smart mobile devices (SMD) generate numerous computationally intensive tasks, such as autonomous driving, AR/VR, real-time monitoring, and others [2]. Due to the limited computational capacity of SMD, relying solely on local computing resources is insufficient to meet the demands of such tasks. Mobile Edge Computing (MEC) is a novel distributed computing paradigm that entails the migration of cloud-centric computational capabilities towards edge networks [3]. Within the edge computing environment, edge servers are strategically positioned in proximity to end-users to deliver services, known as edge services [4]. Mobile users can request edge servers to offload local tasks to these servers, leveraging edge node computational resources to alleviate local resource scarcity. Despite the relatively abundant resources in edge nodes compared to end-user devices, the influx of mobile users accessing these nodes may lead to high loads, resulting in resource shortages. Additionally, due to user mobility, sustained long-term connections between users and servers become impractical, leading to service interruptions. These factors can potentially impact the quality of service (QoS) experienced by users. Hence, it becomes particularly important to reasonably allocate edge users to optimize QoS.

Traditional approaches [5]–[8] typically assume that user positions remain static, thereby formulating the user allocation problem as a static optimal allocation problem. However, users' locations are constantly dynamic in the real world. Therefore, traditional static optimization methods may not apply to user allocation in real-world scenarios. Furthermore, some researchers have begun to focus on user allocation in dynamic mobile environments [9]–[11]. These methods often focus on reducing delay or energy consumption from the perspective of service providers or mobile and IoT devices. However, in resource-constrained mobile edge environments,

* Corresponding Author.

J. Li, S. Ji, H. Jin, H. Dong, Z. Ge and P. Zhang, "Dynamic Adaptive User Allocation in Mobile Edge Computing," 2024 IEEE International Conference on Software Services Engineering (SSE), Shenzhen, China, 2024, pp. 179-187, doi: 10.1109/SSE62657.2024.00035.

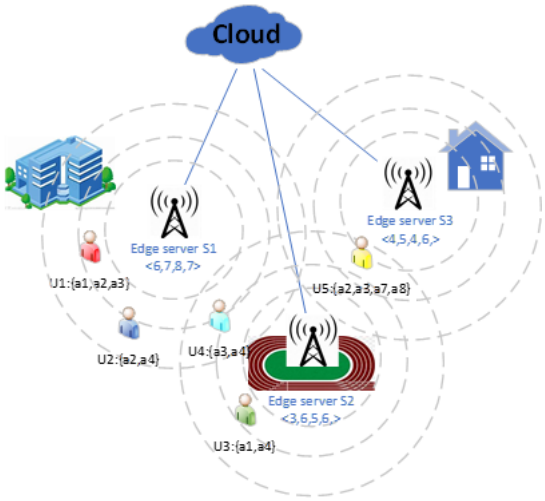

Fig. 1. Mobile Edge Service Invocation Scenario.

solely maximizing user resource demand from the service provider's perspective may result in the satisfaction of some users at the expense of others, leading to a decrease in the Quality of Experience (QoE) for users. Therefore, during the user allocation process, it is essential to consider maximizing global user QoE under limited resources.

We illustrate the motivation using the following scenario of users requesting edge services. As shown in Fig.1, each of the five edge users generates a set of tasks. For instance, edge user $u1$ generates a request set $\{a1, a2, a3\}$. Assuming that each request has resource requirements of $< 1, 1, 1, 1 >$, representing CPU, RAM, Memory, and Bandwidth needs respectively, and adhering to the signal coverage constraints of the edge servers. Edge server $s1$ fully satisfies $u1$ and $u2$, while the requests of $u3$ and $u5$ are allocated to edge servers $s2$ and $s3$. However, the remaining resources on $s2$ are insufficient to fulfill the request of $u4$. If user $u4$ is assigned to a remote cloud, it will result in a high service delivery delay, leading to a poor user experience for $u4$. Typically, a user's task can be decomposed into sub-tasks that multiple edge servers can collaboratively process. Consequently, we can construct user-server fitness relationships based on user mobility trajectories. In this scenario, let us assume that the edge server set $s1, s2$ exhibits the highest fitness compatibility with user $u4$. Therefore, the tasks submitted by $u4$ can be collaboratively processed by edge servers $s1$ and $s2$, where task $a3$ is allocated to $s1$ and task $a4$ is allocated to $s2$. As a result, the total resource requirement for executing the submitted tasks on $s1$ amounts to $< 6, 6, 6, 6 >$, which does not exceed $s1$'s resource of $< 6, 7, 8, 7 >$. Additionally, $s2$ possesses ample resources to handle the allocation of tasks for other users, thereby ensuring the overall quality of service for user requests.

We propose an optimization method based on an improved adaptive ant colony algorithm to address the user allocation problem in the aforementioned mobile edge environment. The main contributions of this paper are as follows:

- We first model the Edge User Allocation (EUA) problem as a 0-1 knapsack problem under finite resource constraints. Based on the evolving trajectories of users in mobile scenarios, we establish concealed fitness relationships between users and servers to derive a candidate service set, thereby ensuring the stability of links between users and servers.
- To address the EUA problem, we simultaneously consider both user allocation costs and the QoE for mobile users. We propose an improved ant colony algorithm with adaptive pheromone updates to prevent the algorithm from falling into the pitfalls of premature convergence and local optima. This approach allows us to obtain the optimal allocation strategy, maximizing user QoE while minimizing the overall user allocation costs.
- Through simulation experiments, we compared our proposed method with several baseline approaches. The results demonstrate that our algorithm significantly outperforms other methods in terms of user allocation rate, resource utilization efficiency, and service response time.

