DOI: 10.3969/j. issn. 1673-5188. 2017. 02. 001
http://kns.cnki.net/kcms/detail/34.1294.TN.20170418.1708.002.html, published online April 18, 2017

Adaptive Service Provisioning for Mobile Edge Cloud

HUANG Huawei[1] and GUO Song[2]
(1. School of Computer and Engineering, The University of Aizu, Aizu-wakamatsu 965-0006, Japan;
2. Department of Computing, The Hong Kong Polytechnic University, Hong Kong SAR 852, China)

Abstract

A mobile edge cloud provides a platform to accommodate the offloaded traffic workload generated by mobile devices. It can significantly reduce the access delay for mobile application users. However, the high user mobility brings significant challenges to the service provisioning for mobile users, especially to delay-sensitive mobile applications. With the objective to maximize a profit, which positively associates with the overall admitted traffic served by the local edge cloud, and negatively associates with the access delay as well as virtual machine migration delay, we study a fundamental problem in this paper: how to update the service provisioning solution for a given group of mobile users. Such a profit-maximization problem is formulated as a nonlinear integer linear programming and linearized by absolute value manipulation techniques. Then, we propose a framework of heuristic algorithms to solve this Nondeterministic Polynomial (NP)-hard problem. The numerical simulation results demonstrate the efficiency of the devised algorithms. Some useful summaries are concluded via the analysis of evaluation results.

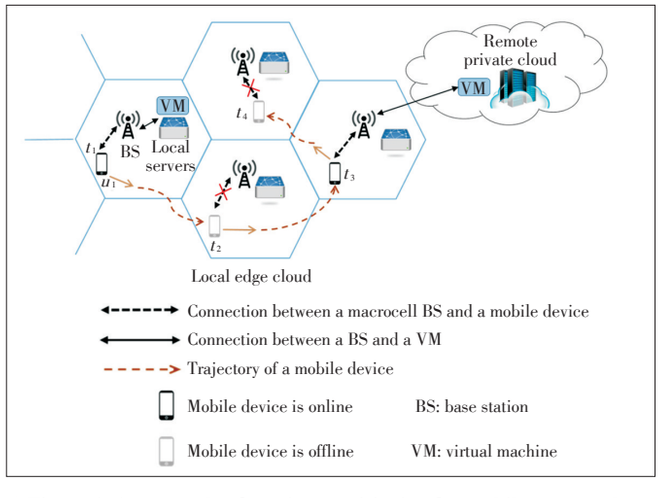
Keywords

edge cloud; mobile computing; service provisioning

1 Introduction

In recent years, the fast development of mobile cloud technologies [1]–[3] has incubated large varieties of mobile online applications to facilitate our daily life, e.g., mobile online games, big data applications [4], [5]. More importantly, most of them are normally highly delay-sensitive when executed in smartphones [6]. Nowadays the mobile devices are facing numbers of challenges such as suffering the shortage of computing capacity [4] and the battery poverty [7]. Therefore, the computational-intensive workload generated from the mobile devices is suggested to offload to a remote private cloud [8]–[11] for execution.

To alleviate these challenges, recent studies [9], [12]–[19] pay particular attentions to the cluster of distributed servers in the intermediate layered edge cloud network, called cloudlet. However, in a cloudlet based network such as a metropolitan area network [18], a certain group of mobile users normally join in (or become online) and leave (or become offline) the network randomly when they are using a particular mobile application, as shown in **Fig. 1**. Therefore, the disruption of connection between the mobile device and the server under a mobile application frequently occurs at different locations and different time frames. This brings a frequent churn to the service provisioning in cloudlet based network. Furthermore, in a real world, the access delay between each mobile device and the base station often dynamically changes in different locations even in a same cell (macrocell or smallcell).



▲Figure 1. An example of service provisioning for mobile users under a cloudlet based network. The workload generated from a mobile device can be offloaded to a VM, which resides in the local edge cloud or in a remote private cloud. Meanwhile, this figure also demonstrates the dynamic characteristics of an edge network, e.g., a mobile user alternates in online and offline status frequently.

This work was partially supported by JSPS KAKENHI under Grant Number JP16J07062.

Via an extensive survey in the next section over the existing related studies, we find out that the challenge to deal with the dynamic characteristics of the mobile cloudlet based networks has not been well addressed so far. Therefore, we are motivated to study a fundamental problem in this paper: how to update (partially or entirely) the service provisioning solution for a certain group of online mobile-application users in a cloudlet based network, supposed that the trajectory of each mobile device can be obtained according to the daily routine of each user. We try to answer the following two questions: 1) when to update the service provisioning solution for each mobile user, and 2) how to make a trade-off between the admitted traffic rate offloaded by the local edge cloud and the induced access delay and VM-migration delay while updating the current configuration.

Our study leads to the major contributions as follows.

- We study a service provisioning problem in the cloudlet based network, and try to find a near optimal update scheme for updating the service provisioning solution for each mobile user at each time-frame if the trajectory of each mobile user is provided.
- With the objective to maximize a weighted profit for network operators, we first formulate this problem to a nonlinear programming problem, which is then transformed to a solvable integer linear programming using the absolute value manipulation techniques.
- Because of the NP-hardness of the formulated problem, we have designed a series of heuristic Algorithms to solve the problem. Extensive numerical simulation results show that the devised algorithms can yield a near optimal solution. We also conclude some useful findings via the discussion of evaluation results.

The remaining paper is organized as follows. Section 2 reviews the related work. Section 3 presents the system model and gives the problem statement. The heuristic algorithms are elaborated in Section 4. Section 5 demonstrates the numerical evaluation. Finally, Section 6 concludes this paper.

2 Related Work

2.1 Cloudlet Based Edge Computing

Recently, edge computing has attracted wide-spread research efforts [9], [12]–[20] for the mobile computing. For instance, Xia et al. [9], [12] explored a location-based offloading problem, aiming to permit requests offloaded to a cloudlet network. Then authors proposed several efficient online algorithms that can dynamically handle the requests from users. A novel hierarchical edge cloud architecture constituted with multiple cloudlets has been proposed in [17] to efficiently serve the peak loads originating from mobile users. Then, to adaptively balance the tradeoff between response delay of mobile applications and energy efficiency, Tong et al. [20] proposed both offline and online algorithms to schedule the transmission in mobile cloud computing.

