Mobility-Aware Dynamic Service Placement in D2D-Assisted MEC Environments

Jing Li[†], Weifa Liang[†], Mengyu Chen[†], and Zichuan Xu[‡]
† The Australian National University, Canberra, ACT 2601, Australia
‡ Dalian University of Technology, Dalian, 116020, P. R. China

2021 IEEE Wireless Communications and Networking Conference (WCNC) | 978-1-7281-9505-6/20/$31.00 ©2021 IEEE | DOI: 10.1109/WCNC49053.2021.9417358

***Abstract*—Mobile Edge Computing (MEC) has emerged as a promising networking paradigm that provides delay-sensitive service for mobile users at the edge of core networks, where mobile users can offload their computing-intensive tasks to MEC networks for processing on no time. Furthermore, with the advance of communication and fabrication technologies, mobile devices now have adequate computing and storage processing capabilities. The device-to-device (D2D) offloading as a new offloading technique enables mobile users to offload their tasks to other mobile devices (referred as helper mobile devices) for processing, thereby alleviating the processing burden on servers in MEC. However, fully utilizing the D2D technique in an MEC network for task offloading service is challenging. Particularly, the mobility of both mobile users and their helper mobile devices makes efficient offloading service placement become difficult. In this paper, we study a novel Mobility-aware Dynamic Offloading Service Placement (MDOSP) problem in a D2D-assisted MEC environment with the aim to minimize the total cost of offloading task services that consists of the computing cost, communication delay cost and migration cost, without the knowledge of future mobility information of mobile users and helper mobile devices. We first formulate an Integer Nonlinear Programming (INP) for the offline setting of the problem. We then prove the NP-hardness and develop an online algorithm with a provable competitive ratio for the problem. We finally evaluate the performance of the proposed algorithms through experimental simulations. Experimental results demonstrate that the proposed algorithms are promising, compared with existing baseline algorithms.**

I. Introduction

Fueled by the burst demands of mobile users in both resource-sensitivity and delay-sensitivity dimensions, Mobile Edge Computing (MEC) has emerged as a promising technology to provision the cloud-computing service hosted on cloudlets (edge servers) at the edge of core network in the proximity of mobile users [4]. MEC is gaining momentum to provide a plethora of network services with much lower response time and augmented processing abilities [6].

Traditional task offloading in MEC by offloading mobile user tasks to edge servers has been extensively studied in the past several years, and such offloading has its limitations. For example, edge servers in an MEC network may not be able to meet computing resource demands of a large number of mobile users, especially in a metropolitan area with a dense distribution of mobile users [10]. Meanwhile, due to the spatial distribution of mobile users, a mobile user could be far from his nearest cloudlet, which leads to unacceptable offloading delay [3]. To mitigate this limitation, a new task-offloading scheme - the device-to-device (D2D) assisted MEC, has drawn much attention in recent years. With the advent of mobile devices, their computing capacities have evolved to be non-negligible and comparable to those of desktops. And it is very likely in many application scenarios where some mobile devices are idle or have much available resource for other users [1]. To fully utilize these mobile device resources to reduce the offloading response delays and alleviate the workload of cloudlets in an MEC network [12], it is desirable to make use of these mobile devices. These idle mobile devices in the network are enabled as *helper mobile devices* to share their computing resources with active mobile users for their offloaded task processing [1].

To facilitate the efficient task offloading in a D2D-assisted MEC network, there are two main challenges: one is how to choose between the MEC and D2D platforms for each task offloading, and how to determine which cloudlet or mobile device in the chosen platform to perform the offloaded task; another is how to cope with the dynamic and agnostic mobility of both mobile users and helper mobile devices [9]. The mobility of mobile users may lead to substantial delays and degrade the offloading service quality. A widely-adopted approach is to migrate the offloaded services dynamically to adapt to the movement of mobile devices. However, the mobility of helper mobile devices should also be taken into consideration. It is difficult to make global optimal migration decisions without the knowledge of future mobility information. Unlike previous studies that only considered the mobility of mobile users [2], [6], [9], in this paper we consider the mobility of both mobile users with task offloading requests and helper mobile devices by providing efficient solutions to task offloading.

The novelty of this paper is that we study the *Mobility-aware Dynamic Offloading Service Placement (MDOSP) problem* in a D2D-assisted MEC environment, where both cloudlets and helper mobile devices serve offloaded tasks for mobile users to reduce the service delay significantly. To the best of our knowledge, we are the first to consider the MDOSP problem with the aim to minimize the total cost of offloading task services consisting of the computing cost, communication delay cost and migration cost without the knowledge of future mobility information of both mobile users and helper mobile devices. We also propose an online algorithm with a provable competitive ratio for the problem.