The rest of this paper is organized as follows: Section II introduces the related work. Section III describes the details of the user Allocation problem model in MEC. Section IV describes the proposed task offloading method. Section V presents the experimental evaluation. Finally Section VI concludes the paper.

II. Related Work

Mireslami et al. [12] proposed a multi-objective optimization algorithm to obtain the optimal combination of cloud resources satisfying customer demands. This algorithm aims to minimize deployment costs while meeting QoS performance requirements. However, this method is only applicable to fixed workload scenarios and known supplier pricing strategies. Miao et al. [13] combined artificial intelligence techniques to design an intelligent computation offloading method. They proposed a computation task prediction algorithm based on Long Short-Term Memory (LSTM) and an optimal computation offloading strategy based on task prediction. In large-scale computing and service scenarios, their approach effectively reduces the total task delay. Ding et al. [14] transformed the resource allocation problem into a real-time linear programming sub-problem, proposing a centralized resource allocation strategy that effectively addresses the service allocation optimization problem under budget constraints. This method utilizes Lyapunov optimization techniques to convert the original problem into a series of real-time linear programming subproblems. It introduces a centralized algorithm for resource allocation to user requests. Li et al. [15] introduced a novel user allocation method that implements edge user allocation strategies under constraints such as budget and coverage. This method targets allocating the maximum edge users while minimizing the number of edge servers utilized. They proposed EUA-FOA, utilizing the Fruit Fly Optimization Algorithm to

address the EUA problem, significantly enhancing the efficiency of user allocation. However, these studies only consider the benefits to service providers, such as reducing overall service delay and device energy consumption. The QoE of end-users, who are the consumers of these services, is also crucial for user allocation.

He et al. [9], [10], [16] modeled the user allocation problem considering constraints such as available resources and distance in the edge environment. They proposed heuristic methods to solve the edge user allocation problem. Building upon this, He et al. further considered the dynamic service quality levels of edge service users to find a solution that improves the overall QoE for application users. They also explored edge user allocation based on distance awareness and adversarial awareness. However, these methods often treat user requests as independent entities. In the real world, tasks generated by users may be more complex, and relying solely on a single server may not meet all user needs. Peng et al. [17] considered the high mobility of edge users and viewed the edge user allocation problem as an online decision-making and optimization process. They provided online decisions based on user mobility and temporal characteristics and developed a method named MobMig, which incorporates mobility awareness and migration functionality for real-time user allocation. However, this method is limited to a single service provider offering services to edge users and may not be suitable for scenarios with multiple service providers competing for limited resources. Wu et al. [18] proposed a novel real-time user allocation method. They treated the edge user allocation problem as a static optimization process, considering long-term edge user allocation rates, edge server leasing costs, and edge server energy consumption from the perspective of mobile application providers. They designed a distributed reactive approach based on a fuzzy control mechanism for real-time allocation decisions. However, employing static optimization methods may not be suitable for real-world scenarios characterized by real-time connections and dynamic changes.

The above-mentioned methods primarily optimize user allocation problems in terms of service provider costs, resource utilization at the server end, and energy consumption. However, as service users, user experience plays a crucial role in user allocation strategies. Additionally, since users are mobile, interruptions in connectivity between users and servers may occur. Due to the mobility of users, connections between users and servers may experience interruptions. Therefore, it is imperative to minimize the cost of user service migration and alleviate the impact on user service quality during user mobility.

III. System Model And Problem Description

A. System Model

In a mobile edge environment, each base station covers a specific area, with edge servers placed at these base stations. Services provided by service providers are deployed on these edge servers, limiting access to edge services only to edge users covered by the respective base station. This is known as coverage constraint. During communication between users and edge servers, wireless transmission follows a slow decay pattern. In other words, wireless signal strength depends on the distance between users and edge servers, with closer distances resulting in stronger signal strength. Additionally, each edge server is equipped with available server resources such as CPU, memory, storage, and bandwidth. The resource capacity of servers or the resource requirements of user tasks are represented as a vector $\langle CPU, RAM, Memory, Bandwidth \rangle$. When assigning an edge user to an edge server, the server should have sufficient available server resources to accommodate it, known as resource capacity constraint.

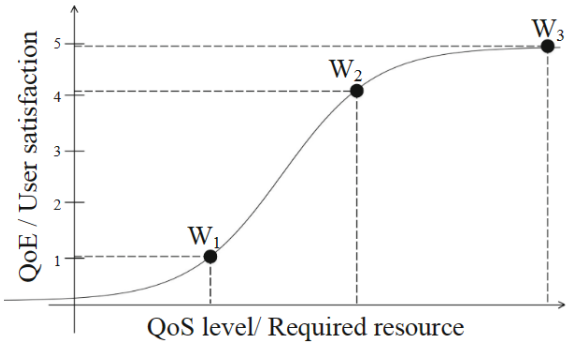

Fig. 2. Quantitative Correlation Chart between QoS and QoE [9].

We define a set of edge server $S = \{s_1, ..., s_m\}$, the set of edge users at time t is denoted as $U_t = \{u_1, ..., u_n\}$. The requests submitted by users can be decomposed into a set of independent tasks. It can be represented as $A_t(u_i) = \{a_1, ..., a_n\}$. Each task a_k can be processed by an edge server s_j covering the user. The total resource demand of all tasks on each edge server must not exceed the available capacity during the respective time, as exceeding this limit may lead to server overload, resulting in performance degradation or even service interruption. The user allocation problem in edge computing environments aims to allocate users to edge servers across multiple time slots, satisfying task requests from users in specific areas and adhering to relevant constraints during user allocation. The objective is to maximize overall user satisfaction and minimize service migration costs.