In wireless networks, the cloudlet placement problem also has been studied in [13], [14], [16], [18]. For example, in a wireless metropolitan area network (WMAN), in order to solve the problem of cloudlet placement, Jia et al. [14] proposed a placement scheme for a number of limited cloudlets. This approach is proved to greatly improve the mobile cloud performance. Similarly, Xu et al. [13], [16] also focused on the cloudlet placement problem, in which capacitated cloudlets need to find the best deployment locations within a given set of candidate locations. The objective is to minimize the average access delay between these activated cloudlets and mobile devices. To this end, some approximate algorithms have been devised with approximation ratios proved if all the cloudlet servers own the identical computing capability.

2.2 Task Offloading Using Edge Cloud

Wang et al. [21] studied a cost reduction problem in mobile edge clouds by deciding the assignment of mobile offloaded tasks. The authors formulated such a problem as a mixed integer program at first. Then, by introducing admission control, the problem is simplified and solved by the proposed efficient two-phase scheduling algorithm. To solve the decision making problem of computation offloading among multiple mobile users, Chen et al. [22] first formulated the problem as a multi-user computation offloading game, and proved that the game always assures a Nash equilibrium. Then, a game theoretic distributed algorithm is proposed to offload computation intensive tasks over the mobile edge could.

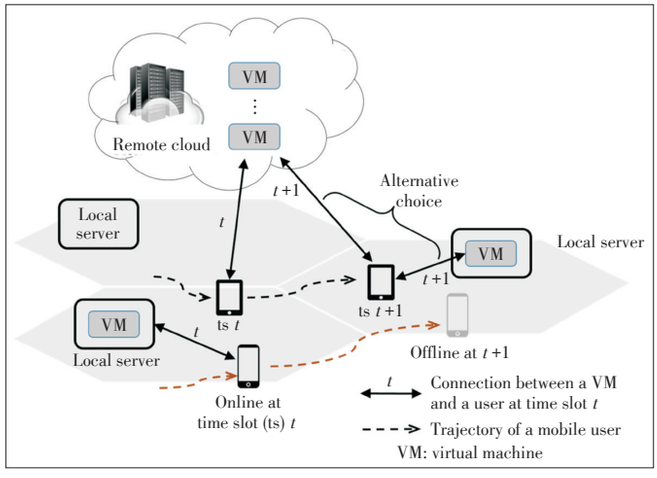
2.3 Comparison

Different from all efforts made by existing work mentioned above, this paper particularly studies the service provisioning update problem while considering the online and offline status of mobile users during their trajectories, as well as the highly dynamic characteristics of edge cloud networks. We find that this problem has not been well studied yet. To fill this gap, in this paper, we strive to design highly effective update schemes of service provisioning for edge cloud network operators.

3 Network Model and Problem Statement

3.1 System Model

The network that we focus on includes a cloudlet-based edge cloud and a remote private cloud. The former network consists of a set S of local edge servers. Without loss of generality, as shown in **Fig. 2**, we assume that a powerful edge server locates at each macrocell. Therefore, a mobile user connecting with a macrocell base station is equivalent to connecting with the corresponding local edge server. In such a cloudlet based network, a set U of mobile application users traverse at differ-



▲Figure 2. System model.

ent places in different time slots. Meanwhile, each of them becomes online and offline randomly while using the application on their mobile devices such as smartphone, tablet, etc. Suppose that the given trajectory of each mobile user is traced with the ID of its associated macrocell and online/offline status at each time slot. As a result, a timeslot labeled trajectory of a mobile user is constituted of a consecutive list of macrocell IDs. For example, a mobile user's trajectory looks like [$\langle t_1,cell\text{a}\rangle$, $\langle t_2,cell\text{b}\rangle$,...,$\langle t_n,0\rangle$,$\langle t_{n+1},cell_p\rangle$,$\langle t_{n+2},cell_q\rangle$,...], where $\langle t_n,0\rangle$ particularly represents that this user is offline at time-slot t_n. When the granularity of trace is quite fine, a same macrocell ID may continually appears many times if the mobile user keeps online in the macrocell area.

With the provided trajectories of all mobile users, the network operator needs to make a decision on where to deploy the required VM for each user at each time slot only when the user online. There are generally three categories of optimization models [15] when planning a service provisioning solution in the cloudlet based networks: 1) static planning, in which both the user mobility and VM mobility are not taken into account; 2) planning with non-real-time VM migrations, in which both user mobility and M-migrations are considered; 3) planning with delay-sensitive live VM migrations, in which the difference from the previous category is that the live VM-migrations are taken into account. In this paper, the mobile applications are assumed as highly delay sensitive ones. Therefore, we adopt the optimization scenario under the third category, i.e., considering the live VM-migrations. However, according to practice, we only concern the live VM migrations between the remote cloud and the local cloudlet network, and ignore the delay of intra-cloudlet VM migrations. **Table 1** shows the symbols and variables used in this paper.

3.2 Problem Statement and Formulation

We first define a binary variable x_u^t to denote the location to deploy the VM for an online mobile user $u \in U$ at the time-slot

▼Table 1. Symbols and variables

Notation	Description
U	the set of mobile users in network
S	the set of servers in the local cloudlet based network
T	the set of candidate time slots when to update the provisioning solution for each online mobile user
D_u	the demanded traffic rate of user $u \in U$
C_s	the traffic processing capacity of server $s \in S$
$F(u)$	a set of time-slots, in each of which user u becomes online from offline status, according to its given trajectory
R_u^t	the access delay from user u to the remote private cloud at time slot t
C_u^t	the access delay from user u to the local edge server at time slot t
Δ^t	total access delay of all mobile users at time-slot t
ζ	the normalized VM-migration delay between the private cloud and a local edge server
Γ_u^t	total VM-migration delay of all mobile users at time-slot t
x_u^t	binary variable indicating the location where to deploy a VM for an online user $u \in U$ at time-slot $t \in T$
z_u^t	binary variable denoting whether to migrate a VM between the remote private cloud and the local cloudlet network for an online user $u \in U$ at time-slot $t \in T$

