Received: 27 July 2018 | Revised: 18 January 2019 | Accepted: 27 January 2019
DOI: 10.1002/spe.2685

SPECIAL ISSUE PAPER

WILEY

User allocation-aware edge cloud placement in mobile edge computing

Yan Guo[1] | Shangguang Wang[1] | Ao Zhou[1] | Jinliang Xu[1] | Jie Yuan[1] | Ching-Hsien Hsu[2]

[1] State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing, China

[2] Department of Computer Science and Information, Chung Hua University, Hsinchu, Taiwan

Correspondence
Shangguang Wang, State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing, China.
Email: sgwang@bupt.edu.cn

Funding information
National Science Foundation of China, Grant/Award Number: 61602054 and 61472047; BUPT Excellent Ph.D. Students Foundation, Grant/Award Number: CX2019211

Summary

Mobile edge computing is emerging as a novel ubiquitous computing platform to overcome the limit resources of mobile devices and bandwidth bottleneck of the core network in mobile cloud computing. In mobile edge computing, it is a significant issue for cost reduction and QoS improvement to place edge clouds at the edge network as a small data center to serve users. In this paper, we study the edge cloud placement problem, which is to place the edge clouds at the candidate locations and allocate the mobile users to the edge clouds. Specifically, we formulate it as a multiobjective optimization problem with objective to balance the workload between edge clouds and minimize the service communication delay of mobile users. To this end, we propose an approximate approach that adopted the K-means and mixed-integer quadratic programming. Furthermore, we conduct experiments based on Shanghai Telecom's base station data set and compare our approach with other representative approaches. The results show that our approach performs better to some extent in terms of workload balance and communication delay and validate the proposed approach.

KEYWORDS

edge cloud placement, mixed-integer quadratic programming, mobile edge computing, user allocation, workload balance

1 | INTRODUCTION

The Internet has witnessed three radical changes in the past decade: pervasive mobile devices, rapidly growing cloud computing, and the Internet of Things (IoT). Cloud computing is an emerging commercial computing model for enabling ubiquitous on-demand access to a shared pool of configurable computing resources (eg, computer networks, servers, storage, applications, and services).[1,2] With the development of cloud computing, it has reaped its field from enterprises to personal end users.[3,4]

Meanwhile, mobile devices powered by the rapidly developing mobile network have become increasingly pervasive, such as smartphones and tablets. According to Cisco's conservative estimate, there will be 1.5 mobile devices per capita, and the total number of smartphones will be over 50% of global devices and connections by 2021. Although mobile applications are becoming increasingly computational-intensive, many devices still have limited battery power and processing capabilities, and hence cannot support computational intensive tasks.[5] To overcome this problem, researchers propose mobile cloud computing that enables mobile devices to offload some workload to remote resource-rich clouds.[6-9] However, the remote clouds are geographically far away from mobile users, and it cannot guarantee fast and reliable access to services for end users in a mobile environment.

In addition, with the rapid development of IoT in the context of smart cities, high-data-rate sensors are becoming ubiquitous.[10,11] Massive sensing data streams are generated, which are geospatially distributed and make traditional cloud computing paradigm suffer long latency due to bandwidth bottleneck of the core network.

Motivated by these three trends, mobile edge computing is emerging.[12] In contrast to the centralized cloud computing or mobile cloud computing, mobile edge computing can provide a highly distributed computing environment. This distributed computing environment can be used to deploy applications and services as well as to store and process content in close proximity to mobile users.[13,14] It is an extension of cloud computing to the edge of networks. The edge clouds deployed at the edge of networks are important bridges between mobile users and remote clouds. The computational resources available at each edge cloud can reduce access latency and network bandwidth usage by providing services for mobile applications in the form of computation, storage, and software.

In mobile edge computing, how to place the edge clouds is a key issue. An appropriate edge cloud placement scheme can ensure the utilization of the computation resources of each edge cloud, the real-time of services provided by edge clouds, and the quality of experience of mobile users. Most existing research works focused on offloading workloads of mobile users in mobile edge computing, assuming that the edge clouds have already been placed.[6,15-17] Little attention has been paid to the placement of edge clouds. Although there are few related studies on edge cloud placement, there are a few research works on the cloudlets placement in recent years.[5,18,19] The cloudlet is a resource-rich server cluster colocated with wireless access points in a local network, and mobile users can offload their tasks to local cloudlets for processing. Both of the cloudlet and the edge cloud are the alternative solutions of powerful remote clouds, therefore, they are considered as same in this paper. Although the aforementioned approaches are effective, they do not consider the workload balance[5] or the communication delay of mobile users when they obtain services from edge cloud at the same time.[18] In addition, the algorithms to solve the placement problem always are not scalable.

In this paper, we study the edge cloud placement problem, which is to place the edge clouds at the candidate locations and allocate the mobile users to the edge clouds with the objective to balance the workload between edge clouds and minimize the communication delay between mobile users and the edge clouds serving the users. Note that edge clouds are placed at some base station locations for mobile user access by endowing the existed infrastructure with cloud functionalities due to the cost and convenience of the deployment.

Compared to the existing study, we focus on not only the minimum of the communication delay but also the balance of workload between edge clouds. Then, we formulate the user allocation-aware edge cloud placement problem as a multiobjective optimization problem and prove that this multiobjective optimization problem is an NP-hard problem. To find the optimal edge cloud placement solution, we propose an approximate edge cloud placement approach, which combines the K-means algorithm[20] and mixed-integer quadratic programming algorithm.[21] To evaluate the performance of our approach, we experiment based on Shanghai Telecom's base station data set and compare our approach with other representative approaches in terms of workload balance and communication delay. The experimental results show that our approach outperforms several representative approaches to some extent.

The remainder of this paper is organized as follows. Section 2 introduces the related work in this area. Section 3 introduces the system model and the definition of the edge cloud placement problem. Section 4 describes our approach in details including the formulation of edge cloud placement optimization problem and the approach of finding the optimal edge cloud placement solution. Section 5 demonstrates the benefits of our approach according to experimental evaluation. At last, Section 6 concludes the paper and discusses the future work.