B. Quality of Experience

Existing research [19]–[21] has already demonstrated a quantitative correlation between QoS and QoE, as depicted in Fig.2. At a certain point (e.g., W3), user satisfaction tends to converge, meaning that regardless of the QoS level, QoE remains nearly constant at its highest level. Each user u_i is allocated resources corresponding to a specific QoS level, resulting in different levels of QoE. Currently, most research measures user QoE through the quality of service delivered to the user [19]. Generally, the QoS of a user is nonlinearly correlated with its QoE [20]. The correlation between QoE and QoS is modeled using a *sigmoid* function, expressed by

the following equation:

$$E_i^0 = \frac{L}{1 + e^{-\alpha(x_i - \beta)}} \tag{1}$$

where $x_i = \frac{\sum_{k \in \mathcal{D}} w_i^k}{|\mathcal{D}|}$, $\mathcal{D}$ represents the set of server tasks. w_i^k represents the number of resources required for the task, L is the maximum achievable QoE value, β is the target level that QoE should reach, α is the growth rate of QoE, indicating how quickly QoE changes from its minimum to maximum. It is worth noting that if user u_i is not allocated, then $E_i = 0$.

The signal strength of a user weakens as the distance from the edge server increases. The attenuation of data rate in wireless transmission must be considered to measure user signal strength. According to the Free-Space Path Loss (FSPL) model [22], the signal power attenuation in free space can be calculated as:

$$f(d) = G_t G_r \left(\frac{\lambda}{4\pi d_{ij}} \right)^2 \tag{2}$$

where G_r and G_t are respectively the receiver and transmitter antenna gains, typically set to 1, λ is the wavelength, and d_{ij} is the distance between the edge server s_j and edge mobile user u_i.

In the scenario of user allocation with collaborative multi-edge servers based on the attenuation coefficients above, the calculation of user QoE is expressed by the following formula. The W_i^t represents the resources allocated to user u_i at time t.

$$E_i = \frac{W_i^t}{\sum_{a_k \in A_t(u_i)} w_i^k} f(d) E_i^0 \tag{3}$$

C. User Allocation Cost

In MEC, user locations continuously change over time, and user mobility can lead to service interruptions. To prevent a decrease in user QoE due to service disruptions, it is necessary to migrate user tasks, which incur associated migration costs. We define the cost $l_{a_k}^t(u_i)$ generated by user migration task $a_k \in A_{u_i}^t$ as the span between the migration start time and the start time of the migration task on the target edge server. The computation is as follows:

$$l_{a_k}^t(u_i) = z_{a_k}^t(u_i) / v_{a_k}^t(u_i) + t_{com} \frac{W_i^t}{C_j^t} \tag{4}$$

Here, $z_{a_k}^t(u_i)$ represents the size of data transmitted between servers, while $v_{a_k}^t(u_i)$ represents the transmission rate. t_{com} represents the delay in waiting for the target server to initiate the service. C_j^t stands for the resource capacity of the destination edge server.

We aim to minimize the migration cost incurred by users during task migration in the user allocation process, defined as follows:

$$\mathbb{P}_1 : \min \sum_{i=1}^{|U_t|} \sum_{k=1}^{|A_t(u_i)|} l_{a_k}^t(u_i) \tag{5}$$

$$\text{s.t.} \quad C_1 : \sum_{i=1}^{|U_t|} \sum_{k=1}^{|A_t(u_i)|} w_k x_{i,j,k} \leq C_j^t \tag{5a}$$

$$C_2 : \sum_{j=1}^{|S|} x_{i,j,k} \leq 1 \tag{5b}$$

where C_1 represents that the total allocated resources for all tasks in server j are less than the resource capacity limit. C_2 represents that each task can only be executed within one server.

IV. Multilateral Collaborative User Allocation Algorithm

The algorithm mainly consists of three parts: Data Collection and Preprocessing, Generating Candidate Resource Schemes, and Obtaining the Optimal User Allocation Strategy. The process is depicted in Fig.3.

Data Collection and Preprocessing. Firstly, existing edge QoS data is analyzed and processed to extract dynamic user trajectory data and edge server QoS sample data to construct a merged dataset. Then, simulation processing is performed on the existing merged dataset to construct a simulation merged dataset that meets the experimental requirements.

Generating Candidate Resource Schemes. Firstly, the future geographical locations of users are predicted using a user trajectory prediction model. Then, based on the coverage area of edge servers, the set of edge servers that users can access the next time are identified. The hidden fitness relationship between each user and edge server in the server set is calculated, and the top M edge servers with the highest hidden fitness are extracted to construct a candidate station set for user allocation (see line 3 in Algorithm 1).

Obtaining the Optimal User Allocation Strategy. Firstly, the QoE model for users and the cost model for user allocation are constructed. These models are combined with weights to form a comprehensive value model for users. Subsequently, the user allocation problem is formulated as a knapsack problem, wherein the comprehensive value of all allocated users is translated into the knapsack's value, and the resource constraints of edge servers are equated to the knapsack's capacity limit. This problem is then optimized and solved using an enhanced adaptive ant colony algorithm to derive the optimal user allocation strategy (see line [4-15] in Algorithm 1).