The main contributions of this paper are as follows. We first formulate a novel MDOSP problem in a D2D-assisted

978-1-7281-9505-6/21/$31.00 ©2021 IEEE

MEC environment, with the aim to minimize the total cost of offloading task services that consists of the computing cost, communication delay cost, and migration cost of offloaded tasks without the knowledge of future mobility information of mobile users or helper mobile devices for a finite time horizon. We then show the NP-hardness of the problem and formulate an Integer Nonlinear Programming (INP) solution to the offline version of the MDOSP problem, where all the mobility information for the time horizon is given. We also devise an online algorithm with a provable competitive ratio for it. We finally evaluate the performance of the proposed algorithm through experimental simulations. Experimental results demonstrate that the proposed algorithm is promising.

The remainder of the paper is organized as follows. Section II reviews the related work. Section III introduces the system model and defines the problem precisely. The NP-hardness of the defined problem is also shown in this section. Section IV formulates an INP solution for the problem under the offline setting, and devises an online algorithm for the problem. Section V evaluates the performance of the proposed algorithms through experimental simulations, and Section VI concludes the paper.

II. Related Work

Several recent studies focused on task offloading in the D2D-assisted MEC environment [1], [3], [12]. For example, Cao *et al.* [1] jointly considered partial offloading for divisible tasks and binary offloading for indivisible tasks to helper mobile devices and MEC servers. Kai *et al.* [3] formulated an energy minimization problem in a D2D-assisted MEC environment, and proposed a near-optimal algorithm under both delay and computing resource capacity constraints. Zhang *et al.* [12] developed an auction scheme to efficiently allocate computing resource in a D2D-assisted MEC network with respect to delay tolerance and resource capacities.

Task offloading in MEC environments with user mobility has also been investigated recently. For example, Gao *et al.* [2] jointly considered the efficient access network selection and dynamic service placement to achieve a trade-off among the access, switching and communication delays. Ma *et al.* [6] studied the problem of providing reliable and seamless network services for mobile users provided that the mobility patterns of mobile users are given. Wang *et al.* [9] addressed the dynamics of the mobility of mobile users by formulating a general cost model, and proposed an efficient online algorithm by adopting the “regularization” technique [9].

Unlike the aforementioned studies, in this paper we consider a mobility-aware dynamic service provisioning in a D2D-assisted MEC environment with the aim to minimize the total cost of offloading task services that consists of the computing cost, communication delay cost and migration cost, without the knowledge of future mobility information of mobile users and helper mobile devices.

III. Preliminaries

A. System model

We consider a Mobile Edge Computing (MEC) network, which is modeled by an undirected graph $G = (V, E)$, where V is the set of Access Points (APs) and E is the set of links connecting APs. Each cloudlet where the network services are implemented is co-located with an AP via optical fibers and the communication delay between them is negligible. Each edge $e \in E$ has an associated communication delay d_e. Each mobile user accesses the MEC and helper mobile device resources through wireless communication with its nearest AP. For the sake of convenience, in this paper we focus on computing resource consumption. We consider a D2D-assisted MEC network environment, where idle mobile devices of mobile users in the network can be enabled as helper mobile devices to allow their computing resource to be shared by other mobile users through offloading their tasks to the helper mobile devices for processing, thereby reducing the workload of cloudlets and improving the performance of the MEC network.

B. User request scheduling

We are given a time horizon $\mathcal{T}$ that is slotted into equal time slots, i.e., $\mathcal{T} = \{1, 2, \ldots, |\mathcal{T}|\}$. We assume that the locations of both mobile users and helper mobile devices can be changed in the beginning of each time slot but unchanged within a time slot.

Denote by U the set of mobile users offloading their tasks over a given time horizon $\mathcal{T}$. In each time slot $t \in \mathcal{T}$, each mobile user $u \in U$ is located at one location (an AP) in the MEC network and connected to the AP by offloading a task demanding computing resource of ρ_u. Denote by $\mathbb{K}$ the set of *offloaded targets* for each offloading task, i.e., the set of helper mobile devices and cloudlets in the system, and denote by c_k the computing capacity of an offloaded target $k \in \mathbb{K}$. Initially, at the beginning of the time horizon ($t = 1$), a mobile user can send his location and an offloading task - a service request to the MEC through its nearest AP. Meanwhile, each helper mobile device can send its location information to the MEC, too. The network central scheduler then performs task allocation by assigning each offloading task to an offloaded target for processing, based on the service requests received. Then, at the beginning of the next time slot, each mobile user and helper mobile device will send their new location information to the MEC for the next round task assignment. The network central scheduler will update the system information and determine whether to migrate the service of a mobile user to a new offloaded target.