2 | RELATED WORK

Before the emergence of mobile edge computing, there are several similar placement problems that have been researched in other networks. For example, Qiu et al[22] explored the problem of Web server replica placement in content distribution networks that offer hosting services to Web content providers. They reduced the Web server replica placement as a K-median problem, which is to choose M replicas (or hosting services) among N candidate sites ($N > M$), and then described graph theoretic formulations of the replica placement problem. To solve this problem, they developed several algorithms, such as tree-based algorithm, greedy algorithm, random, and a super-optimal algorithm based on Lagrangian relaxation with subgradient optimization. Yin et al[23] adopted a new K-means-clustering–based algorithm to select media server locations and identify the optimal matching between clients and media servers in the multimedia environment. They also took advantage of the latest network coordinate technique to reduce the workloads when obtaining the global network information for server placement. Besides, they optimized the trade-off between the service delay performance and the deployment cost under the constraints of client location distribution.

Although the placement problems in different environment can be reduced to a general K-median problem,[24] they are different essentially. In content distribution networks, the replica server or cache server is a mirror of the remote server, and it is to offer content delivery to clients while retaining efficient and balanced resource consumption. The remote users obtain services from the cache/replica server, this model reduces the bandwidth of the remote access, shares the network traffic, and reduces the load of server in the original Web site. However, the edge cloud in mobile edge computing is more powerful. It is a cloud at the edge of network, and it can provide more computational resources to remote users, such as applications and services, storage, and content process. Therefore, different servers in different network environments lead to different placement problems. For example, the objective function in content distribution networks is to minimize the latency, bandwidth consumption, energy consumption, and cost, but not the workload balance or the capacity constraints of edge clouds.

Although there are few related studies on edge cloud placement in mobile edge computing environment, there are some research works on the cloudlets placement in recent years.[5,18,19] For example, Jia et al[18] studied cloudlet placement and mobile user allocation to the cloudlets in a wireless metropolitan area network. They proposed two heuristic algorithms for the K cloudlet placement problem, one is a simple Heaviest-AP first algorithm and the other is a density-based clustering algorithm that surmounts the shortcomings of the heaviest-AP first algorithm. Xu et al[5] studied the cloudlet placement problem in a large-scale wireless metropolitan area network. They formulated the problem as a capacitated cloudlet placement problem that places K cloudlets to some strategic locations in the wireless metropolitan area network. Then, they proposed a heuristic algorithm for the problem to minimize the average communication delay between mobile users and the cloudlets serving the users. Note that the cloudlet and the edge cloud can be seen the same in placement problem as mentioned earlier.

Although the aforementioned approaches are effective, they do not consider the workload balance of edge clouds in the mobile edge network[5] or the communication delay of remote users.[6] Inspired by this, our approach considers the communication delay and the workload balance and formulates the edge cloud problem as a multiobjective problem.

3 | SYSTEM MODEL AND PROBLEM DEFINITIONS

3.1 | System model

The mobile edge network can be considered as a bipartite network $G = (V \cup S, E)$ consisting of many base stations and a set of candidate locations for edge clouds, where V is the set of base stations, S is the set of candidate locations of edge clouds, and E is the set of links between two base stations or between a base station and an edge cloud. For simplicity, all the locations of base stations are regarded as the candidate locations for edge clouds, ie, $S = V$, and each edge cloud is colocated with a base station. In addition, it is assumed that there are k edge clouds to be placed to k locations in S, and the computational capacity of each edge cloud is the same. Note that the proposed approach can also be applied to the placement problem, where the computational resources of each edge cloud is different, by redefining the workload balance.

Mobile users get services by sending requests to base stations, and the base stations offload their tasks to the edge clouds for providing services for mobile users quickly. An edge cloud is responsible for a subset of base stations, and a base station only links with a single edge cloud. There are two specific problems to be solved in user allocation-aware edge cloud placement problem, one is the placement of the edge clouds, and the other is the allocation of mobile users, that is, the allocation of base stations.

For edge cloud placement problem, we focus on the long-term effect of the edge cloud placement scheme. The two evaluation indexes used in this paper are workload balance and communication delay. (1) The user request to a base station is the workload of the base station, and the sum of user request offloaded to an edge cloud through base stations is the workload of the edge cloud. The balance of workload guarantees that there is no situation where some of the edge clouds are overloaded while others are underloaded or even idle. The balanced workload also guarantees the similar computation delay and queuing delay for users served by each edge cloud because the computational capacity of each edge cloud is the same. (2) In this paper, the communication delay is the delay of load from the base station and its edge cloud. Note that, if an edge cloud is colocated with a base station, the users at the base station will have the minimum communication delay. Table 1 shows the symbols used in this paper.

TABLE 1 Symbols

Symbols	Notations
D	A set of edge clouds
V	A set of base stations
n	The numbers of base stations in V
k	The numbers of edge clouds in D
v	A base station in V
d	An edge cloud in D
w	The workload of base station or edge cloud
l	The location of base station or edge cloud
T	An edge cloud placement scheme which contains k locations of edge clouds
C	A base station allocation scheme which contains k subsets of base stations
Bw	The workload balance of an edge cloud placement problem solution
Bw'	The workload balance of a base station allocation problem solution
ρ	The communication delay between a base station and the linked edge cloud
Da	The communication delay of an edge cloud placement problem solution
Da'	The communication delay of a base station allocation problem solution