A. Data Collection and Preprocessing

The objective of data collection and preprocessing is to merge and construct a dataset that satisfies the requirements of mobility awareness and resource constraints. In areas with deployed edge servers, numerous users move between different regions and access edge servers at various locations at different times. Each edge server concurrently processes service requests from many users and records the data of service calls made by mobile users, including the range of regions where mobile users are located, the types of devices used, the latitude and longitude information of the accessed edge servers, and

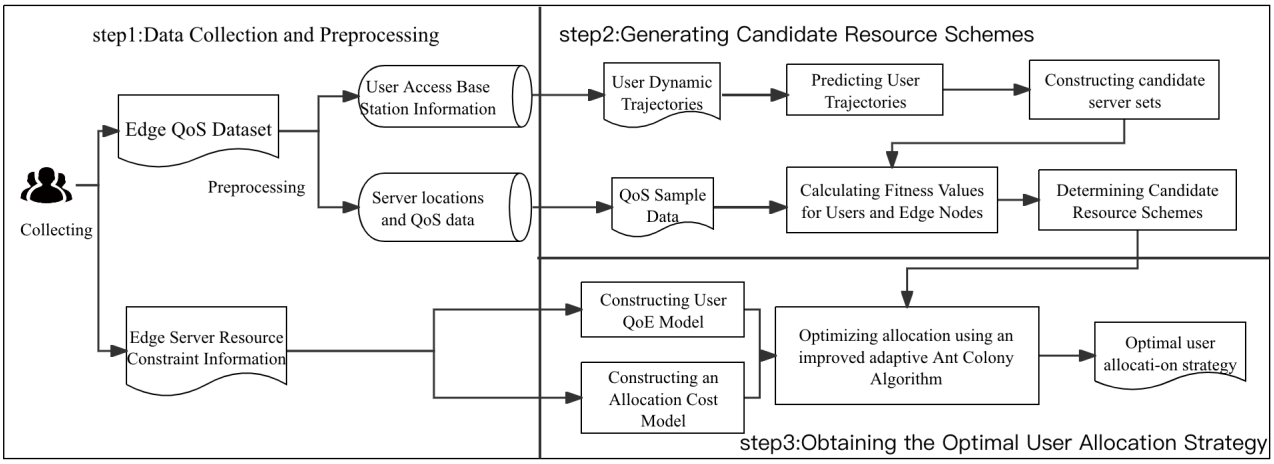

Fig. 3. The overall framework diagram of the method.

Algorithm 1 Multilateral Collaborative User Allocation Algorithm

Input: Trajectory of User, Task information
Output: The Optimal User Allocation Strategy

Initialize Population size P, Evaporation rate e, Pheromone deposition rate m
Initialize Pheromone matrix M
Predicting the position at time $t+1$ based on historical trajectories and obtaining a candidate set of servers J_{t+1}
for each $i = 1, 2, \ldots, N$ **do**
  for each $Ant = 1, 2, , \ldots, n$ **do**
    Ant select paths based on the concentration of pheromones
    Compute the fitness c_i of each Ant's path
    Update the global best solution
    if A new global best solution is found **then**
      Saving global best solution
    end if
    Evaporate all pheromones according to the evaporation rate
    Select the top k paths with the highest fitness and add pheromones to them
  end for
end for

the types of services called. The purpose of the data collection phase is to collect the location information of edge servers and the stored QoS historical data. Data preprocessing mainly involves filtering out invalid data, such as QoS samples with response times of -1, to make the experimental dataset more consistent with practical requirements.

B. Generating Candidate Resource Schemes

We employ the LSTM model to predict the geographical location of users at the next time. The historical trajectory information of users serves as the input to the trajectory prediction model, which is then propagated to subsequent layers to obtain predicted trajectory points. The user's historical trajectory points are represented as $Y(t) = \{lat_t, lng_t\}$. The prediction model is represented as:

$$Y(t+1) = \mathcal{F}\{Y(t), Y(t-1), ...Y(1)\} \tag{6}$$

Based on the coverage range of edge servers and the predicted future geographic locations of users, the set S_{t+1} of candidate edge servers that users can access at the next time step is computed. Each edge server S_i in this set satisfies:

$$(lng_m, lat_m) \in cov(s_i) \tag{7}$$

Next, we calculate the Mobility-awareness Fitness Value (MF) mapped to each edge server in the available server set S_{t+1} based on the coverage area of servers. MF represents the expected duration that a user remains within the coverage area of a particular edge server. As shown in Fig.4, the calculation of the $MF_{i,j}^t$ value between user u_i and server s_j is as follows:

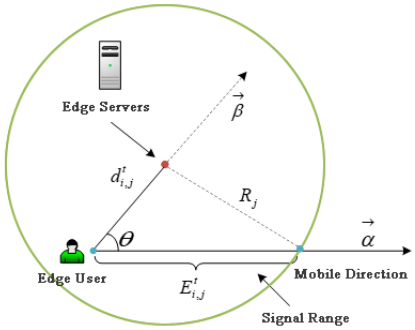

Fig. 4. Mobility Fitness Relationship between User and Edge Server.

$$MF_{i,j}^t = \frac{E_{i,j}^t}{v_i^t} \tag{8}$$

Where $E_{i,j}^t = \sqrt{R_j^2 - \left(d_{i,j}^t\right)^2 + 2R_j d_{i,j} \cos\theta}$ represents the expected movement distance of user u_i within the signal range of edge server s_j at time t, v_i^t the speed of the user u_i at time t, $d_{i,j}^t$ denotes the distance between the user u_i and edge server s_j at time t, and R_j represents the coverage range of the edge server. Subsequently, all MF mapped between users and edge servers are sorted and the pre-allocated candidate nodes $J_{t+1} = \{s_1, s_2, \cdots, s_n\}$ for users are formed.