Denote by $x_{u,k,t}$ a binary decision variable indicating whether mobile user u offloads his task to an offloaded target k at time slot $t \in \mathcal{T}$. Then the total computing resource consumption on an offloaded target k is no more than its computing capacity at any time slot t, i.e.,

$$\sum_{u \in U} \rho_u \cdot x_{u,k,t} \leq c_k, \qquad \forall k \in \mathbb{K},\ \forall t \in \mathcal{T}. \tag{1}$$

Due to the mobility of mobile devices, let $l_{u,t}$ and $l_{k,t}$ be the locations of a mobile user u and an offloaded target k at time slot t, respectively. Note that the locations of a mobile user and a helper mobile device may have different locations at different time slots while the location of each cloudlet in the MEC network is always stationary.

C. Cost modeling

We express the total cost of offloading task services during a given time horizon as the sum of the costs of all offloading tasks with each cost consisting of the computing cost, communication delay cost and migration cost for that time horizon. The precise definitions of these costs are given as follows.

The computing Cost. Denote by a_k the cost of one unit computing resource on offloaded target k. Then, the computing cost of offloading task services at time slot t, denoted by E_c^t, is defined as follows.

$$E_c^t = \sum_{u \in U} \sum_{k \in \mathbb{K}} a_k \cdot \rho_u \cdot x_{u,k,t}, \tag{2}$$

The communication delay cost. Denote by $d(v, v')$ the communication delay along a shortest routing path between nodes u and v, and the weight of each link $e \in E$ is its communication delay d_e. Denote by w the coefficient of the communication delay cost. Then, the communication delay cost of all offloading tasks at time slot t is defined as follows.

$$E_d^t = \sum_{u \in U} \sum_{k \in \mathbb{K}} w \cdot d(l_{u,t}, l_{k,t}) \cdot x_{u,k,t}. \tag{3}$$

Without loss of generality, we assume the result downloading time of an offloaded task is negligible, because the volume size of a result usually is much smaller than the size of the task uploading [1].

The migration Cost. We need to dynamically conduct service migrations to alleviate the communication delays, considering the mobility of mobile users and helper mobile devices. Denote by $b_u(v, v')$ the migration cost by migrating the task of mobile user u from node v to node v', which is determined by the task size and the migration delay along the shortest routing path in G from node v to node v'. The migration cost of all offloading tasks between two consecutive time slots $(t-1)$ and t thus is defined as follows.

$$E_m^t = \sum_{u \in U} \sum_{k \in \mathbb{K}} \sum_{k' \in \mathbb{K} \setminus k} b(l_{k,t}, l_{k',t}) \cdot x_{u,k,t-1} \cdot x_{u,k',t}. \tag{4}$$

The total cost E^t of offloading task services at time slot t thus is defined as follows.

$$E^t = E_c^t + E_d^t + E_m^t. \tag{5}$$

D. Problem definition

Given an MEC network $G(V, E)$, a time horizon $\mathcal{T}$, a set U of mobile users in which each user $u \in U$ has a service request, and a set of helper mobile devices allowing to share their computing resource for services, the *Mobility-aware Dynamic Offloading Service Placement (MDOSP) problem* in a D2D-assisted MEC environment is to offload user services to cloudlets and helper mobile devices with the aim to minimize the total cost of offloading task services for the time horizon while incorporating the mobility of both mobile users and helper mobile devices, subject to the computing capacity on each cloudlet and each helper mobile device.

Theorem 1: The MDOSP problem is NP-hard.

Proof: The NP-hardness of the MDOSP problem is shown by reducing from a NP-hard problem - the generalized assignment problem (GAP) [8] as follows.

Given n bins and m items, bin i has a capacity W_i, $1 \leq i \leq n$, item j has a cost c_{ij} with weight w_j if assigned to bin i, $1 \leq j \leq m$. The GAP aims to minimize the total cost through assigning items to bins, subject to the bin capacities.