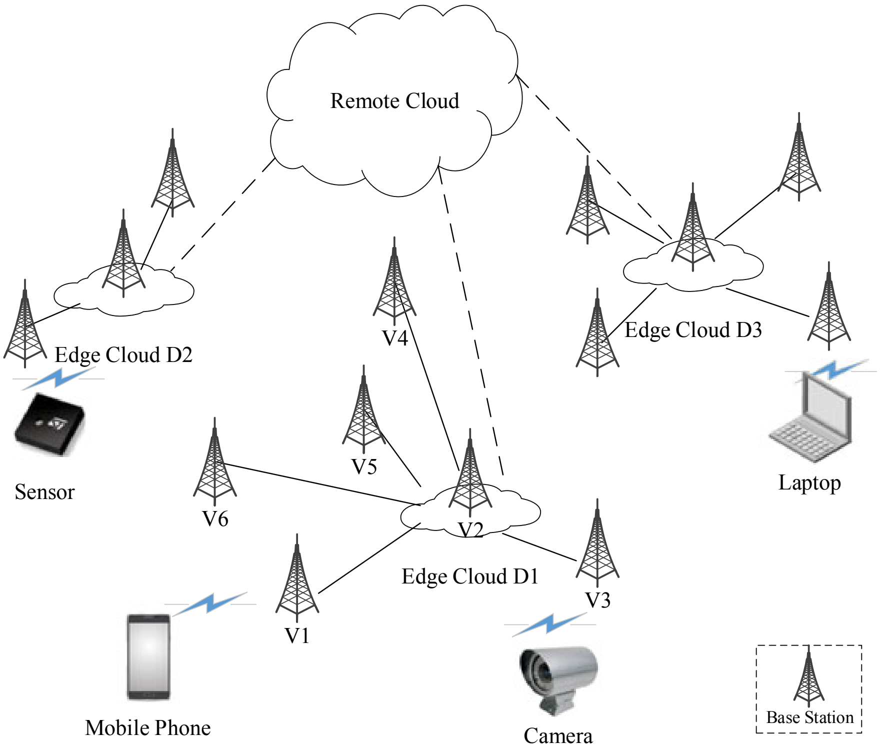


FIGURE 1 A model of edge cloud placement in mobile edge computing [Colour figure can be viewed at wileyonlinelibrary.com]

To illustrate the edge cloud placement problem directly, we introduce an example in Figure 1. Take edge cloud *D1* as an example, it is responsible for six base stations surrounding itself, and the relations are represented by the links between *D1* and the base stations. The mobile users connect to one base station directly, and this connection is used as an intermediate node to access services from an edge cloud. Note that the links between two base stations are not shown in Figure 1 because these links are not our focus.

3.2 | Problem definitions

The edge cloud placement problem in mobile edge environment $G = (V \cup S, E)$ is defined as follows. Let n and m be the numbers of base stations and links in V and E, ie, $|V| = n$, $|E| = m$, and $|S| = n$. Let $\{v_1, v_2, \dots, v_n\}$ be the base stations, and, for each base station $v_i(1 \le i \le n)$, $w(v_i)$ represents the workload of the base station and $l(v_i)$ represents the location of the base station. Let $e_{i,j} \in E$ be the communication delay of each link between two base stations v_i and v_j. Let $D = \{d_1, d_2, \dots, d_k\}$ be the k edge clouds to be placed at the candidate locations in S. Obviously, $k \le n$, and, for each edge cloud $d_j(1 \le j \le k)$, $w(d_j)$ represents the workload of the edge cloud, $l(d_j)$ represents the location of the edge cloud, and c_j represents the subset of base stations, which link with the edge cloud. Let $T = \{l(d_1), \dots, l(d_j), \dots, l(d_k)\}$ be an edge cloud placement scheme, $C = \{c_1, \dots, c_j, \dots c_k\}$ is a base station allocation scheme, therefore, (T, C) is a solution of edge cloud placement problem. As the workload of edge cloud is the user requests offloaded through base stations, which link with the edge cloud, $w(d_j) = \sum_{v_i \in c_j} w(v_i)$.

The optimization goal of the user allocation-aware edge cloud placement problem is the balance of workload between edge clouds and the minimum of the communication delay between the edge clouds and base stations.

Due to the same computational capacity of each edge cloud, the workload balance can be represented by the variance of the workload of edge clouds. Let $Bw(T, C)$ be the workload balance between edge clouds in an edge cloud placement solution, and it can be formulated as follows:

$$Bw(T, C) = \frac{\sum_{j=1}^{k} \left(w(d_j) - \overline{w}(d)\right)^2}{k}, \tag{1}$$

$$\overline{w}(d) = \frac{\sum_{j=1}^{k} w(d_j)}{k}, \tag{2}$$

where $\overline{w}(d)$ is the average value of the workload of all edge clouds. The definition indicates that the smaller the value of $Bw(T, C)$ is, the more balanced the workload of edge clouds is.

Each edge cloud is placed at the location of a base station in the existing network topology, so the transmission path between a base station and its edge cloud is composed of the links in E. Let $\rho(v, d)$ be the communication delay between a base station v and the linked edge cloud d, it is the value of the shortest path between base station v and the base station at location $l(d)$ in the weighted graph E. Let $Da(T, C)$ be the communication delay between base stations and edge cloud in an edge cloud placement solution, it is defined as the average of all the communication delay between base stations and edge clouds, and it can be formulated as follows:

$$Da(T, C) = \frac{\sum_{d_j \in D} \sum_{v_i \in c_j} \rho(v_i, d_j)}{n}. \tag{3}$$

Based on the aforementioned definitions, the user allocation-aware edge cloud placement problem can be formulated briefly as follows:

- Input: finite set of base stations S, the number of edge cloud k;
- Output: the locations of edge clouds $T \in S$, a base station allocation scheme $C_j\,(1 \le j \le k)$;
- Goal: minimize the workload balance $Bw(T, C)$ and the communication delay $Da(T, C)$.

Lemma 1. *The edge cloud placement problem in mobile edge environment is an NP-hard problem.*

Proof. We reduce the metric K-median problem[25] to the placement location problem of edge clouds as follows. Consider the metric K-median problem in a complete metric undirected graph $G' = (V', E')$, $V'$is the locations of each client $v'_i \subset V'$, the $d_{ij} \subset E'$ is the cost of the facility placed at location v'_j for serving the client v'_i. The metric K-median problem is to choose K locations for facilities, which provide services for clients, and the overall goal is to minimize the cost.[25] Construct a mobile edge network $G = (V \cup S, E)$ from G', where $V = V'$, $S = V'$, and $E = E'$. The computation resource of each edge cloud is sufficient for all the base stations, in other words, the workload balance can be ignored. In this case, an optimal solution to the edge cloud placement problem in G also is an optimal solution to the G'. As we all know, the metric K-median problem is an NP-hard problem,[25] therefore, the edge cloud placement problem is an NP-hard problem.