C. Obtaining the Optimal User Allocation Strategy

The ant colony algorithm simulates the behavior of ants discovering paths while searching for food to find optimized paths in graphs. In the ant colony algorithm, the pheromone trail is a distributed numerical information that ants utilize for decision-making analysis. It has been widely applied to solve problems such as the traveling salesman problem (TSP) and the 0-1 Knapsack Problem. When considering the allocation of users under the constraint of edge server resources, we model the user allocation problem as a 0-1 Knapsack Problem, as shown in Table I.

TABLE I
MODEL MAPPING RELATIONSHIP

0-1 Knapsack Problem	User Allocation Optimization Problem
Value of the goods	Maximum weighted value after user allocation
The volume of the backpack	Resource constraints on base stations

The weighted value of each requested task allocated to various edge servers (i.e., the weighted sum of user QoE and allocation cost) is considered the value of each item placed into the knapsack. The resource constraints of each edge server are treated as the volume of the knapsack. Selecting appropriate items to maximize the total value of the knapsack ensures that the total weighted value of the user's various request tasks offloaded to the candidate resource set of edge servers is maximized.

Firstly, we have constructed a directed graph $G = (V, E)$, V represents the set of points, which consists of V_s and V_d, where V_s represents the nest location of the ants (i.e., user position), and V_d represents the location of the items (i.e., edge server positions in the candidate resource solutions). E is the set of edges, which includes the distance from the user u_i to each edge server in the candidate resource solutions.

Let $\tau_i (i = 0, 1, ..., n)$ be the pheromone on the path, where the value of the pheromone represents the concentration of the pheromone on the path. Due to the lack of data such as the time delay of the user's assignment to the target edge server when performing the initial user assignment, we only consider the distance between users and edge servers when initializing pheromone values. In calculating the pheromone on the path between the service invoked with the user and the edge server, it is represented by the following equation:

$$\tau_i = 1/d_{i,n} \tag{9}$$

where $d_{i,n}$ represents the distance from user u_i to the edge server s_n.

Secondly, We assume the route taken by the i-th ant in the colony in the k-th step is $L(i) = (0, t_1, t_2, t_3, \cdots, t_n)$, which means that the ants start from the origin and arrive at $t_1, t_2, t_3, \cdots, t_n$ in turn, the ants follow the transfer probability formula to choose t_{k+1} and choose the location with the largest transfer probability as the next step. If $\sum_{j=1}^{k+1} a_{i_j} \leq b$, where b represents the capacity of the knapsack (available resources of the edge server), and a_{i_j} represents the resources required for the j-th task called by u_i, if the capacity constraint of the knapsack is satisfied, then t_{k+1} is added to the knapsack, i.e., $L(s) = (0, t_1, t_2, t_3, ..., t_{k+1})$; otherwise, the ant stops walking and returns to the starting point.

The weighted sum of all service requests allocated to the edge server) given by $\sum_{i \in L(s)} c_i (s = 1, 2, ..., m)$, where c_i represents the weighted value of each item, calculated as follows:

$$c_i = \omega_1 \frac{E_i - E_{min}}{E_{max} - E_{min}} + \omega_2 \frac{l_{a_k}^t(u_i) - l_{a_{kmax}}^t(u_i)}{l_{a_{kmax}}^t(u_i) - l_{a_{kmin}}^t(u_i)} \tag{10}$$

ω_1 and ω_2 respectively represent the importance of the user's QoE and allocation cost, with $\omega_1 + \omega_2 = 1$.

The updating of pheromones mainly consists of two parts: strengthening the pheromones on arc $L(s)$ and evaporating the pheromones on other arcs. To prevent the algorithm from falling into a local optimal solution prematurely and facilitate the global search, we adopt an adaptive pheromone updating strategy to dynamically adjust the pheromone intensity on the paths searched by ants that are in a state of local convergence, making the user allocation result more reasonable. Therefore, the concentration of pheromones will be adjusted according to the following formula.

$$\begin{cases} \tau_{ij}(t+1) = (1-\rho)^{1+\omega(m)} \bullet \tau_{ij} + \Delta_{\tau ij}, & \tau \geq \tau_{max} \\ \tau_{ij}(t+1) = (1-\rho)^{1-\omega(m)} \bullet \tau_{ij} + \Delta_{\tau ij}, & \tau \leq \tau_{max} \end{cases} \tag{11}$$

where ρ and $1-\rho$ respectively represent the retention level and evaporation level of pheromones. $\Delta\tau_{ij}^k$ represents the residual pheromone concentration of the ants during the optimization process in the period t to $t+n$. $\omega(m) = k/l$, k is the convergence iteration count. Here, l is a constant used to adaptively adjust the strength of pheromones on each path according to the distribution of solutions, thereby updating the pheromones adaptively. This approach prevents the algorithm from prematurely falling into a local convergence, thus enhancing the global search capability of the ant colony algorithm. Finally, we choose the ant with the maximum total knapsack value, indicating that the user allocation strategy has the highest overall value. We select the set of edge servers within this value to collaborate in providing services for the user's requests.

V. EXPERIMENTS

The PyTorch 1.9.1 framework is used to implement the proposed method. The model is trained with a computer with

NVIDIA GTX1650Ti GPU, AMD Ryzen 7 CPU@2.90 GHz. All methods are compared under the same environment.