We consider a special case of the MDOSP problem where the locations of all mobile users and helper mobile devices do not change. And we do not need to consider service migrations. Then, if we place the service of mobile user j at offloaded target i, it consumes computing resource of w_j and incurs a cost c_{ij} consisting of the corresponding computation cost and communication delay cost. To this end, the MDOSP problem aims to minimize the total cost of offloading task services, subject to the capacity W_i of each offloaded target i. And the special case of the MDOSP problem is equivalent to the GAP. Hence, the MDOSP problem is NP-hard, due to the NP-hardness of the GAP [8]. ■

E. The competitive ratio of an online algorithm

Denote by $OPT(\mathcal{P})$ the optimal solution of a minimization problem $\mathcal{P}$ in its offline version. An online algorithm for problem $\mathcal{P}$ achieves a competitive ratio α if the solution delivered by the algorithm is no greater than $\alpha \cdot OPT(\mathcal{P})$ with $\alpha \geq 1$.

IV. Algorithm

In this section, we first formulate an Integer Nonlinear Programming (INP) solution for the offline version of the MDOSP problem. We then give an Integer Linear Programming (ILP) for the sub-problem of the problem at each time slot t. We finally devise an online algorithm with a provable competitive ratio for the problem.

A. Integer Nonlinear Programming

Assuming that the movement information of all mobile users and helper mobile devices during the time horizon is given, the offline version of the MDOSP problem is formulated as follows.

$$\text{minimize} \quad \sum_{t \in \mathcal{T}} E^t, \tag{6}$$

subject to

$$(1), (2), (3), (4), (5),$$

$$\sum_{k \in \mathbb{K}} x_{u,k,t} = 1, \qquad \forall u \in U,\ \forall t \in \mathcal{T}, \tag{7}$$

$$x_{u,k,t} \in \{0, 1\}, \qquad \forall u \in U,\ \forall k \in \mathbb{K},\ \forall t \in \mathcal{T}, \tag{8}$$

where Constraint (1) is the computing capacity constraint on each offloaded target, from Constraint (2) to Constraint (5) the total cost is defined, and Constraint (7) ensures that each mobile user u only offloads his task to one offloaded target (a helper mobile device or a cloudlet) at any time slot.

As Eq. (4) is a nonlinear constraint, it is difficult to devise an efficient algorithm for the MDOSP problem without the future mobility information of mobile users or mobile devices.

To tackle this INP problem, we instead first decompose the MDOSP problem into sub-problems at each time slot $t \in \mathcal{T}$. We then devise an online algorithm for it derived from the solutions of the sub-problems.

With the optimization objective (6), an Integer Linear Programming (ILP) solution for the sub-problem of the MDOSP problem at time slot t is proposed as follows.

$$\text{minimize} \quad E^t, \tag{9}$$

subject to

$$(1), (2), (3), (4), (5), (7), (8).$$

Note that it is an ILP at time slot t because $x_{u,k,t-1}$ has been determined already when we consider time slot t, by Eq. (4).

B. Online Algorithm

Although we devise an efficient algorithm to solve the sub-problem (9), we cannot apply the developed algorithm to the sub-problem at each time slot directly without the future mobility information of mobile users and helper mobile devices, because the migration cost may be prohibitively large, which will become the dominant of the total cost. Inspired by the work in [11], we here introduce the migration control policy for service migration to bound the migration cost. That is, we define the sum of the computing cost and the communication delay cost of an offloading task service as its *static cost*, while its service migration cost is defined as the *dynamic cost*. Denote by E_S^t and E_D^t the static and dynamic costs of all offloading tasks at time slot t, i.e., $E_S^t = E_c^t + E_d^t$ and $E_D^t = E_m^t$, respectively.

The proposed online algorithm, `Algorithm` 1 proceeds iteratively, one iteration for each time slot.

For a given time slot t, based on the updated location information of mobile users and helper mobile devices, we first calculate the cost of offloading each mobile user task to each offloaded target with sufficient computing resource, consisting of the computing cost, communication delay cost and migration cost. Among all pairs of mobile users and their offloaded targets, a pair (mobile user u' and offloaded target k') with the lowest total cost is chosen. The task of user u' then is offloaded to the associated offloaded target k', and the residual capacity of the offloaded target k' is updated. For the rest mobile users, the cost of offloading each mobile user task to each offloaded target is then recalculated. This procedure continues until the offloading tasks of all mobile users are done. Note that at the very first time slot, we do not need to consider the dynamic cost. We then figure out the feasible placement decision at each time slot. However, we cannot apply the obtained result directly, because the migration cost may become dominant without future mobility information. We thus need to design a migration control policy to avoid a large migration cost as follows.