$t \in T$ during its trajectory:

$$x_u^t = \begin{cases} 1, \text{ if a VM is deployed for an online user } u \\ \text{in a local edge server at the time slot } t; \\ 0, \text{ if a VM is deployed for an online user } u \\ \text{in the romote cloud at the time slot } t. \end{cases}$$

It can be seen that, different VM deployments for an online user indicate different access delays and VM-migration delays. To represent such two terms of delays, we then define an event named inter-cloud VM-migration, in which the VM serving an online user $u \in U$ is migrated between the remote private cloud and the local edge cloud. Then, another binary variable z_u^t is defined to denote whether the inter-cloud VM-migration event occurs at the time-slot $t \in T$:

$$z_u^t = \begin{cases} 1, \text{ if an inter}-\text{cloud VM}-\text{migration event occurs} \\ \text{for an online mobile user } u \text{ at the time slot } t; \\ 0, \text{ otherwise.} \end{cases}$$

By analyzing the given trajectory of each mobile user $u \in U$, we find that in some time slots, u becomes online from the offline status. Such a set of the online-activating time slots is denoted by $F(u)$. Naturally, we consider there is no inter-cloud VM-migration event occurring in each time slot $t \in F(u)$.

The objective is to maximize a weighted profit, which positively associates with the overall admitted traffic rate that is served by the local cloudlet network and negatively associates with the total access delay and the migration delay. In particular, letting ϕ_u^t denote the access delay of user $u \in U$ at the time-slot $t \in T$, we can calculate it as:

$$\phi_u^t = x_u^t \cdot C_u^t + \left(1 - x_u^t\right) \cdot R_u^t, \quad \forall t \in T, u \in U, \tag{1}$$

where C_u^t and R_u^t represents the access delay from user u to the local edge server and to the remote private cloud, respectively.

Then, we compute the access delay, which is denoted by Δ^t, at the time slot t in the following manner:

$$\Delta^t = \sum_{u \in U} \phi_u^t, \ \forall t \in T. \tag{2}$$

On the other hand, we let Γ^t indicate the total VM-migration delay of all mobile users at the time slot t, and it can be calculated as:

$$\Gamma^t = \sum_{t \in T} z_u^t \cdot \zeta, \ \forall t \in T, \tag{3}$$

where ζ denotes the normalized VM-migration delay between the private cloud and a local edge server.

Then, a profit-maximization is formulated as the following nonlinear programming:

$$\max P = \sum_{t \in T} \sum_{u \in U} D_u x_u^t - \sum_{t \in T} (w_1 \Delta^t + w_2 \Gamma^t), \tag{4a}$$

$$s.t. \sum_{u \in U} x_u^t \cdot D_u \cdot 1\big|_{(s=L(u,t))} \leqslant C_s, \forall t \in T, s \in S, \tag{4b}$$

$$z_u^t = \left| x_u^t - x_u^{t-1} \right|, \ \forall u \in U, \forall t, t-1 \in T \backslash F(u), \tag{4c}$$

$$z_u^t = 0, \ \forall t \in F(u), \forall u \in U, \tag{4d}$$

$$x_u^t, z_u^t \in \{0, 1\}, \ u \in U, \forall t \in T. \tag{4e}$$

In the objective function (4a), the first term $\sum_{t \in T} \sum_{u \in U} D_u x_u^t$ calculates the total admitted traffic rate that is served by the local cloudlet network, and w_1 and w_2 in the second term indicate the weight coefficients of the overall access delay and migration delay, respectively. Constraint (4b) expresses that the capacity of each server should not be expired. Note that $1|_{(\cdot)}$ is a binary indicator, which returns 1 if and only if the given condition is satisfied, and $L(u, t)$ is a location function that returns the cell where user u locates. Equation (4c) describes the relationship between variables z_u^t and x_u^t. As shown in this constraint, in any two successive time slots that user u is active in both, the case under $\left| x_u^t - x_u^{t-1} \right| = 0$ indicates that both x_u^t and x_u^{t-1} have the same binary value, meaning that there is no inter-cloud VM-migration event occurring at the time slot t for user u. On the other hand, once the inter-cloud VM-migration event occurs at the time slot t, we have the situation $\left| x_u^t - x_u^{t-1} \right| = 1$, which implies x_u^t and x_u^{t-1} must take different binary values, enforcing z_u^t = 1. Furthermore, (4d) imposes the aforementioned special rule for variable z_u^t when user u is in each timeslot of set $F(u)$.

It is worth noting that the objective function of (4) contains z_u^t, which is decided by the constraints (4c) and (4d). However, (4c) involves the absolute value functions, making (4) become nonlinear and not able to be solved using linear programming methods. Therefore, we particularly transform (4c) to two linear constraints through the following manipulation of the absolute value expression:

$$\left| x_u^t - x_u^{t-1} \right| = 0. \tag{5}$$

Finally, the nonlinear profit-maximization (4) can be reformulated as the following linear programming:

$$\begin{aligned} &\max P \\ &\text{s.t. } (4b), (5) \text{ and } (4d), \\ &x_u^t, z_u^t \in \{0, 1\}, u \in U, \forall t \in T. \end{aligned} \tag{6}$$

4 Heuristic Algorithms

Conventionally, the service provisioning problem under the constraints of resource capacity is known as NP-hard [23]–[26]. To solve the aforementioned profit-maximization problem, in this section, we present two types of fast heuristic algorithms and their variants, aiming to yield the service provisioning solutions in each time frame for each mobile user. The major contribution of this section is the proposal of the framework of heuristic algorithms, i.e., **Algorithm 1**, using which many variants of heuristic algorithms can be devised.

4.1 The Framework of Heuristic Algorithms

We first present a framework of the heuristic algorithms in Algorithm 1, based on which we are going to devise several heuristic algorithms in the third subsection.