A. Experimental Setup

Experimental Data. This experiment mainly involves two datasets:

1) Dataset 1 is the Shanghai Telecom dataset[1]. This dataset consists of real geolocation information for 3,233 base stations and the invoked service records of 611,507 base stations, including the start and end times of service calls, server addresses (latitude and longitude), and user IDs.
2) Dataset 2 is a real-world service quality dataset released by the Chinese University of Hong Kong[2]. The dataset provides real service QoS data, recording the QoS information for 4,500 edge services invoked by 142 users across 64 different time slices (each time slice being 15 minutes apart).
3) Based on the aforementioned two datasets, a validation dataset for simulation was constructed. We simulated edge server resource capacity data and user resource request data. Each server's resource capacity data and user requests consist of four attributes: CPU, RAM, Memory, and Bandwidth. The number of resources required for each attribute is a random number between 1 and 5. The simulation parameters are shown in TableII.

TABLE II
SIMULATION DATASET PARAMETER SETTINGS

Parameters	Default Values	Range
Number of Edge Servers	60	
Number of Edge Users	160	
Number of Tasks	4	(1∼5)
Server Resource Capacity	(10,10,10,10)	(1∼10,1∼10,1∼10,1∼10)
User Resource Demand	(1,1,1,1)	(1∼10,1∼10,1∼10,1∼10)

B. Comparison Methods.

We compared the Ant Colony Optimization Algorithm for Edge User Allocation (EUA-ACO) with several baseline methods and some of the more effective service quality optimization methods in recent years to verify the superiority of EUA-ACO. These optimization methods include EUA-Random, EUA-GA, EUA-ABC, EUA-FOA [15], and EUA-ILP [9]. Their descriptions are as follows:

- EUA-Random: A random service quality optimization method. This method forms a user allocation strategy by randomly assigning edge servers based on user mobility while satisfying server resource constraints and signal coverage constraints.
- EUA-GA: A quality of service optimization method based on Genetic Algorithm (GA). This method searches for feasible edge user assignment strategies based on user mobility and iterative selection based on genetic algorithms, taking into account resource constraints and signal coverage constraints.
- EUA-ABC: A service quality optimization based on the Artificial Bee Colony algorithm (ABC). This method allocates users by simulating the process of a bee colony harvesting honey using the Artificial Bee Colony algorithm while considering resource limitations and signal coverage constraints.
- EUA-FOA: A service quality optimization method based on the Fruit Fly Optimization Algorithm (FOA). This method uses the FOA to allocate edge users to edge servers by simulating the predator-prey process of fruit flies, forming allocation strategies, and seeking optimal solutions.
- EUA-ILP: A service quality optimization method based on Integer Linear Programming (ILP). This method employs a heuristic approach based on integer linear programming to provide users with edge servers that offer sufficient computing resources from a set of candidates.
- EUA-ACO: This paper proposes an optimization method for user allocation. The method is based on an improved adaptive ant colony algorithm, which allocates edge servers to users by simulating the process of ant colony searching for food.

C. Experimental Results and Analysis.

Algorithm Performance. To verify the algorithm's convergence speed, we compared its performance under different numbers of iterations. As shown in Fig.5. It can be observed that the proposed method achieves the highest fitness value starting from the second iteration throughout the entire period. It remains relatively stable after the third iteration. The optimization effects of Random-EUA and EUA-GA are weaker, and their values remain stable in the later stage of the iteration process. The lower fitness value of EUA-GA is attributed to its susceptibility to getting trapped in local optima during the search process. The optimization performance of EUA-ILP and EUA-FOA strategies is relatively poor because they cannot guarantee the highest overall value of users in the region. EUA-ILP, being a heuristic method, stabilizes in value after the third iteration. EUA-FOA, based on swarm intelligence technology, obtains higher fitness values than EUA-ILP but exhibits slow convergence and is also prone to get trapped in local optima during the search process. Overall, the method proposed in this paper outperforms other methods in terms of convergence speed, achieving the highest fitness value and making user allocation decisions more quickly.

Algorithm Feasibility. To verify the speed of convergence of the algorithm, we conducted a set of experiments to analyze the feasibility of the method by comparing the user allocation rate and resource utilization rate. Fig.6 shows the user allocation rates for different algorithms. It can be observed that the user allocation rate for EUA-Random is relatively low, while EUA-ILP, EUA-GA, and EUA-FOA exhibit higher user allocation rates. EUA-ACO and EUA-ABC can successfully allocate a larger number of users within the experimental area. This discrepancy arises because EUA-Random only randomly allocates users to edge servers based on server resource

[1] $http://sguangwang.com/TelecomDataset.html$
[2] $http://wsdream.github.io/dataset/wsdream_dataset2.html$

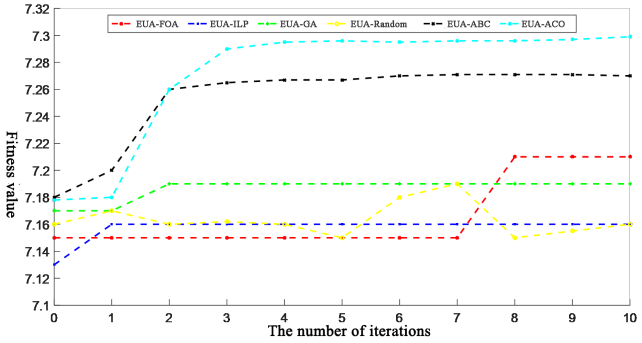

Fig. 5. Performance of Different Algorithms.