Let $\hat{t}$ be the last migration time slot so far. Initially, $\hat{t}$ is set as 1. Denote by $\beta > 0$ a control parameter, which is used to control the service migration cost. If the migration cost in the current time slot t is no greater than $\frac{1}{\beta}$ times the sum of static costs from time slot $\hat{t}$ to time slot $(t-1)$, i.e., $E_D^t \leq \frac{1}{\beta}\sum_{t'=\hat{t}}^{t-1} E_S^{t'}$, we adopt the new placement decision at time slot t and migrate the placed services; otherwise, we do nothing. A smaller value of β indicates less tolerance on service delays are allowed and thus more frequent service migrations are needed. Otherwise (a larger β) implies that less frequent service migrations are performed and more service delays are tolerant. The detailed algorithm for the MDOSP problem is given in `Algorithm` 1.

Algorithm 1 Algorithm for the MDOSP problem

```
Input: An MEC network $G = (V, E)$, a set of mobile users $U$ and a set
    of helper mobile devices moving around in the network over the time
    horizon $\mathcal{T}$, and there is no future mobility information.
Output: An assignment of offloading tasks to cloudlets and helper mobile
    devices to minimize the total cost of task offloading for all mobile users
    within a given time horizon.
 1: Compute the routing paths between each pair of APs in $G$ with the least
    communication delay;
 2: $t \leftarrow 1$, $\hat{t} \leftarrow 1$;
 3: while $t \leq |\mathcal{T}|$ do
 4:     Update the locations of mobile users and helper mobile devices;
 5:     $U' \leftarrow U$; /* the set of mobile users whose services have not been
        offloaded*/
 6:     while $U' \neq \emptyset$ do
 7:         for each $u \in U'$ do
 8:             for each $k \in \mathbb{K}$ with sufficient computing resource do
 9:                 Calculate the cost of offloading the service of mobile user $u$
                    to offloaded target $k$ by Eq. (5);
10:             end for;
11:         end for;
12:         Pick the offloading pair (mobile user $u'$ and offloaded target $k'$)
            with the lowest cost,;
13:         $U' \leftarrow U' \setminus u'$;
14:         Offload the service of user $u'$ to the offloaded target $k'$;
15:         Update the residual capacity of the offloaded target $k'$;
16:     end while;
17:     if $t \geq 2$ then
18:         Compute the incurred dynamic cost $E_D^t$, by the obtained new
            placement decision at time slot $t$ from Step 6 to Step 16;
19:         if $E_D^t \leq \frac{1}{\beta}\sum_{t'=\hat{t}}^{t-1} E_S^{t'}$ then
20:             Apply the obtained new placement decision at time slot $t$ and
                conduct the corresponding service migrations;
21:             $\hat{t} \leftarrow t$;
22:         else
23:             Apply the past placement decision at time slot $(t-1)$ with no
                service migration;
24:         end if;
25:     end if;
26:     $t \leftarrow t + 1$;
27: end while;
```

C. Algorithm analysis

Lemma 1: Given an MEC network $G(V, E)$ and a set U of mobile users, under the migration control policy in `Algorithm` 1, the total dynamic cost is no larger than $\frac{1}{\beta}$ times the total static cost over the time horizon $\mathcal{T}$, i.e.,

$$\sum_{t=1}^{|\mathcal{T}|} E_D^t \leq \frac{1}{\beta}\sum_{t=1}^{|\mathcal{T}|} E_S^t, \tag{10}$$

where β is a control parameter, E_D^t is the dynamic cost at time slot t and E_S^t is the static cost at time slot t.

Proof: Denote by $\hat{t}_i$ the time slot that the ith migration occurs and $\hat{t}_0 = 1$. And we have $E_D^{\hat{t}_0} = 0$ because we do not need to consider the dynamic cost at the very first time slot. Before the $(i+1)$th service migration, following the migration control policy in `Algorithm` 1, the dynamic migration cost at time slot $\hat{t}_{i+1}$ is no larger than $\frac{1}{\beta}$ times the total static cost occurred in $[\hat{t}_i, \hat{t}_{i+1})$, i.e., $E_D^{\hat{t}_{i+1}} \leq \frac{1}{\beta}\sum_{t=\hat{t}_i}^{\hat{t}_{i+1}-1} E_S^t$.

