DMRA: A Decentralized Resource Allocation Scheme for Multi-SP Mobile Edge Computing

Chen Zhang[*], Hongwei Du[*], Qiang Ye[a], Chuang Liu[*], and He Yuan[*]

[*]School of Computer Science and Technology, Harbin Institute of Technology (Shenzhen), Shenzhen, China
[a]Faculty of Computer Science, Dalhousie University, Halifax, Canada

***Abstract*—Mobile Edge Computing (MEC) is a burgeoning paradigm that pushes data and services away from remote clouds to distributed Base Stations (BSs) equipped with MEC servers, which are deployed by Service Providers (SPs) at the edge of cellular networks. Normally, a SP prefers to use its own BSs, instead of those deployed by other SPs, to provide data and storage services. This can not only improve the quality of user experience but also increase its own revenue. In a densely-deployed MEC network where a User Equipment (UE) tends to be covered by multiple BSs from varied SPs, how to allocate the resources in the BSs to provide the best service is a challenging problem. In this paper, we propose a novel resource allocation scheme, Decentralized Multi-SP Resource Allocation (DMRA), for densely-deployed MEC networks in order to maximize the total profit of all SPs and provide high-quality services. Our experimental results indicate that the proposed scheme outperforms the existing resource allocation algorithms for MEC.**

***Index Terms*—Mobile Edge Computing, Resource Allocation, Profit Maximization**

I. Introduction

The integration of cloud computing and mobile computing leads to a novel computation paradigm, Mobile Cloud Computing (MCC). MCC is capable of providing computation and storage resources for mobile devices in a centralized manner [1]. However, over the past years, MCC has encountered a series of challenges. One of the challenges is associated with latency-sensitive applications. Specifically, for applications such as Virtual/Augmented Reality (VR/AR) [2], video streaming [3] and Internet of Things (IoT) [4], MCC can hardly guarantee the quality of services [5]. In order to solve this problem with MCC, Mobile Edge Computing (MEC) was proposed. The main idea adopted by MEC is to push data and services away from remote centralized clouds to distributed nodes with computing and storage resources. For example, Base Stations (BSs) equipped with MEC servers are deployed at the edge of mobile networks, which is closer to User Equipments (UEs) compared with remote clouds [6]. With this new approach, applications with low latency tolerance can deploy their services on MEC servers, which can not only achieve lower latency and reduce traffic load in backbone networks but also greatly alleviate the constraints of UEs (e.g. computation and energy limitation).

With the development of MEC, more and more Service Providers (SPs), such as China Mobile and China Unicom, are expected to deploy their own BSs and MEC servers. Each MEC server hosts a set of services, which are used to process the computing tasks offloaded by UEs. In order to guarantee the quality of service, all SPs desire to deploy BSs in popular areas to meet the requirements of their users. Consequently, the coverage of the BSs from different SPs is likely to overlap and therefore UEs tend to be able to receive the signals from multiple BSs at the same time. In this scenario, a UE should choose one of the available BSs and thereafter offload its computation task to the selected BS. However, a UE is not authorized to access the resources in BSs or remote clouds directly. It has to resort to the SP that it subscribes to in order to complete the offloading task. If the nearby BS does not have enough resources, the offloaded task needs to be forwarded to remote clouds, which increases the transmission delay [7]. Note that it is more cost-efficient for a SP to forward the offloaded task to the BSs deployed by itself than to those deployed by other SPs. Hence, the resource allocation scheme directly determines the profit of each SP.

In our research, we focus on the resource allocation problem in this multi-UE multi-SP environment. Our goal is to find an optimal resource allocation scheme to maximize the total profit of all SPs and provide high-quality services to UEs. The allocation optimization problem in the multi-UE multi-SP environment involves the following two important aspects. First of all, each SP has its own preferred BSs. The impact of UE, BS, and SP on resource allocation has to be taken into consideration. Secondly, the computing tasks offloaded by UEs consume both computing and radio resources of BSs. Therefore, we should jointly consider the limited computing and radio resources that are available in BSs.

Technically, we propose a Decentralized Multi-SP Resource Allocation (DMRA) scheme to find the optimal resource allocation for UEs' computing tasks in a densely-deployed network. With DMRA, the total profit of all SPs is maximized. The main contributions of this paper are presented as follows.