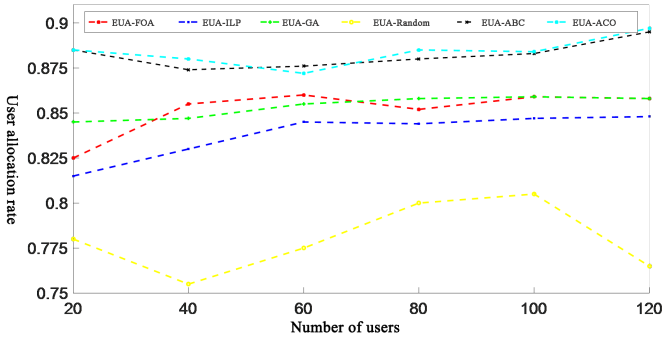

Fig. 6. User Allocation Rates Across Different User Scales.

constraints and signal coverage constraints without considering the overall resource utilization in the region, resulting in a poor user allocation rate. EUA-ILP, EUA-GA, and EUA-FOA do not jointly allocate all users in the region from an overall perspective using multiple edge servers. In contrast, EUA-ACO and EUA-ABC aim to allocate all users to edge servers at the edge of the region from a holistic viewpoint by coordinating multiple edge servers, resulting in a larger scale of successfully allocated users. As the user scale increases, the method proposed in this paper consistently maintains a high user allocation rate.

Next, we compared the resource utilization rates across different regions under different algorithms. Users are divided into four regions based on their location distribution, with 40 users in each region. The resulting regional division is illustrated in Fig.7. The resource utilization rates of different algorithms are shown in Fig.8. It can be seen that the EUA-ACO achieves the highest resource utilization rate, while the

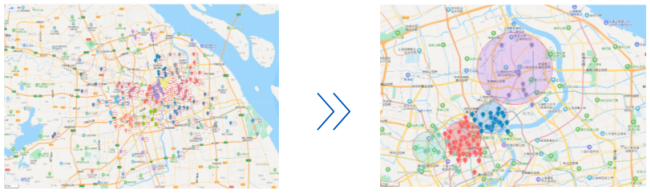

Fig. 7. User Distribution by Region.

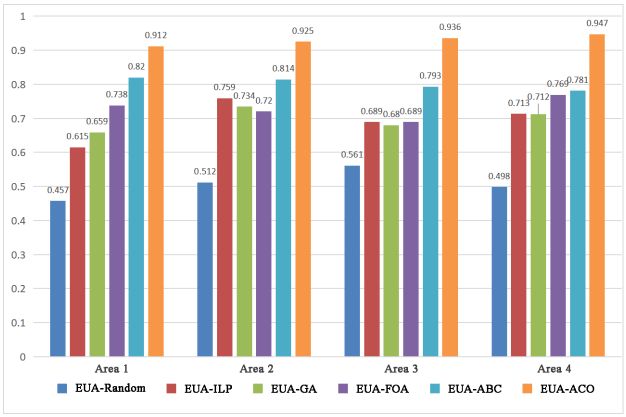

Fig. 8. Resource Utilization Rates for Different Algorithms.

resource utilization rates of the other methods are relatively low. The disparity arises because EUA-ILP, EUA-GA, and EUA-FOA optimize user allocation without considering minimizing allocation costs to the minimum extent. Conversely, EUA-ACO considers the collaboration of surrounding edge servers for resource sharing, thus improving resource utilization rates.

Optimization Effectiveness. The optimization effect of user allocation is specifically reflected in the user's quality of service experience, with the response time of the service being the most intuitive indicator. In this experiment, the users were divided into four groups of 40 people each based on their geographical distribution. The average response time of edge services before and after optimization was calculated for each group's user allocation optimization strategy. By comparing the response times of the original sample tasks for these four groups of users with the response times after optimizing user allocation, it can be seen that the optimization method proposed in this paper can reduce the average response time of services by approximately 6%. The changes in average response time before and after optimization for these four groups of users are shown in Table III.

TABLE III
THE CHANGES IN RESPONSE TIME BEFORE AND AFTER OPTIMIZATION IN EACH AREA.

Area	Average Response Time/s	After Optimization/s	Reduction Rate
1	2.3727	2.1944	7.51%
2	4.6156	4.2342	8.26%
3	5.6381	5.2914	6.15%
4	5.2351	4.9102	6.21%

VI. Conclusion

This paper proposes a collaborative user allocation method for multi-edge environments. This method utilizes users' future trajectory information and information about the coverage range of surrounding edge servers to dynamically obtain hidden fitness relationships among all users and edge servers within the region. This allows for the pre-allocation of users to perform tasks with higher fitness. The method considers

cooperation between mobile edge servers and addresses multi-resource joint constraint problems. An improved adaptive ant colony algorithm is used to optimize user allocation, aiming to maximize the total benefits obtained by all users within the region. Future work will consider user location privacy and security to avoid exposing sensitive information, thus ensuring the security of user data.

ACKNOWLEDGMENTS

This work is funded by the National Natural Science Foundation of China under Grant No.62272145 and No.U21B2016, the Natural Science Research Startup Foundation of Recruiting Talents of Nanjing University of Posts and Telecommunications (Grant No. NY223166), and the Australian Research Council's Discovery Projects funding scheme (DP220101823).

REFERENCES

[1] Abhishek Hazra, Pradeep Rana, Mainak Adhikari, and Tarachand Amgoth. Fog computing for next-generation internet of things: Fundamental, state-of-the-art and research challenges. *Computer Science Review*, 48:100549, 2023.

[2] Wazir Zada Khan, Ejaz Ahmed, Saqib Hakak, Ibrar Yaqoob, and Arif Ahmed. Edge computing: A survey. *Future Gener. Comput. Syst.*, 97(C):219–235, aug 2019.