The total dynamic cost over time horizon $\mathcal{T}$ is the sum of dynamic costs of all service migrations over the time horizon, i.e.,

$$\sum_{t=1}^{|\mathcal{T}|} E_D^t = \sum_i E_D^{\hat{t}_i} \leq \frac{1}{\beta}\sum_{t=1}^{|\mathcal{T}|-1} E_S^t \leq \frac{1}{\beta}\sum_{t=1}^{|\mathcal{T}|} E_S^t. \tag{11}$$

Hence, the lemma follows. ■

Lemma 2: Given an MEC network $G(V, E)$ and a set U of mobile users with service requests, the total static cost in the solution delivered by `Algorithm` 1 for the MDOSP problem is no greater than λ times the optimal solution, i.e.,

$$\sum_{t=1}^{|\mathcal{T}|} E_S^t \leq \lambda \cdot E^*, \tag{12}$$

where E^* is the total cost of the optimal solution in the offline version of the problem and $\lambda = \max_{t\in\mathcal{T}} \frac{\max_{t\in\mathcal{T}} E_S^t}{\min_{t\in\mathcal{T}} E_S^t}$, indicating the maximum ratio of the largest to the smallest possible static cost at any time slot [11].

Proof: By the definition of λ, we have $E_S^t \leq \lambda \cdot E_S^{*t}$, where E_S^{*t} is the static cost in the optimal solution at time slot t with $1 \leq t \leq |\mathcal{T}|$. We then have

$$\sum_{t=1}^{|\mathcal{T}|} E_S^t \leq \lambda \sum_{t=1}^{|\mathcal{T}|} E_S^{*t} \leq \lambda \cdot E^*. \tag{13}$$

Hence, the lemma follows. ■

Theorem 2: Given an MEC network $G(V, E)$ and a set U of mobile users with service requests for a given time horizon $\mathcal{T}$, there is an online algorithm, `Algorithm` 1, with a $\lambda\cdot(1+\frac{1}{\beta})$ competitive ratio for the MDOSP problem, and the algorithm takes $O(|U|^2 \cdot |\mathbb{K}| \cdot |\mathcal{T}| + |V|^3)$ time for service placement over the time horizon $\mathcal{T}$, where $\beta > 0$ is a control parameter, $\lambda = \max_{t\in\mathcal{T}} \frac{\max_{t\in\mathcal{T}} E_S^t}{\min_{t\in\mathcal{T}} E_S^t}$ and $\mathbb{K}$ is the set of offloaded targets.

Proof: As mentioned, the total cost of all offloading tasks for a given time horizon $\mathcal{T}$ consists of both the static cost and the dynamic cost at each time slot, i.e.,

$$E = \sum_{t=1}^{|\mathcal{T}|} E_S^t + \sum_{t=1}^{|\mathcal{T}|} E_D^t \tag{14}$$

$$\leq (1+\frac{1}{\beta})\sum_{t=1}^{|\mathcal{T}|} E_S^t, \quad \text{by Lemma 1,} \tag{15}$$

$$\leq \lambda\cdot(1+\frac{1}{\beta})\cdot E^*, \quad \text{by Lemma 2.} \tag{16}$$

We finally analyze the time complexity of `Algorithm` 1. It takes $O(|V|^3)$ time to compute the shortest paths between each pair of APs in G. And it takes $O(|U|^2 \cdot |\mathbb{K}|)$ time to find the new service placement decisions at the beginning of each time slot. Thus, `Algorithm` 1 takes $O(|U|^2 \cdot |\mathbb{K}| \cdot |\mathcal{T}| + |V|^3)$ time for the service placement over the time horizon $\mathcal{T}$. Hence, the theorem follows. ■

V. Performance Evaluation

In this section, we study the performance of the proposed algorithm by experimental simulations.