In line 1, the empty solution x_u^t, z_u^t is generated at first. Then, it is initialized in line 3 according to a feasibility specification, which is going to be presented afterwards. Line 4 is to find the set of mobile users who locate at each macrocell where the local server $s \in S$ is deployed. Then, in line 5, algorithms sort all the mobile users decreasingly/increasingly by their demanded rates, and decide the priority to use the local edge server. After that, a priority set $\hat{U}_s^t$ is obtained in line 6 to denote the priority of users at each time slot $t \in T$. Next, the VM deployment for each server at each time slot can be decided as follows. Lines 9–15 show the operation under the case that a local server s is still capable to serve the traffic demanded by user u', while lines 16–22 demonstrate the opposite situation. Finally, algorithms deploy traffic demands in each local cloudlet-server, until the capacity of the server expires, and then deploy the remaining users to the remote cloud.

4.2 Structure and Feasibility Specification of a Solution

As mentioned, we have to specify a special feasibility specification to judge the feasibility of any element in a solution. Such a feasibility specification is elaborated with the explana-

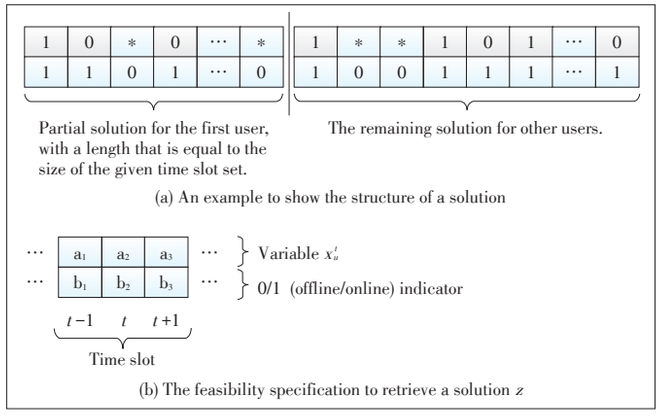
Algorithm 1: Framework of Heuristic Algorithms

Input : U, T, S and trajectory traces
Output: $x_u^t, z_u^t \in \{0,1\}, u \in U, \forall t \in T$

for $t \in T$, $u \in U$ **do**
  $x_u^t, z_u^t \leftarrow \varnothing$
Initialize x_u^t, z_u^t according to the given trajectory trace
Find the set of mobile users located at each macrocell where $\forall s \in S$ is deployed
Check the priority to use the local edge server of each user; sort them decreasingly/increasingly by their demanded rates
Obtain a sequential set $\hat{U}_s^t$ of mobile users by their priorities for each server $s \in S$ at each time slot $t \in T$
/*Decide the VM deployment for each mobile user at each time slot:*/
for $t \in T$, $s \in S$, $u' \in \hat{U}_s^t$ **do**
  if *s is feasible to serve the traffic demanded by* u' **then**
    /*Deploy a VM locally at s for u' */
    $x_{u'}^t \leftarrow 1$
    if t ⩾1 *and* 1 = $x_{u'}^{t-1}$ **then**
      $z_{u'}^t \leftarrow 0$
    else if t ⩾1 *and* 0 = $x_{u'}^{t-1}$ **then**
      $z_{u'}^t \leftarrow 1$
  else
    /*Deploy a VM remotely for u' */
    $x_{u'}^t \leftarrow 0$
    if t ⩾1 *and* 1 = $x_{u'}^{t-1}$ **then**
      $z_{u'}^t \leftarrow 1$
    else if t ⩾1 *and* 0 = $x_{u'}^{t-1}$ **then**
      $z_{u'}^t \leftarrow 0$

tion of solution structure in the following.

An example of the structure in a solution is shown in **Fig. 3a**. We can see that each solution particularly includes two row of binary codes. The intention of each row is illustrated in **Fig. 3b**. The first row indicates the variable $x_u^t(\forall u \in U, \forall t \in T)$, while the second row represents the offline/online status in each time slot. Only the bits in the first row labeled with an online indicator in the second row are valid bits, which are highlighted with shadow in Fig. 3a. The bit labeled with * denotes a "do-not-care" invalid bit, which will not be included in solution x. A valid binary bit in the first row implies that a VM is deployed in the local edge server for the current time slot, if it is equal to 1. Otherwise, it indicates that the VM serving a mobile user is deployed to the remote cloud. According to the given trajectory of each mobile user, the second row of a solution can be retrieved quickly. In the next step, each valid bit in the first row can be initialized randomly. After the initialization of solutions x and z, only the valid bits in the first row are need to be decided according to the chosen algorithm.



▲**Figure 3. The structure and the feasibility specification of a solution.**

We then explain how to retrieve the solution of inter-cloud VM-migration event, i.e., variables $z_u^t(\forall u \in U, \forall t \in T)$, when a solution x is provided. According to the definition of z_u^t and constraints (4d) and (4c), the rules are as follows: 1) to an invalid bit in the first row, we consider no inter-cloud VM-migration event occurs at this current corresponding time slot; 2) to any two adjacent valid bits in the first row, if the bit corresponding to the second time slot is labeled with 1 while the bit corresponding to the first time slot is labeled with 0, we still consider that there is no inter-cloud VM-migration event occurring at the second time slot; 3) if any two adjacent valid bits in the first row are labeled with different binary values, we consider the inter-cloud VM-migration event occurs at the second time slot. For the example shown in Fig. 3b, once $b_1 = 0$, we definitely have $z_u^{t-1} = 0$. On one hand, if $b_1 = 0$, both b_2 and b_3 are labeled with 1, the cases under $a_2 = 0$, $a_3 = 1$ and $a_2 = 1$, $a_3 = 0$ both yield $z_u^t = 0$ and $z_u^{t+1} = 1$. On the other hand, when b_1, b_2 and b_3 are all equal to 1, the same cases under $a_2 = 0$, $a_3 = 1$ and $a_2 = 1$, $a_3 = 0$ will both yield $z_u^{t+1} = 1$ for sure, and the value of z_u^t depends on a_1.

4.3 Heuristic Algorithms and Variants

Based on the algorithm framework, we now present two types of heuristic algorithms and their variants. The first one is called Online-First algorithm, the basic idea of which is to try to assign higher priority to the set of mobile users who are still in online status at the previous one time-slot. As a result, a mobile user who just becomes online at the current time slot has a lower priority than other local online mobile users. Finally, all the mobile users located at a local cell are classified into two groups by their priorities. We further get the final sequential set of users according to their demanded traffic rates. By sorting them decreasingly or increasingly by the demanded traffic rates, we finally receive the variants of such Online-First algorithm, which are named as Online-First-Decreasing and Online

-First-Increasing, respectively.