[3] Olusola Adeniyi, Ali Safaa Sadiq, Prashant Pillai, Mohammad Aljaidi, and Omprakash Kaiwartya. Securing mobile edge computing using hybrid deep learning method. *Computers*, 13(1), 2024.

[4] M.P. Papazoglou. Service-oriented computing: concepts, characteristics and directions. In *Proceedings of the Fourth International Conference on Web Information Systems Engineering, 2003. WISE 2003.*, pages 3–12, 2003.

[5] Phu Lai, Qiang He, Mohamed Abdelrazek, Feifei Chen, John Hosking, John Grundy, and Yun Yang. Optimal edge user allocation in edge computing with variable sized vector bin packing. In Claus Pahl, Maja Vukovic, Jianwei Yin, and Qi Yu, editors, *Service-Oriented Computing*, pages 230–245, Cham, 2018. Springer International Publishing.

[6] Lei Yang, Bo Liu, Jiannong Cao, Yuvraj Sahni, and Zhenyu Wang. Joint computation partitioning and resource allocation for latency sensitive applications in mobile edge clouds. *IEEE TSC*, 14(5):1439–1452, 2021.

[7] Mengting Liu, F. Richard Yu, Yinglei Teng, Victor C. M. Leung, and Mei Song. Distributed resource allocation in blockchain-based video streaming systems with mobile edge computing. *IEEE TWC*, 18(1):695–708, 2019.

[8] Zhengyu Song, Yuanwei Liu, and Xin Sun. Joint radio and computational resource allocation for noma-based mobile edge computing in heterogeneous networks. *IEEE Communications Letters*, 22(12):2559–2562, 2018.

[9] Phu Lai, Qiang He, Guangming Cui, Xiaoyu Xia, Mohamed Abdelrazek, Feifei Chen, John Hosking, John Grundy, and Yun Yang. Edge user allocation with dynamic quality of service. In Sami Yangui, Ismael Bouassida Rodriguez, Khalil Drira, and Zahir Tari, editors, *Service-Oriented Computing*, pages 86–101, Cham, 2019. Springer International Publishing.

[10] Phu Lai, Qiang He, Mohamed Abdelrazek, Feifei Chen, John Hosking, John Grundy, and Yun Yang. Optimal edge user allocation in edge computing with variable sized vector bin packing. In Claus Pahl, Maja Vukovic, Jianwei Yin, and Qi Yu, editors, *Service-Oriented Computing*, pages 230–245, Cham, 2018. Springer International Publishing.

[11] Jiadi Liu, Songtao Guo, Quyuan Wang, Chengsheng Pan, and Li Yang. Optimal multi-user offloading with resources allocation in mobile edge cloud computing. *Computer Networks*, 221:109522, 2023.

[12] Seyedehmehrnaz Mireslami, Logan Rakai, Behrouz Homayoun Far, and Mea Wang. Simultaneous cost and qos optimization for cloud resource allocation. *IEEE Transactions on Network and Service Management*, 14(3):676–689, 2017.

[13] Yiming Miao, Gaoxiang Wu, Miao Li, Ahmed Ghoneim, Mabrook Al-Rakhami, and M. Shamim Hossain. Intelligent task prediction and computation offloading based on mobile-edge cloud computing. *Future Generation Computer Systems*, 102:925–931, 2020.

[14] Yan Ding, Kenli Li, Chubo Liu, Zhuo Tang, and Keqin Li. Budget-constrained service allocation optimization for mobile edge computing. *IEEE Transactions on Services Computing*, 16(1):147–161, 2023.

[15] Tingting Li, Wenqi Niu, and Cun Ji. Edge user allocation by foa in edge computing environment. *Journal of Computational Science*, 53:101390, 2021.

[16] Qiang He, Guangming Cui, Xuyun Zhang, Feifei Chen, Shuiguang Deng, Hai Jin, Yanhui Li, and Yun Yang. A game-theoretical approach for user allocation in edge computing environment. *IEEE Transactions on Parallel and Distributed Systems*, 31(3):515–529, 2020.

[17] Qinglan Peng, Yunni Xia, Zeng Feng, Jia Lee, Chunrong Wu, Xin Luo, Wanbo Zheng, Shanchen Pang, Hui Liu, Yidan Qin, and Peng Chen. Mobility-aware and migration-enabled online edge user allocation in mobile edge computing. In *2019 IEEE International Conference on Web Services (ICWS)*, pages 91–98, 2019.

[18] Chunrong Wu, Qinglan Peng, Yunni Xia, Yong Ma, Wangbo Zheng, Hong Xie, Shanchen Pang, Fan Li, Xiaodong Fu, Xiaobo Li, and Wei Liu. Online user allocation in mobile edge computing environments:a decentralized reactive approach. *Journal of Systems Architecture*, 113:101904, 2021.

[19] Mohammed Alreshoodi and John Woods. Survey on qoe\qos correlation models for multimedia services, 2013.

[20] Markus Fiedler, Tobias Hossfeld, and Phuoc Tran-Gia. A generic quantitative relationship between quality of experience and quality of service. *IEEE Network*, 24(2):36–41, 2010.

[21] Tobias Hoßfeld, Michael Seufert, Matthias Hirth, Thomas Zinner, Phuoc Tran-Gia, and Raimund Schatz. Quantification of youtube qoe via crowdsourcing. In *2011 IEEE International Symposium on Multimedia*, pages 494–499, 2011.

[22] Tolulope T. Oladimeji, Pradeep Kumar, and Mohamed K. Elmezughi. Performance analysis of high order close-in path loss model at 28 and 38 ghz. In *2023 Conference on Information Communications Technology and Society (ICTAS)*, pages 1–5, 2023.