- We formally formulate the resource allocation problem in a densely-deployed network, aiming to maximize the total profit of all SPs. In the formulated problem, the impact of UE, BS and SP on the performance of resource allocation is jointly taken into account.
- We propose a novel resource allocation scheme, which takes a series of factors into consideration. The considered factors include the distance between UE and BS,

Corresponding author: Hongwei Du, hwdu@hit.edu.cn

2575-8411/19/$31.00 ©2019 IEEE
DOI 10.1109/ICDCS.2019.00046

the number of BSs that a UE can reach, the amount of remaining computing and radio resources in BS, and the diversity of services requested by UE.

- We devise a decentralized algorithm to solve the resource allocation problem. Based on the matching theory [8] [9], the proposed algorithm is capable of maximizing the total profit of SPs through transforming the resource allocation problem into a UE-BS matching problem.

The rest of this paper is organized as follows. Section II includes the related work. We describe the system model in Section III and formulate the problem of total SP profit maximization in Section IV. The details of the proposed algorithm are presented in Section V. Section VI includes our simulation results. Finally, Section VII concludes this paper.

II. RELATED WORK

MEC has drawn a wide range of attention both in the academic and industrial community in recent years. The concept of pulling computing resources from remote clouds to edge clouds which are closer to users has been widely considered in previous work. Islam *et al.* [10] proposed the idea of introducing cloud computing facility at the edge of the Internet to leverage the benefits of virtual-clients in the future Internet architecture in conjunction with increasing focus on content production and delivery. They designed an application 'surrogate' running on top of the cloud to support virtual-clients, which was able to simplify the management of the network, giving SPs more opportunities to be directly involved in service delivery, and support services in an efficient way. Ceselli *et al.* [11] designed a mobile edge cloud network architecture for mobile access metropolitan area networks. They considered both static and dynamic status of the network, aiming to correctly place cloudlet on available sites and assign sets of access points. Tong *et al*. [12] proposed to deploy cloud servers at the edge of the network and designed it as a tree-based hierarchy of geo-distributed servers. This architecture aggregated the peak loads across different tiers of the cloud, aiming to maximize the amount of mobile loads being served. Furthermore, a workload placement algorithm was proposed by them to ensure the utilization of cloud resources. Mu *et al.* [13] studied the real-time pricing proplem for local power supplier in smart community. Other than the computation offloading problem [14] [15], service migration problem [16] [17], computation caching problem [18] [19] have been studied in prior works. Resources allocation is another important issue in MEC that should be studied to guarantee the quality of services provided by SPs.

There are a few existing studies that focus on the resource allocation problem to improve the quality of service in MEC. In [20], Sardellitti *et al.* studied the densely-deployed MIMO multi-cell system, in which multiple UEs asked for computation offloading. They formulated a resource optimization problem, aiming at minimizing the energy consumption of users, and both radio and computing resources were considered in the resource allocation process to achieve the joint optimization of the system. In [21], You *et al.* investigated the energy-efficient resource allocation problem in multi-user MEC offloading system based on TDMA/OFDMA. They discussed the performance of the system when clouds have infinite or finite capacity. The aim of this paper was minimizing the sum of mobile energy consumption. However, the above two papers only consider the system with single MEC server. With the increase of the number of latency-sensitive tasks and the complexity of small-cell networks, computing resources supplied by only single MEC server is not enough. So some studies have focused on MEC with multi-user and multi-server in recent years. Tianze *et al.* [22] designed a task scheduling mechanism for ad-hoc based MEC, aiming to minimize the overhead of each UE. They indicated that UEs could cooperate with each other and developed a potential game for their model. Four factors, energy consumption, opportunity consumption, time delay and monetary cost, were taken into account in their work. Xie *et al.* [23] proposed a multi-dimensional pricing scheme based on a two-side market game. In the study, three types of prices are given by them, and a distributed price-adjustment algorithm for resource allocation and QoS-aware offloading scheduling were proposed based on the three prices. The price based mechanism can significantly improve the performance of the system. Zhang *et al.* [24] studied the resource allocation problem in a multi-tier LTE unlicensed network, through combining the Stackelberg game and the bargaining together. In [25], the authors studied computing resources allocation problem to obtain joint optimization among FN (fog nodes), data service subscribers (DSS) and data service operators (DSO) in the three-tier IoT fog network. In this system, DSO got resources from FNs to serve their DSSs. BSs equipped with MEC servers were deployed by SPs in the densely deployed IoT network. Each SP had preference to