Another heuristic algorithm is called First-Fit, which is widely adopted to solve the bin-packing problem [24]. Similarly, according to the decreasingly/increasingly sorting manner towards the demanded traffic rate of each mobile user, the variants of First-Fit are labeled as First-Fit-Decreasing and First-Fit-Increasing, respectively.

5 Performance Evaluation

In this section, we conduct extensive numerical simulations to evaluate the presented 4 heuristic algorithms: First-Fit-Decreasing (FFD), First-Fit-Increasing (FFI), Online-First-Decreasing (OFD), and Online-First-Increasing (OFI).

The basic ideas of these 4 heuristic algorithms have been widely used by existing studies related to the resource allocation in cloud. Here we mainly compare the performance differences of the 4 heuristic algorithms designed under our proposed algorithm-framework. Furthermore, we are also interested in the performance gaps between such 4 algorithms and the Optimal one under different system settings. Finally, we would like to draw some useful conclusions over their performance by analyzing the simulation results, and try to suggest the service providers which heuristic algorithm is the best choice under a network configuration.

5.1 Simulation Settings

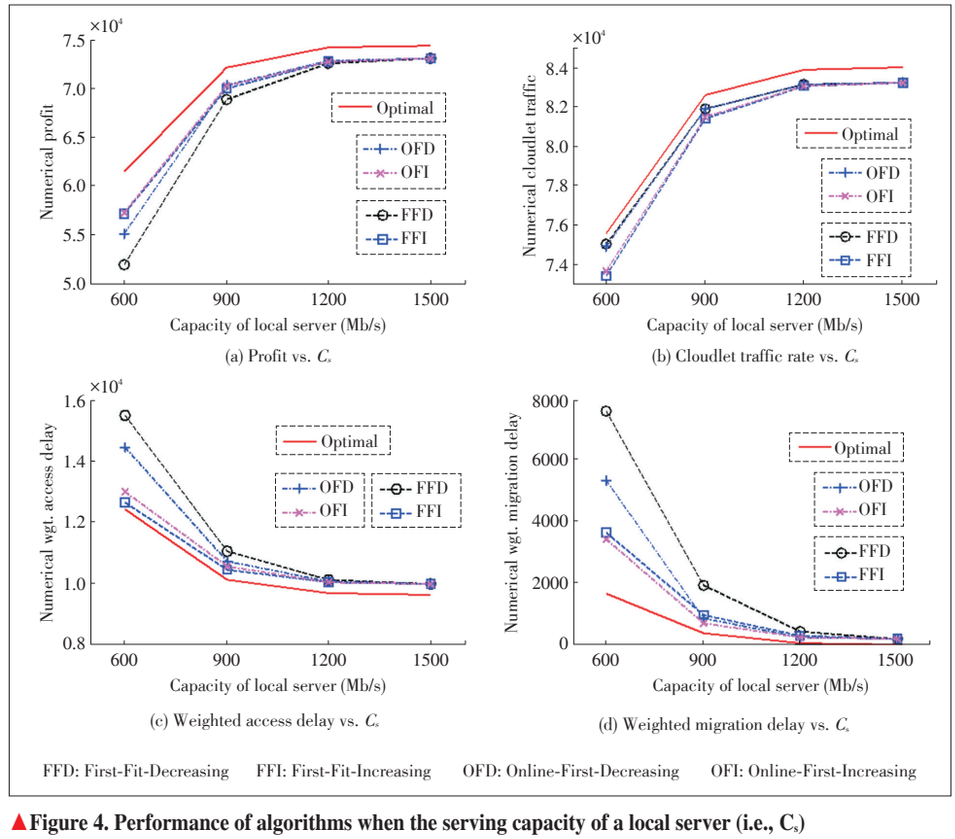
The network topology adopted in our simulations is a cloudlet based urban access network with 10 adjacent macrocells, each of which has an isolated local server that can only serve the mobile users located in the current macrocell. We randomly generate a traffic demand trace for each mobile user within [10, 100] Mb/s. The access delay to the remote cloud is fixed to 10 ms, while the local access delay of any mobile user to its local edge server is randomly generated within [1, 3] ms. Furthermore, the inter-cloud VM-migration delay is normalized to 10 ms.

We then generate a sequential trajectory for each mobile user within 20 time slots. At each time slot, we first decide the online status of any mobile user using a predefined probability, which is fixed to 0.8 in this paper. If a user is offline in a time slot, we mark its traversed cell ID to 0. Otherwise, we find a cell location following a twofold rule: 1) when a mobile user becomes online from an offline status, we randomly find a cell that it appears at; 2) when a mobile user keeps online from the previous one time slot, we find a cell for the current time slot within its located cell and the neighboring cells as well. On the other hand, as a benchmark to compare performance with our devised heuristic algorithms, we also solve (6) to retrieve the Optimal solution using Gurobi 6.0 [27], under each simulation setting. We compare heuristic algorithms and the optimal solution in terms of 4 metrics, i.e., total numerical profit, total traffic rate allocated to the local edge cloud, the weighted access delay and the weighted migration delay.

5.2 Effect of Traffic Processing Capacity of Edge Servers

In the first group of the simulations, we study the effect of server's traffic processing capacity by varying $C_s \in \{600, 900, 1200, 1500\}$ Mb/s, and fixing both w_1 and w_2 to 3. From **Figs.4a** and **4b**, we can observe that the profit and total numerical cloudlet traffic rate are increasing functions over the capacity of servers. When the capacity is insufficient, e.g., when C_s = 600 Mb/s, algorithms FFI and OFI perform better than the other two heuristics. This is because in the previous two algorithms, more mobile users who request traffic demands with small rates can be served in the local cloudlet servers resulting in smaller total access delay and migration delay.

Furthermore, in **Figs. 4c** and **4d**, we can see that the access



▲**Figure 4. Performance of algorithms when the serving capacity of a local server (i.e., C_s) varies in a range 600 Mb/s–1500 Mb/s.**

and migration delays decrease as the traffic processing capacity grows. As expected, the algorithms considering the demands with small traffic rates to be first served, i.e., FFI and OFI, have the lower delays than FFD and OFD algorithms.