Due to the difficulty of solving this NP-hard problem directly, we propose an approximation approach to find the optimal user allocation-aware edge cloud placement solution. □

4 | OUR APPROACH

4.1 | Edge cloud placement optimization

In this section, we formulate the user allocation-aware edge cloud placement problem as a mixed-integer quadratic programming problem.

First, we introduce two binary decision variables: $y_l \in \{0,1\}(1 \leq l \leq n)$ and $x_{i,l} \in \{0,1\}(1 \leq i, l \leq n)$. y_l is to indicate whether there is an edge cloud located at the location of v_l, where $y_l = 1$ if an edge cloud is placed at $l(v_l) \in S$, $y_l = 0$ otherwise. There are k edge clouds to be placed in the edge cloud placement problem, therefore, $\sum_{l=1}^{n} y_l = k$. $x_{i,l}$ is to indicate whether the base station v_i is linked with the edge cloud located at $l(v_l) \in S$, where $x_{i,l} = 1$ if the base station v_i is allocated to the edge cloud located at $l(v_l) \in S$, $x_{i,j} = 0$ otherwise. Assume that the edge cloud located at $l(v_l)$ is d_j, so $v_i \in C_j$. Each base station must have one edge cloud and can only be allocated to an edge cloud at a scheme, therefore, $\sum_{l=1}^{n} x_{i,l} = 1$ and $x_{i,l} \leq y_l$.

Second, we construct the fitness function. A set of candidate value of $\{x_{i,l}, y_l\}$ corresponds to an edge cloud placement scheme (T, C). The workload of edge cloud d_l is $w(d_l) = \sum_{i=1}^{n} w(v_i) x_{i,l}$, and the workload balance Bw and the communication delay Da of this scheme are as follows:

$$Bw(T,C) = \frac{\sum_{l=1}^{n} \left(w(d_l) - \overline{w}(d)\right)^2 - (n-k) \times \overline{w}(d)^2}{k}, \tag{4}$$

$$Da(T,C) = \frac{\sum_{l=1}^{n} \sum_{i=1}^{n} \rho(i,l) x_{i,l}}{n}. \tag{5}$$

To obtain an overall optimization index, we transform the edge cloud placement problem into a single-objective optimization problem by employing the simple additive weighting model. The fitness function is as follows:

$$F(T,C) = \mu \frac{Bw(T,C)}{B\max} + (1-\mu) \frac{Da(T,C)}{D\max}, \tag{6}$$

with

$$B\max = \frac{\left(\sum_{i}^{n} w(v_i) - \overline{w}(d)\right)^2 + (k-1) \times \overline{w}(d)^2}{k}, \tag{7}$$

$$D\max = \max_{i,l} \rho(i,l), \tag{8}$$

where $B\max$ represents the maximum value of the workload balance, $D\max$ represents the maximum value of the communication delay, and μ represents the weight of the workload balance. Through the mathematical transformation, $\frac{Bw(T,C)}{B\max}$ is limited in (0, 1] and $\frac{Da(T,C)}{D\max}$ is limited in [0, 1].

Finally, the user allocation-aware edge cloud placement problem can be formulated as follows:

$$\mathbf{P1}: \qquad \text{Min} \quad F(T,C), \tag{9}$$

$$\text{subject to} \sum_{l=1}^{n} y_l = k, \tag{10}$$

$$\sum_{l=1}^{n} x_{i,l} = 1, \tag{11}$$

$$\sum_{l=1}^{n} x_{i,l} = 1, \tag{12}$$

$$x_{i,l} \leq y_l, x_{i,l}, y_l \in \{0,1\}. \tag{13}$$

4.2 | Finding the optimal solution

The problem formulated in the last section is a mixed-integer quadratic programming problem with high complexity and a huge set of feasible solutions. To find the optimal solution of the user allocation-aware edge cloud placement problem, we propose an approximation approach, which contains two steps, one is the determination of the edge cloud locations according to the communication delay and the other is the allocation of base stations under a scenario where edge clouds have been placed.

Assumed that the edge clouds have been placed, $\{y_l\}$ in P1 is determined, therefore, the P1 can be simplified as follows:

$$\mathbf{P2}: \quad \text{Min } F'(C), \tag{14}$$

$$\text{subject to } \sum_{j=1}^{k} x'_{i,j} = 1, \tag{15}$$

$$x'_{i,j} \in \{0,1\}, \tag{16}$$

where $x'_{i,j}$ $(1 \leq i \leq n, 1 \leq j \leq k)$ is a decision variable of base station allocation problem to indicate whether the base station v_i is linked with the edge cloud d_j, it is the reduced matrix of $x_{i,l}(1 \leq i, l \leq n)$ and their relationship is as follows:

$$x_{i,l} = \begin{cases} x'_{i,j}, & y_l = 1 \ \ and \ \ l\left(d_j\right) = l(v_l); \\ 0, & \text{others}. \end{cases} \tag{17}$$

The objective function of base station allocation problem is as follows:

$$F'(C) = \mu \frac{Bw'(C)}{B\max} + (1-\mu)\frac{Da'(C)}{D\max}, \tag{18}$$

with

$$Bw'(C) = \frac{\sum_{j=1}^{k} \left(\sum_{i} w(v_i) x'_{i,j} - \overline{w}(d) \right)^2}{k}, \tag{19}$$

$$Da'(C) = \frac{\sum_{j=1}^{k} \sum_{i=1}^{n} \rho(i,j) x'_{i,j}}{n}. \tag{20}$$

The problem P2 is still a mixed-integer quadratic programming problem, however, compared with P1, the number of decision variables declines from $O(n^2)$ to $O(n)$.

Our proposed approach is as shown in Algorithm 1, it includes the following two steps. The first step is to select k locations for the edge clouds by using K-means algorithm.[20] We cluster n base stations into k clusters by minimizing the communication delay, and the centers of the k clusters are the locations for edge clouds $\{y_l\}$. Lines 1 to 9 in Algorithm 1 is the location selection phase, we select k location from S as initial edge cloud locations and allocate n base stations to the k clusters according to the communication delay until the k clusters remain the same.