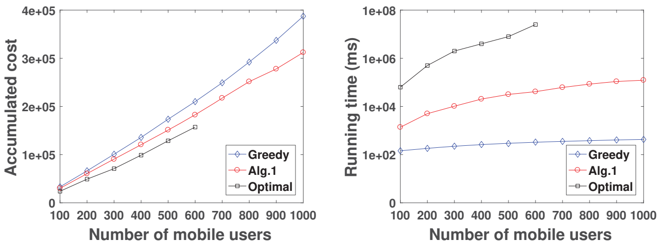
A. Environment settings

The topologies of MEC networks are generated via a tool GT-ITM [7]. We consider an MEC network with 100 APs, 10% of which are co-located with cloudlets. The capacities of cloudlets are randomly drawn between 30 GHz and 150 GHz [4]. We further assume that there are 100 helper mobile devices with computing capacities from 3 GHz to 10 GHz [12]. The amount of computing resource demanded by a mobile user is randomly drawn between 0.4 GHz and 2 GHz [4]. The communication delay of a link ranges from 3ms to 8ms [5]. The cost of per unit computing resource on a helper mobile device varies from $0.1 to $0.4 per GHz, while the cost of per unit computing resource of each cloudlet is randomly drawn from a range between $0.4 to $0.8 per GHz [10]. The coefficient of the communication delay cost is set at 0.1. For simplicity, the price of migrating a service of user u from node v to node v' is set as 0.1 times the product of the consumed computing resource of user u and the communication delay from v to v', i.e., $b_u(v, v') = 0.1 \cdot \rho_u \cdot d(v, v')$. The migration control parameter β is set as 4. There are $|\mathcal{T}| = 20$ time slots, and all mobile users or devices move randomly. The actual running time of each algorithm is based on a desktop with a 3.60 GHz Intel 8-Core i7-7700 CPU and 16 GB RAM. Unless specified, the above parameters are adopted by default.

We evaluate `Algorithm` 1 (referred to as `Alg.1`) against two benchmarks: One is an optimal solution of the offline version of the MDOSP problem obtained by INP (6), referred to as `Optimal`; another is a greedy algorithm `Greedy`, which chooses an offloaded target with the lowest total cost at the current time slot for each mobile user.

B. Performance evaluation of different algorithms

We first studied the performance of different algorithms by varying the number of mobile users from 100 to 1,000. Fig. 1(a) and Fig. 1(b) demonstrate the total costs and running times of different algorithms. It can be seen from Fig. 1(a) that the total cost of `Alg.1` is 19.4% less than that of algorithm `Greedy` with 1,000 mobile users. While the total cost of algorithm `Optimal` is 13.9% less than that of `Alg.1` with 600 mobile users. However, it can be seen from Fig. 1(b) that algorithm `Optimal` takes a prohibitively long time while `Alg.1` takes a much shorter time. Also, when the number of mobile users reaches 700, algorithm `Optimal` fails to deliver any solution within a reasonable time. This is because `Alg.1` establishes a reasonable migration control to avoid aggressive service migrations during each time slot with the updated mobility information.



(a) The total cost

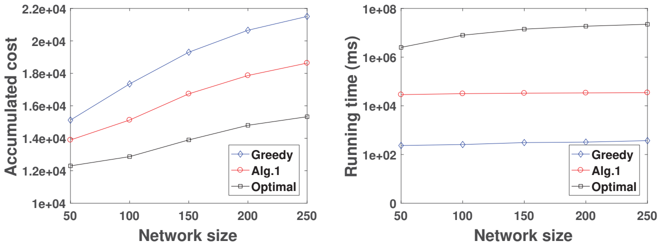
(b) The running time

Fig. 1. Performance of different algorithms by varying the number of mobile users from 100 to 1,000.

C. Impact of different parameters on the proposed algorithm

We then evaluated the impact of important parameters on the proposed algorithms, such as network size, the number of helper mobile devices, and the value range of the control parameter β, while fixing the number of mobile users at 500.

We studied the impact of network size on the performance of the proposed algorithm, by varying the network size from 50 to 250. Fig 2(a) and 2(b) depict the total costs and running times of different algorithms. As shown by Fig 2(a), when the network size is 250, the total cost of algorithm Optimal is 82.3% of that of Alg.1, while the total cost of Alg.1 is 86.7% of that of algorithm Greedy. With the increase on network size, the total cost of each comparison algorithm increases, too. The rationale behind is that a larger network size leads to not only more unpredictable user mobility but also higher delay and service migration costs.



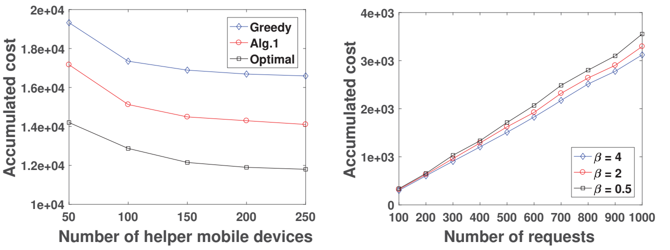
(a) The total cost

(b) The running time

Fig. 2. Impact of network size on different algorithms, by varying the number of nodes from 50 to 250.