Finally, once the processing capacity of local edge servers grows sufficiently, the performance of all algorithms becomes same. This can be explained by the reason that every algorithm yields a similar solution and performs close to the optimal solution, when the processing capacity of edge servers is not the bottleneck resource any more.

5.3 Effect of w_1

Using the same traces, we evaluate the effect of the weight of access delay, by varying $w_1 \in \{1, 2, 3, 4, 5\}$ and fixing $w_2 = 1$ and C_s = 500 Mb/s. **Fig. 5** illustrates the same four metrics of the previous group of simulations. Because the access delay contributes negatively to the objective function, we observe the decreasing profits in Fig. 5a and the increasing numerical weighted (shorten as wgt.) access delay in Fig 5c, while enlarging the weight of access delay from 1 to 5. FFI and OFI show the larger profits than that of the other two algorithms. The reason is same with the previous simulation.

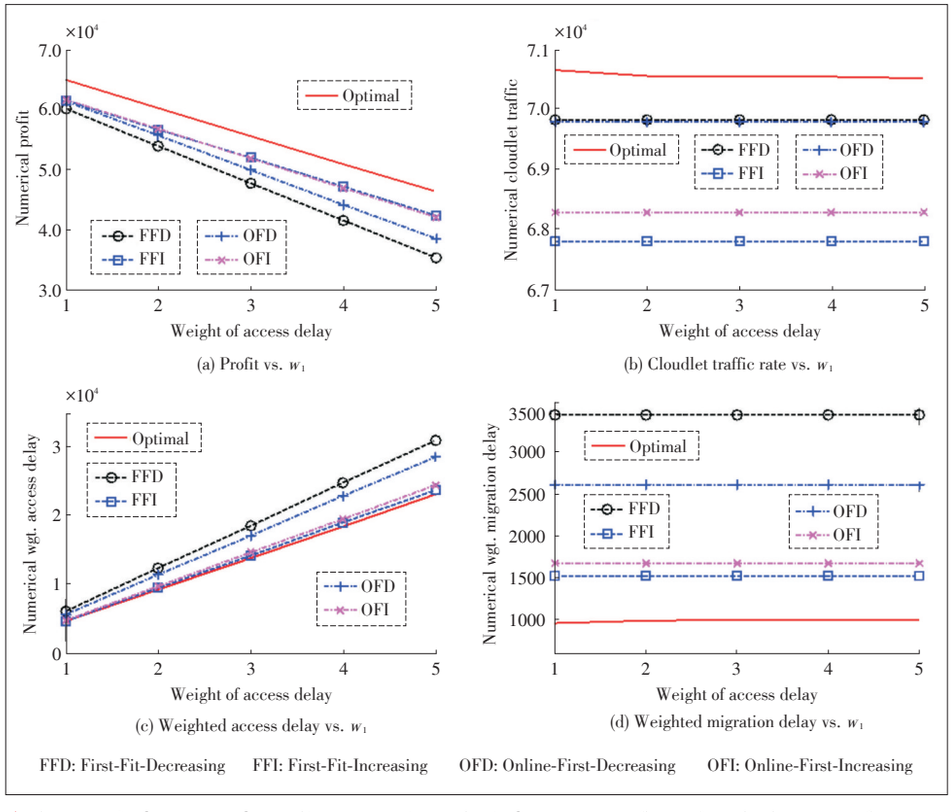
Interestingly, Figs. 5b and 5d demonstrate that improving the weight of access delay has no effect to the total cloudlet traffic and the weighted migrations delay. This is because changing w_1 will not significantly affect the task allocation to the local edge cloud or to the remote cloud. This is a useful finding to network operators.

5.4 Effect of w_2

By varying $w_2 \in \{1, 2, 3, 4, 5\}$ and setting w_1 to 1, we then study the effect of the weight of migration delay in this group of simulations. **Fig. 6** presents the 4 metrics of the four heuristic algorithms and the optimal solution as well. In Figs. 6a and 6b, we have similar observations on both the total profit and the total cloudlet traffic rate, compared with the previous group of simulations. This is because w_2 plays a similar role with w_1 to the system objective.

Although w_2 in all heuristic algorithms has no effect on the weighted access delay from Fig.6c, the increasing weight of migration delay makes the weighted migration delay higher. Thus, the total profit is reduced significantly. Especially under FFD, more traffic demands with small traffic rates have to experience the inter-cloud VM-migration, than that under other algorithms. This is because when the server capacity is limited, only a small number of requests can be provisioned in the local edge cloud. The VMs serving other users with tiny-rate demands have to be migrated to the remote cloud, thus incurring higher migration delay when performing the FFD and OFD algorithms.

In a summary, via all the simulation results, we can always observe that the FFI and OFI have a similar performance and outperform the other two heuristics in terms of total profit, the weighted access delay, and the weighted migration delay.



▲**Figure 5. Performance of algorithms when the weight of access delay (i.e., w_1) varies in a range 1–5.**

6 Conclusions

In this paper, we study the update problem of service provisioning in the cloudlet based mobile edge network. We try to find an adaptive update scheme to decide when to update the service provisioning solution for each mobile user at each time-frame, if the trajectory of each mobile user is known. With the objective of maximizing a weighted profit for network operators, we first formulate this problem as nonlinear programming problem. Then, it is transformed to solvable integer linear programming using the absolute value manipulation technique. Next, to solve this problem, we devise a series of heuristic algorithms. Extensive numerical simulation results demonstrate that