The second step is to allocate base stations where edge clouds have been placed. After the determination of the edge cloud locations, the edge cloud placement problem can be simplified to a base station allocation problem. Therefore, P1 can be simplified to P2. Lines 10 and 11 in Algorithm 1 is the base station allocation phase, we solve the simplified user allocation problem by using a mixed-integer quadratic programming algorithm solver.[21] The principle of this solver is the Boolean quadric polytope cutting plane method.

After the two steps, we can get the location of edge clouds and the allocation of base stations to edge clouds.

Algorithm 1 The user allocation-aware edge cloud placement algorithm

```
Input: the workload of the base station $w(v_i)$, the location of the base station $l(v_i)$, the number of edge cloud $k$
Output: the locations of edge clouds $T$, a base station allocation scheme $C$
Select $k$ initial locations randomly $\boldsymbol{K}$
Repeat:
  for $v_i(1 \le i \le n) \in V$
      $x_{i,l} = 1$ if $l = \underset{k \in K}{\text{argmin}}\{\rho(i,k)\}$
  end for
  for $k \in K$
      $k' = \textit{the center of}\ \{v_i | x_{i,k} = 1\}$
  end for
Until $K' = K$
$y_l = 1\ \textit{iff}\ l \in K$
Solve P2
```

5 PERFORMANCE EVALUATION

To evaluate our approach, we implement it based on a real-world data set, and then compare our approach with other approaches in terms of the workload balance and the communication delay. Extensive experimental results indicate that our approach is superior to other traditional approaches. Moreover, we also study the placement parameter in our approach.

5.1 Data set description

The real-world data set used in our experiments is the Shanghai Telecom's base station data set. It contains internet information of mobile users through accessing 3233 base stations and the exact locations of these base stations. Figure 2 shows the distribution of 3233 base stations. More specifically, the data set contains 4.6 million call records and 7.5 million flow records of about 10 thousand mobile users during six successive months. Each call/flow record contains the detailed start time and end time of accessing base station for each mobile user. Table 2 describes partial information of 10 base stations, which are selected randomly from the data set, the workload of each station is the total request time calculated according to the start time and end time of their mobile users. From Table 2, we can see that there exists a great workload imbalance among these base stations. If the edge cloud placement problem only focus on the communication delay without the workload, it will lead to an unbalance workload between edge clouds.

TABLE 2 Information of a part of base stations

Base Station ID	Longitude	Latitude	User Number	Workload (min)
11	121.422303	31.180175	14	18 841
277	121.306923	31.206547	21	25 506
330	121.369095	31.121363	185	252 354
361	121.387532	31.324464	22	29 467
1349	121.448422	31.162868	589	706 841
1448	121.471519	30.824719	103	133 736
1889	121.768142	31.16872	31	42 071
1919	121.341904	30.733903	476	613 174
1994	121.009542	31.099755	55	69 600
2564	121.513919	31.246946	335	448 576
2978	121.182589	31.152749	86	113 609

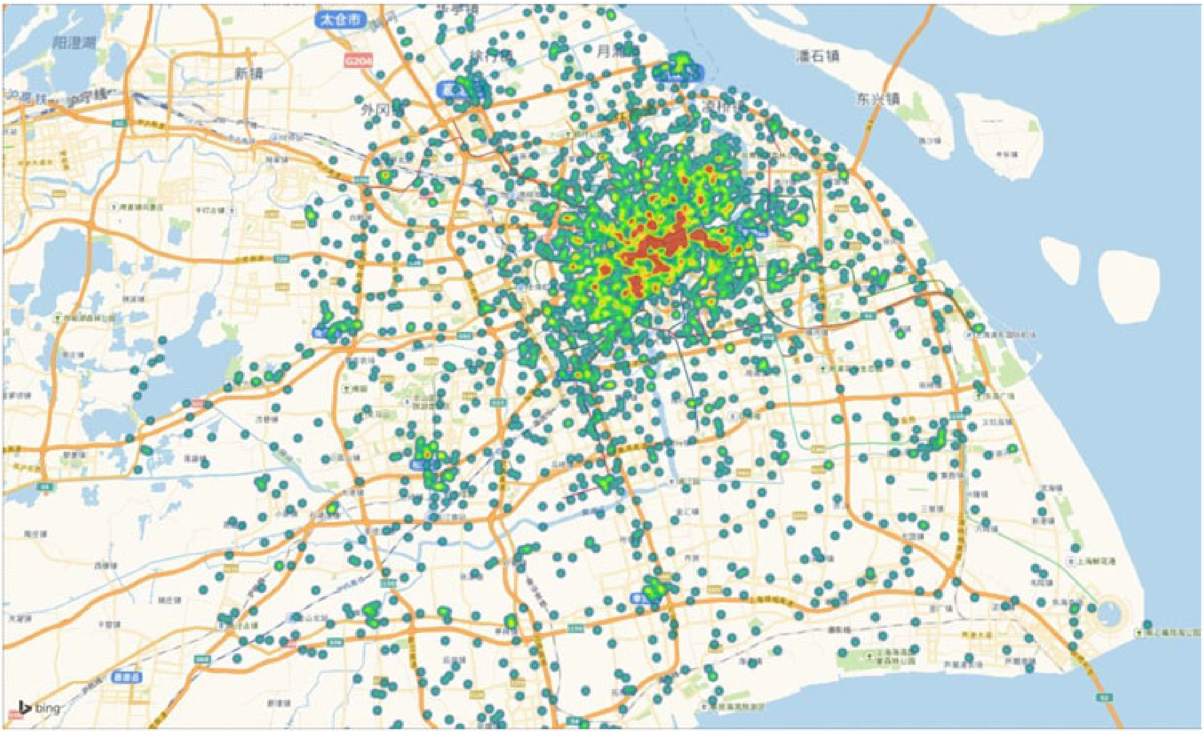


FIGURE 2 Distribution of 3233 base stations in Shanghai [Colour figure can be viewed at wileyonlinelibrary.com]

5.2 | Experiments setup

To evaluate the performance of different approaches in terms of the workload balance and the communication delay, three experiments are designed: (1) the number of edge cloud k is from 5 to 30, whereas the number of base stations n is 200; and (2) the number of base stations n is from 20 to 200, whereas the number of edge cloud k is 10. In addition, we analyze the placement parameter of our approach. To study the edge cloud placement ratio R, we design an experiment where the ratio R is from 0.05 to 0.15, and the number of base stations n is 200. In all experiments, the weight of the workload balance μ is 0.5.