We investigated the impact of the number of helper mobile devices on the performance of the proposed algorithm. Fig. 3(a) depicts the total costs of different algorithms, by varying the number of helper mobile devices from 50 to 250. From Fig. 3(a), we can see that the total costs of all mentioned algorithms decrease with the increase on the number of helper mobile devices. And the total cost of algorithm Alg.1 is 15.1% less than that of algorithm Greedy when the number of helper mobile devices is 250, which can be justified by that in our setting, the computing resource of helper mobile devices is relatively cheaper.

We finally evaluated the impact of the control parameter β on the performance of the proposed algorithm. Fig. 3(b) depicts the total cost of the proposed algorithm Alg.1 when $\beta = 0.5$, 2, and 4 respectively. From Fig. 3(b), we can see that when there are 1,000 mobile users, the total cost by algorithm Alg.1 with $\beta = 4$ is 87.8% of itself with $\beta = 0.5$. The rationale behind is that with a larger β, the MEC network intends to be more tolerant on service delays and avoid aggressive service migration costs.



(a) Impact of the number of helper mobile devices

(b) Impact of the migration control parameter β

Fig. 3. Impact of different parameters on the proposed algorithm.

VI. Conclusion

In this paper, we formulated a novel MDOSP problem in a D2D-assisted MEC environment with the aim to minimize the total cost that consists of the computing cost, communication delay cost and migration cost, without the knowledge of future mobility information of mobile users or helper mobile devices. We showed the NP-hardness of the problem, and formulated an Integer Nonlinear Programming (INP) solution to the offline setting of the problem. We then devised an online algorithm with a provable competitive ratio for the problem. We finally evaluated the performance of the proposed algorithms through experimental simulations. Experimental results demonstrated that the proposed algorithm is promising.

References

[1] X. Cao, F. Wang, J. Xu, R. Zhang, and S. Cui. Joint computation and communication cooperation for energy-efficient mobile edge computing. *IEEE Internet of Things Journal*, vol 6, no. 3, pp. 4188 – 4200, 2019.

[2] B. Gao, Z. Zhou, F. Liu, and F. Xu. Winning at the starting line: Joint network selection and service placement for mobile edge computing. *Proc. of INFOCOM'19*, pp. 1459 – 1467. IEEE, 2019.

[3] Y. Kai, J. Wang, and H. Zhu. Energy minimization for d2d-assisted mobile edge computing networks. *Proc. of ICC'19*, IEEE, 2019.

[4] J. Li, W. Liang, M. Huang, and X. Jia. Reliability-aware network service provisioning in mobile edge-cloud networks. *IEEE Transactions on Parallel and Distributed Systems*, vol. 31, no. 7, pp. 1545 – 1558, 2020.

[5] J. Li, W. Liang, W. Xu, Z. Xu, and J. Zhao. Maximizing the quality of user experience of using services in edge computing for delay-sensitive IoT applications. *Proc. of MSWiM'20*, ACM, 2020.

[6] Y. Ma, W. Liang, and S. Guo. Mobility-aware delay-sensitive service provisioning for mobile edge computing. *Proc. of INFOCOM WKSHPS'19*, pp. 270 – 276. IEEE, 2019.

[7] GT-ITM. http://www.cc.gatech.edu/projects/gtitm/.

[8] L. Özbakir, A. Baykasoğlu, and P. Tapkan. Bees algorithm for generalized assignment problem. *Applied Mathematics and Computation*, vol. 215, no. 11, pp. 3782 – 3795, 2010.

[9] L. Wang, L. Jiao, J. Li, J. Gedeon, and M. Mühlhäuser. Moera: Mobility-agnostic online resource allocation for edge computing. *IEEE Transactions on Mobile Computing*, vol. 18, no. 8, pp. 1843 – 1856, 2018.

[10] Z. Xu, W. Liang, M. Huang, M. Jia, S. Guo, and A. Galis. Efficient nfv-enabled multicasting in sdns. *IEEE Transactions on Communications*, vol. 67, no. 3, pp. 2052 – 2070, 2019.

[11] L. Zhang, C. Wu, Z. Li, C. Guo, M. Chen, and F. Lau. Moving big data to the cloud: An online cost-minimizing approach. *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 12, pp. 2710 — 2721, 2013.

[12] R. Zhang, W. Shi, J. Zhang, and W. Liu. An auction scheme for computing resource allocation in d2d-assisted mobile edge computing. *GLOBECOM'19*, pp. 1 – 6, IEEE, 2019.