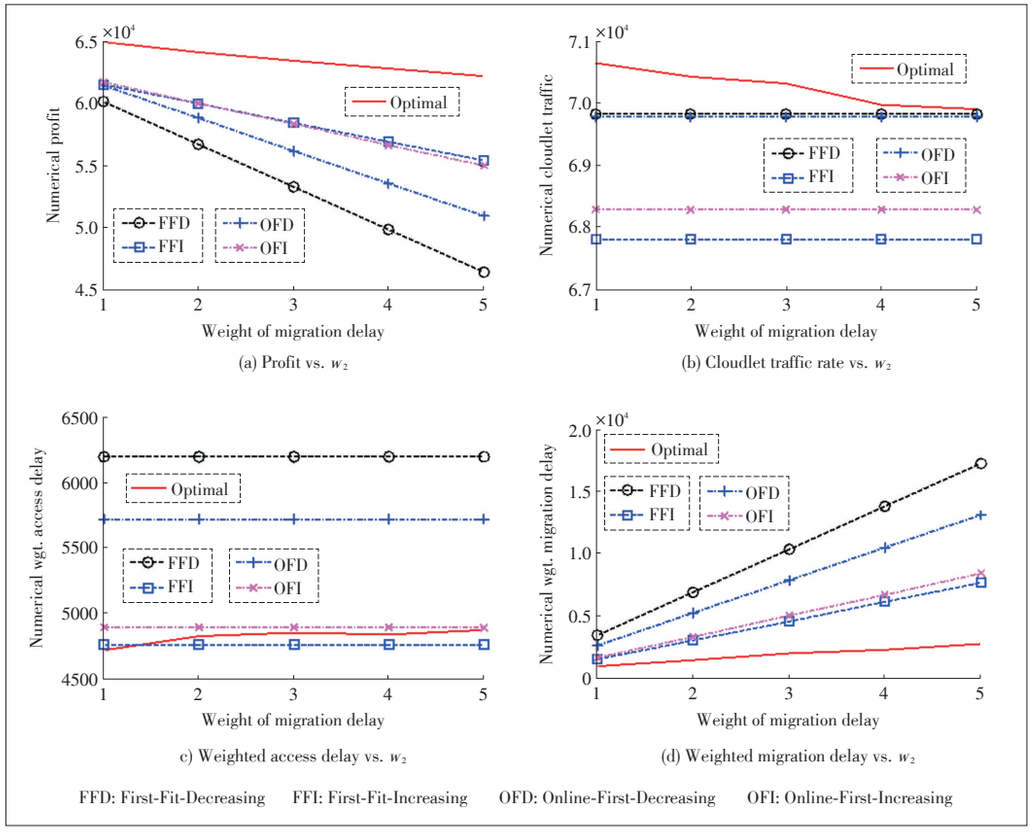


Figure 6. Performance of algorithms when the weight of migration delay (i.e., w_2) varies in a range 1–5.

the devised algorithms could yield near optimal solutions. Some useful findings have been also revealed through the evaluation of simulation results.

References

[1] T. Verbelen, P. Simoens, F. De Turck, and B. Dhoedt, "Cloudlets: bringing the cloud to the mobile user," in *Third ACM Workshop on Mobile Cloud Computing and Services*, Low Wood Bay, Lake District, UK, 2012, pp. 29–36. doi: 10.1145/2307849.2307858.

[2] D. Meilander, F. Glinka, S. Gorlatch, et al., "Using mobile cloud computing for real - time online applications," in *IEEE International Conference on Mobile Cloud Computing, Services, and Engineering*, Oxford, UK, 2014, pp. 48–56. doi: 10.1109/MobileCloud.2014.19.

[3] N. Fernando, W. L. Seng, and W. Rahayu, "Mobile cloud computing: A survey," *Future Generation Computer Systems*, vol. 29, no. 1, pp. 84–106, 2016.

[4] A. T. Lo'ai, W. Bakheder, and H. Song, "A mobile cloud computing model using the cloudlet scheme for big data applications," in *IEEE First International Conference on Connected Health: Applications, Systems and Engineering Technologies*, Washington, DC, USA, 2016, pp. 73–77. doi: 10.1109/CHASE.2016.40.

[5] A. T. Lo'ai, R. Mehmood, E. Benkhelifa, and H. Song, "Mobile cloud computing model and big data analysis for healthcare applications," *IEEE Access*, vol. 4, pp. 6171–6180, 2016. doi: 10.1109/ACCESS.2016.2613278.

[6] K. Ha, Z. Chen, W. Hu, et al., "Towards wearable cognitive assistance," in *International Conference on Mobile Systems*, Bretton Woods, USA, 2014, pp. 68–81. doi: 10.1145/2594368.2594383.

[7] K. Yang, S. Ou, and H. H. Chen, "On effective offloading services for resource-constrained mobile devices running heavier mobile internet applications," *IEEE Communications Magazine*, vol. 46, no. 1, pp. 56– 63, 2008. doi: 10.1109/MCOM.2008.4427231.

[8] D. Kovachev, T. Yu, and R. Klamma, "Adaptive computation offloading from mobile devices into the cloud," in *IEEE International Symposium on Parallel and Distributed Processing with Applications*, Madrid, Spain, 2012, pp. 784–791. doi: 10.1109/ISPA.2012.115.

[9] Q. Xia, W. Liang, and W. Xu, "Throughput maximization for online re-quest admissions in mobile cloudlets," in *IEEE 38th Conference on Local Computer Networks*, Sydney, Australia, 2013, pp. 589–596. doi: 10.1109/LCN.2013.6761295.

[10] W. Gao, Y. Li, H. Lu, T. Wang, and C. Liu, "On exploiting dynamic execution patterns for workload offloading in mobile cloud applications," in *IEEE 22nd International Conference on Network Protocols (ICNP)*, Raleigh, USA, 2014, pp. 1–12. doi: 10.1109/ICNP.2014.22.

[11] E. J. Haughn, "Mobile device management through an offloading network," U. S. Patent 8,626,143, Jan. 7, 2014.

[12] Q. Xia, W. Liang, Z. Xu, and B. Zhou, "Online algorithms for location-aware task offloading in two-tiered mobile cloud environments," in *IEEE/ACM 7th International Conference on Utility and Cloud Computing*, London, UK, 2014, pp. 109–116. doi: 10.1109/UCC.2014.19.

[13] Z. Xu, W. Liang, W. Xu, M. Jia, and S. Guo, "Capacitated cloudlet placements in wireless metropolitan area networks," in *IEEE 40th Conference on Local Computer Networks*, Clearwater Beach, USA, 2015, pp. 570–578. doi: 10.1109/LCN.2015.7366372.

[14] M. Jia, J. Cao, and W. Liang, "Optimal cloudlet placement and user to cloudlet allocation in wireless metropolitan area networks," *IEEE Transactions on Cloud Computing*, vol. 6, no. 25, pp. 1–14, 2015. doi: 10.1109/TCC.2015.2449834.