All the experiments are conducted on the same computer with an Intel(R) Xeon(R) 2.4 GHz processor, 32.0 GB of RAM, and on Python 3.5 with source code.

5.3 | Compared approaches

In order to evaluate the performance of our edge cloud placement approach, we compare it with the following placement approaches.

1. **Top-K.** This approach is to select the locations of the k busiest base station for edge clouds, ie, the workload of each selected base station is in top-k. Each base station is allocated to the edge cloud, which is the closest to the base station.
2. **K-means.** This approach is a cluster algorithm that only considers the communication delay and not the workload balance. It clusters n base stations into k clusters by minimizing the overall communication delay, the centers of the k cluster are the locations for edge clouds, and the clusters are the allocation scheme of base stations.
3. **Random.** This approach is to select the k base station locations randomly for edge clouds. Each base station is allocated to the edge cloud, which is the closest to the base station.

5.4 | Comparison results with the number of edge clouds

Figure 3A and Figure 3B show the comparison results in terms of the workload balance and the communication delay, respectively, with different numbers of edge clouds ranging from 5 to 30. As shown in Figure 3, the workload balance

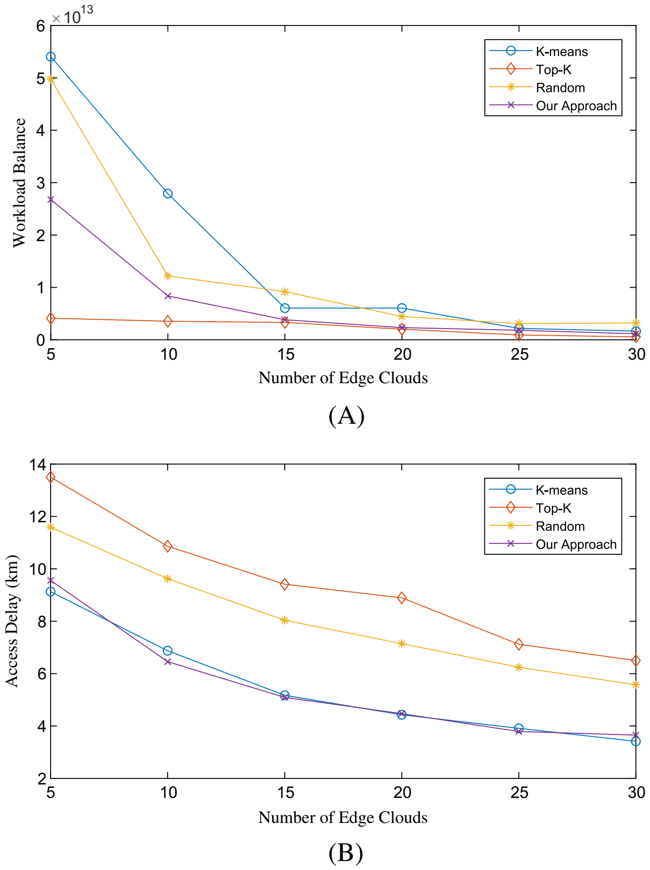


FIGURE 3 Comparison results with respect to the number of edge clouds; A, Workload balance for the edge cloud placement result with respect to the number of edge clouds; B, Communication delay for the edge cloud placement result with respect to the number of edge clouds [Colour figure can be viewed at wileyonlinelibrary.com]

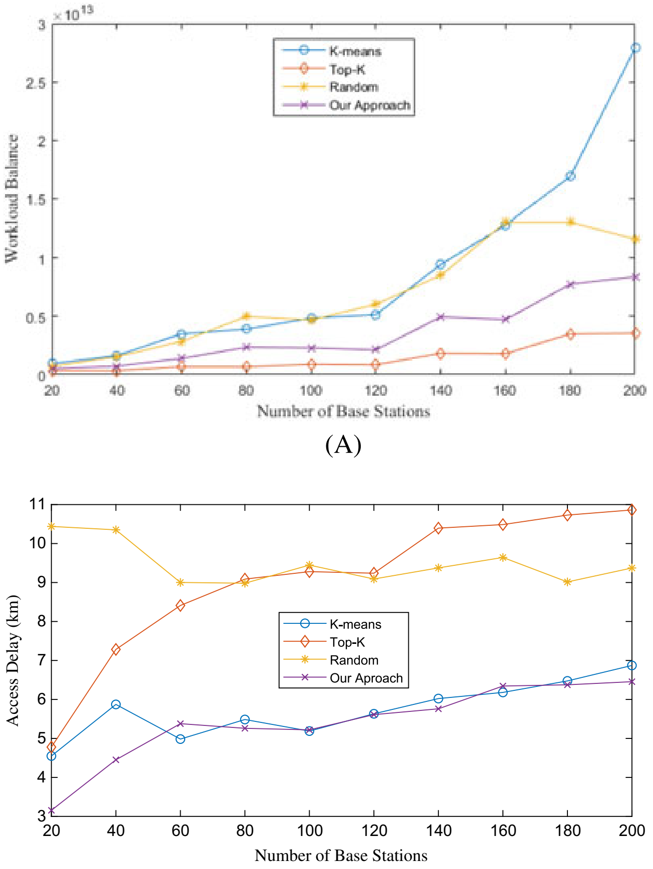
TABLE 3 The overall objective values of each edge cloud placement approach with respect to the number of edge clouds

k	K-means	Top-K	Random	Our Approach
5	0.1048	0.0723	0.1125	**0.0772**
10	0.0887	0.0614	0.0720	**0.0486**
15	0.0430	0.0566	0.0661	**0.0363**
20	0.0445	0.0521	0.0522	**0.0310**
25	0.0296	0.0400	0.0454	**0.0273**
30	0.0261	0.0357	0.0455	**0.0246**

reduces rapidly with increasing the number of edge clouds, and the communication delay reduces gradually with increasing the number of edge clouds. The workload balance of our approach is better than Random and K-means, and it is not as good as Top-K. However, the communication delay of our approach is better than Top-K and Random, and it is nearly identical with K-means.