[15] A. Ceselli, M. Premoli, and S. Secci, "Cloudlet network design optimization," in *IFIP Networking Conference*, Toulouse, France, 2015, pp. 1–9. doi: 10.1109/IFIPNetworking.2015.7145315.

[16] Z. Xu, W. Liang, W. Xu, M. Jia, and S. Guo, "Efficient algorithms for capacitated cloudlet placements," *IEEE Transactions on Parallel and Distributed Systems*, vol. 27, no. 10, pp. 2866–2880, 2016.

[17] L. Tong, Y. Li, and W. Gao, "A hierarchical edge cloud architecture for mobile computing," in *IEEE International Conference on Computer Communications*, San Francisco, USA, 2016, pp. 1–9. doi: 10.1109/INFOCOM.2016.7524340.

[18] M. Jia, W. Liang, Z. Xu, and M. Huang, "Cloudlet load balancing in wireless metropolitan area networks," in *IEEE International Conference on Computer Communications*, San Francisco, USA, 2016, pp. 1– 9. doi: 10.1109/INFOCOM.2016.7524411.

[19] X. Sun and N. Ansari. (2016). *Green cloudlet network: A distributed green mo-*

bile cloud network [Online]. Available: https://arxiv.org/abs/1605.07512
[20] L. Tong and W. Gao, "Application - aware traffic scheduling for workload offloading in mobile clouds," in *IEEE International Conference on Computer Communications*, San Francisco, USA, 2016, pp. 1–9. doi: 10.1109/INFOCOM.2016.7524520.
[21] L. Wang, L. Jiao, D. Kliazovich, and P. Bouvry, "Reconciling task assignment and scheduling in mobile edge clouds," in *IEEE 24th International Conference on Network Protocols*, Singapore, 2016, pp. 1–6.
[22] 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, 2016. doi: 10.1109/TNET.2015.2487344.
[23] N. M. K. Chowdhury, M. R. Rahman, and R. Boutaba, "Virtual network embedding with coordinated node and link mapping," in *IEEE International Conference on Computer Communications*, San Francisco, 2009, pp. 783–791. doi: 10.1109/INFOCOM.2009.5061987.
[24] H. Huang, D. Zeng, S. Guo, and H. Yao, "Joint optimization of task mapping and routing for service provisioning in distributed datacenters," in *IEEE International Conference on Communications*, Sydney, Australia, Jun. 2014, pp. 4196–4201.
[25] H. Huang, P. Li, S. Guo, and B. Ye, "The joint optimization of rules allocation and traffic engineering in software defined network," in *IEEE 22nd International Symposium of Quality of Service*, Hong Kong, China, 2014, pp. 141–146. doi: 10.1109/IWQoS.2014.6914313.
[26] H. Huang, S. Guo, P. Li, B. Ye, and I. Stojmenovic, "Joint optimization of rule placement and traffic engineering for QoS provisioning in software defined network," *IEEE Transactions on Computers*, vol. 64, no. 12, pp. 3488–3499, 2015. doi: 10.1109/TC.2015.2401031.
[27] Gurobi Optimization. (2016). *Gurobi optimizer reference manual* [Online]. Available: http://www. gurobi. com

Manuscript received: 2017-01-15

Biographies

HUANG Huawei (davyhwang.cug@gmail.com) received his Ph.D. in computer science from the University of Aizu, Japan. His research interests mainly include network optimization and algorithm design/analysis for wired/wireless networks. He is a member of IEEE and a JSPS Research Fellow.

GUO Song (song.guo@polyu.edu.hk) received his Ph.D. in computer science from University of Ottawa, Canada. He is currently a full professor at Department of Computing, The Hong Kong Polytechnic University (PolyU), China. Prior to joining PolyU, he was a full professor with the University of Aizu, Japan. His research interests are mainly in the areas of cloud and green computing, big data, wireless networks, and cyber-physical systems. He has published over 300 conference and journal papers in these areas and received multiple best paper awards from IEEE/ACM conferences. His research has been sponsored by JSPS, JST, MIC, NSF, NSFC, and industrial companies. Dr. GUO has served as an editor of several journals, including *IEEE TPDS*, *IEEE TETC*, *IEEE TGCN*, *IEEE Communications Magazine*, and *Wireless Networks*. He has been actively participating in international conferences serving as general chairs and TPC chairs. He is a senior member of IEEE, a senior member of ACM, and an IEEE Communications Society Distinguished Lecturer.

Call for Papers

ZTE Communications Special Issue on

Motion and Emotion Sensing Driven by Big Data

Motion and emotions are two critical features of human presence and activities. Recent developments in the field of indoor motion and emotion sensing have revealed their potentials in enhancing our living experiences through applications like public safety and smart health. However, existing solutions still face several critical downsides such as the availability (specialized hardware), reliability (illumination and line - of - sight constraints) and privacy issues (being watched). To this end, this special issue seeks original articles describing development, relevant trends, challenges, and current practices in the field of applying Artificial Intelligence to address various issues of motion and emotion sensing brought by the "Big data". Position papers, technology overviews, and case studies are also welcome.

Appropriate topics include, but not limited to

- Motion and Emotion Model/Theory driven by Big Data
- Motion and Emotion Sensing Algorithms driven by Big Data
- Multi-modality Data Processing for Motion and Emotion Sensing
- Multi - modality Data Mining for Motion and Emotion Sensing
- Novel Motion and Emotion Applications/Systems Supported by Big Data
- Evaluation Metrics and Empirical Studies of Motion and Emotion Sensing Systems
- Quality - enhanced and adaptive sensing models driven by Big Data
- Inherent Relationship Modeling between Motion and Emotions driven by Big Data

First submission due: August 15, 2017
Peer review: September 30, 2017
Final submission: October 15, 2017
Publication date: December 25, 2017

Submission Guideline:

Submission should be made electronically by email in WORD format.

Guest Editors:

Prof.Fuji Ren (ren@is.tokushima - u.ac.jp), University. of Tokushima, Japan

Prof.Yixin Zhong (zyx@bupt.edu.cn), Beijing University Of Posts and Telecommunications, China

Prof.Yu Gu (yugu.bruce@ieee.org), Hefei University of Technology, China