Table 3 shows the overall performance of each edge cloud placement approach with different numbers of edge clouds. As shown in Table 3, except for the results when the number of edge clouds is 5, the overall performance of our approach is prior to the other approaches. In a word, although the performance of our approach is not all the best in terms of workload and communication delay, the overall performance of our approach is prior to other approaches to some extent with the different numbers of edge clouds.

Note that, when the number of edge clouds is 5, because the edge cloud placement ratio is too small, the workload of Top-K is more balanced than other approaches. Hence, the overall performance of Top-K is better than our approach in this case.



(A)

(B)

FIGURE 4 Comparison results with respect to the number of base stations; A, Workload balance for the edge cloud placement result with respect to the number of base stations; B, Communication delay for the edge cloud placement result with respect to the number of base stations [Colour figure can be viewed at wileyonlinelibrary.com]

5.5 | Comparison results with the number of base stations

Figure 4A and Figure 4B show the comparison results in terms of the workload balance and communication delay, respectively, with different numbers of base stations ranging from 20 to 200. As shown in Figure 4, the workload balance increases gradually with increasing the number of base stations because the number of base stations served by an edge cloud is larger and the difference of workload among these edge clouds is greater. Like the performance with different numbers of edge clouds, the workload balance of our approach is better than Random and K-means, and it is not as good as Top-K. However, the communication delay of our approach is better than Top-K and Random, and it is nearly identical with K-means.

Table 4 shows the overall performance of each edge cloud placement approach with different numbers of base stations. As shown in Table 4, except for the results when the number of base stations is 20 and 40, the overall performance of our approach is prior to the other approaches. In a word, although the performance of our approach is not all the best in terms of workload and communication delay, the overall performance of our approach is prior to other approaches to some extent with the different numbers of base stations.

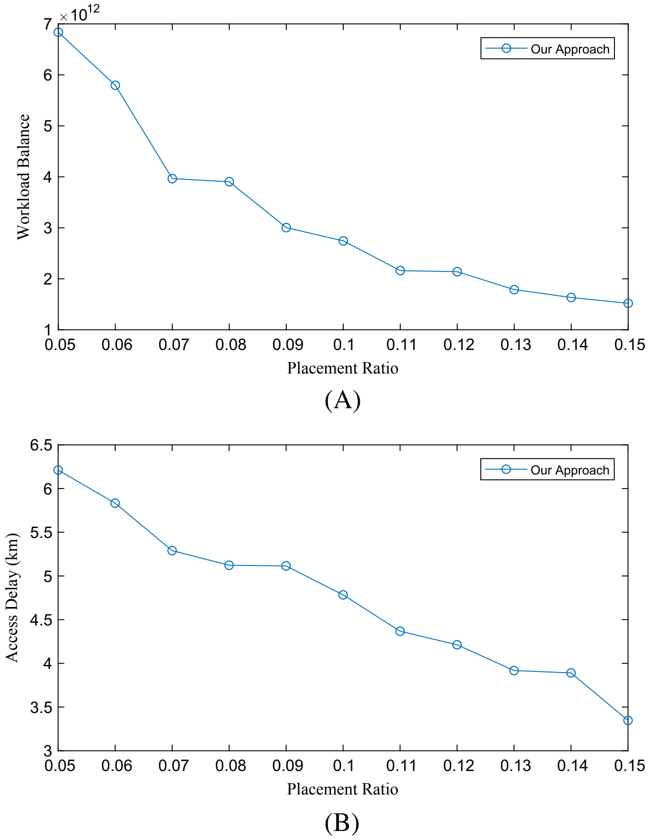
Note that, when the number of base stations is 20 and 40, the edge cloud placement ratio is too high, the communication delay of Top-K is small. Hence, the overall performance of Top-K is better than our approach in this case. However, with the increasing number of base stations, the communication delay of Top-K increases rapidly.

5.6 | Parameter analysis of the placement ratio

In reality, it is difficult to decide the number of edge clouds to deploy if there is no reference. To solve this problem, we perform a set of experiments with different placement ratios. Figure 5A and Figure 5B show the comparison results in

TABLE 4 The overall objective values of each edge cloud placement approach with respect to the number of base stations

n	K-means	Top-K	Random	Our Approach
20	0.1264	0.0698	0.1429	**0.0922**
40	0.1408	0.0561	0.1210	**0.0744**
60	0.1004	0.0610	0.1091	**0.0589**
80	0.0786	0.0589	0.1115	**0.0582**
100	0.0683	0.0544	0.0882	**0.0462**
120	0.0598	0.0516	0.0826	**0.0412**
140	0.0657	0.0591	0.0789	**0.0475**
160	0.0684	0.0579	0.0864	**0.0456**
180	0.0700	0.0616	0.0742	**0.0492**
200	0.0887	0.0614	0.0696	**0.0487**



(A)

(B)

FIGURE 5 Effect of the placement ratio on the performance of our approach; A, Workload balance for the edge cloud placement result with different placement ratios; B, communication delay for the edge cloud placement result with different placement ratios [Colour figure can be viewed at wileyonlinelibrary.com]

terms of the workload balance and the communication delay, respectively, with the different placement ratios ranging from 0.05 to 0.15.

As shown in Figure 5, the workload balance and communication delay reduce with increasing the placement ratio because the high placement ratio means that there are more edge clouds that can share the tasks of base stations, and the base stations have more and better choices when they need to offload their tasks to edge clouds. This result can be a reference when a city or a company want to deploy edge clouds, and they can decide the placement ratio according to their specific requirements.

6 | CONCLUSION

Mobile edge computing has emerged as an important technology that can extend the computational resources of remote cloud and improve the quality of services serving mobile users. In this paper, we study the edge cloud placement problem in mobile edge computing and formulate it as a multiobjective optimization problem, and then propose an approximate approach through combining the K-means algorithm and mixed-integer quadratic programming algorithm. To evaluate the performance of different approaches in terms of the balance workload and the communication delay, we design three experiments based on a real Shanghai Telecom's base station data set. The experiment results show that the whole performance of our approach is better than other approaches to some extent.

Although our approach can obtain an appropriate edge cloud placement scheme, the mixed-integer quadratic programming problem formulated from the edge cloud placement problem is a complicated nonlinear programming problem, therefore, how to reduce the edge cloud placement problem to a simpler model is a key issue in our future work. Although the whole performance of our approach is acceptable, our approach is not much better than other approaches, therefore, how to design an approach to ensure both of the balance workload and the communication delay is another key issue in our future work. In addition, the workload of base stations is dynamically changing, the workload balance of the current edge cloud placement scheme will be broken, therefore, how to place the edge clouds dynamically is also a key issue in our future work.

ACKNOWLEDGEMENTS

This work was supported in part by the National Science Foundation of China (Grant 61602054 and 61472047) and supported by the BUPT Excellent Ph.D. Students Foundation CX2019211.

ORCID

Yan Guo https://orcid.org/0000-0001-5444-0253

REFERENCES

1. Hassan Q. Demystifying cloud computing. *J Def Softw Eng*. 2011;1:16-21.
2. Mell P, Grance T. The NIST definition of cloud computing. 2011. Special publication: 800-145.
3. Sharma S, Chang V, Tim US, Wong J, Gadia S. Cloud and IoT-based emerging services systems. *Clust Comput*. 2018:1-21.
4. Sharma S. Evolution of as-a-service era in cloud. 2015. arXiv preprint arXiv:1507.00939.
5. Xu Z, Liang W, Xu W, Jia M, Guo S. Efficient algorithms for capacitated cloudlet placements. *IEEE Trans Parallel Distributed Syst*. 2016;27(10):2866-2880.
6. Kemp R, Palmer N, Kielmann T, Bal H. Cuckoo: a computation offloading framework for smartphones. In: *Mobile Computing, Applications, and Services: Second International ICST Conference, MobiCASE 2010, Santa Clara, CA, USA, October 25-28, 2010, Revised Selected Papers*. Berlin, Germany: Springer-Verlag Berlin Heidelberg; 2010:59-79.
7. Zhang Y, Liu H, Jiao L, Fu X. To offload or not to offload: an efficient code partition algorithm for mobile cloud computing. Paper presented at: 2012 IEEE 1st International Conference on Cloud Networking (CLOUDNET); 2012; Paris, France.
8. Ahmed E, Akhunzada A, Whaiduzzaman M, Gani A, Ab Hamid SH, Buyya R. Network-centric performance analysis of runtime application migration in mobile cloud computing. *Simul Model Pract Theory*. 2015;50:42-56.
9. Liu J, Ahmed E, Shiraz M, Gani A, Buyya R, Qureshi A. Application partitioning algorithms in mobile cloud computing: taxonomy, review and future directions. *J Netw Comput Appl*. 2015;48:99-117.
10. Satyanarayanan M, Simoens P, Xiao Y, et al. Edge analytics in the Internet of Things. *IEEE Pervasive Comput*. 2015;14(2):24-31.
11. Sharma S. Expanded cloud plumes hiding big data ecosystem. *Futur Gener Comput Syst*. 2016;59:63-92.
12. Shi W, Cao J, Zhang Q, Li Y, Xu L. Edge computing: vision and challenges. *IEEE Internet Things J*. 2016;3(5):637-646.
13. Ahmed A, Ahmed E. A survey on mobile edge computing. Paper presented at: IEEE 10th International Conference on Intelligent Systems and Control; 2016; Coimbatore, India.
14. Qiao X, Ren P, Dustdar S, Chen J. A new era for web AR with mobile edge computing. *IEEE Internet Comput*. 2018;22(4):46-55.
15. Kosta S, Aucinas A, Hui P, Mortier R, Zhang X. ThinkAir: dynamic resource allocation and parallel execution in the cloud for mobile code offloading. Paper presented at: 2012 Proceedings IEEE INFOCOM; 2012; Orlando, FL.
16. Xia Q, Liang W, Xu W. Throughput maximization for online request admissions in mobile cloudlets. Paper presented at: 38th Annual IEEE Conference on Local Computer Networks; 2014; Sydney, Australia.
17. Xia Q, Liang W, Xu Z, Zhou B. Online algorithms for location-aware task offloading in two-tiered mobile cloud environments, In: Proceedings of the 2014 IEEE/ACM 7th International Conference on Utility and Cloud Computing; 2014; Dresden, Germany.

18. Jia M, Cao J, Liang W. Optimal cloudlet placement and user to cloudlet allocation in wireless metropolitan area networks. *IEEE Trans Cloud Comput*. 2017;5(4):725-737.
19. Xiang H, Xu X, Zheng H, et al. An adaptive cloudlet placement method for mobile applications over GPS big data. Paper presented at: 2016 IEEE Global Communications Conference (GLOBECOM); 2016; Washington, DC.
20. Hartigan JA, Wong MA. Algorithm AS 136: a k-means clustering algorithm. *J R Stat Soc Ser C Appl Stat*. 1979;28(1):100-108.
21. Bliek C, Bonami P, Lodi A. Solving mixed-integer quadratic programming problems with IBM-CPLEX: a progress report. In: Proceedings of the 26th RAMP Symposium; 2014; Tokyo, Japan.
22. Qiu L, Padmanabhan VN, Voelker GM. On the placement of web server replicas. In: Proceedings of IEEE 20th Annual Joint Conference of the IEEE Computer and Communications Society; 2001; Anchorage, AK.
23. Yin H, Zhang X, Zhan T, Zhang Y, Min G, Wu DO. NetClust: a framework for scalable and pareto-optimal media server placement. *IEEE Trans Multimed*. 2013;15(8):2114-2124.
24. Dohan D, Karp S, Matejek B. K-median algorithms: theory in practice. 2015.
25. Charikar M, Guha S, Tardos É, Shmoys DB. A constant-factor approximation algorithm for the k-median problem. *J Comput Syst Sci*. 2002;65(1):129-149.

How to cite this article: Guo Y, Wang S, Zhou A, Xu J, Yuan J, Hsu C-H. User allocation-aware edge cloud placement in mobile edge computing. *Softw: Pract Exper*. 2019;1–14. https://doi.org/10.1002/spe.2685