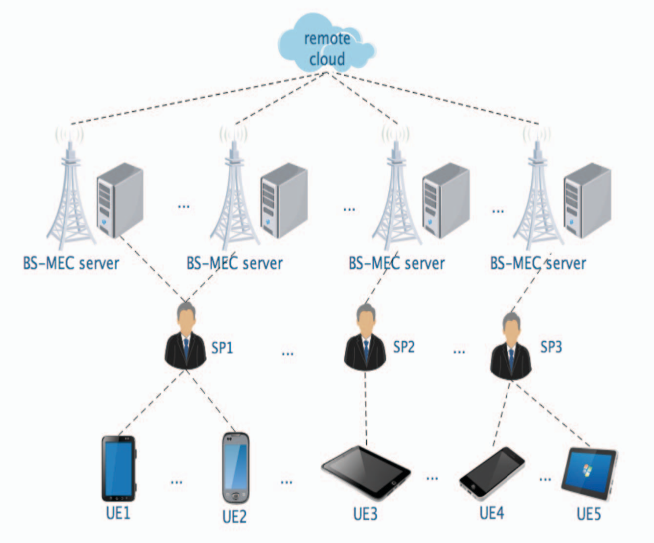

Fig. 1. System architecture

allocate resources in BSs to UEs subscribing to it. They used the matching theory to solve the FN-DSS pairing problem. But this method can not be used in cellular network directly. To the best of our knowledge, no existing work has studied the problem of multi-SP resource allocation in cellular network under consideration of the relationship between UEs, SPs and BSs.

III. SYSTEM MODEL

In this section, the model of the system under investigation is described in detail. We first give an overview of the system model. Then we discuss the details of three aspects of the system under investigation: computing resources, radio resources, and SP utility.

A. System Overview

Fig. 1 includes a generic network architecture that consists of four layers, including UE layer, SP layer, Edge Computing (EC) layer and remote cloud layer. With this architecture, UEs offload their computing tasks to the SPs that they subscribe to. BSs equipped with MECs servers are deployed by SPs to provide MEC services for UEs in EC layer. The offloading tasks that cannot be processed by EC layer will be forwarded to the remote cloud.

Table I includes the notations used in our research. Specifically, ς, U, B and S denote the set of SPs, UEs , BSs and services respectively. In our research, we assume that each UE $u \in U$ subscribes to one SP $k \in \varsigma$ and each BS $i \in B$ is deployed by one SP $k \in \varsigma$. Each BS equipped with a MEC server, which has limited computing and radio resources, provides MEC services to the UEs in its coverage area. Note that, for simplicity, the term "BS" and the term "MEC server" are used interchangeably in this paper. Computing resources in BSs are used to handle the computation tasks offloaded by UEs. Radio resources in BSs are used to receive the offloading-related data from UEs and return the computed results back to them. Each MEC server hosts a service subset $S_i \subseteq S$. In our research, we use the symbol $z_{i,j} \in \{0, 1\}$ to denote the relationship between BS $i \in B$ and service $j \in S$. If BS i hosts service j, then $z_{i,j} = 1$; otherwise $z_{i,j} = 0$.

In a densely-deployed network, the BSs from multiple SPs could be installed to cover the same area. Namely, the coverage area of some BSs might overlap. In this scenario, UEs should choose one of the reachable BSs that host the requested service. In our research, we assume that each UE can only request one MEC service and can only be served by one BS at a time.

Furthermore, we assume that UEs have no authorization to access BSs directly. They need to subscribe to a SP and use the virtual service provided by the SP to offload their computing tasks to BSs. Namely, SPs, as a middle layer, controls the behavior of UEs and BSs. With the coordination of SPs, the resources for each service provided by BSs are allocated to handle the offloaded computing tasks from UEs. To avoid potential network congestion and ensure fast response time, each SP prefers to assign the offloaded computing task to nearby BSs instead of the remote cloud. When all the available BSs do not have the appropriate resources to process the offloaded computing task, the offloading request will be forwarded to the remote cloud, whose capacity is assumed to be unlimited.

TABLE I
LIST OF NOTATIONS

symbol	definition
ς	the set of SPs
U	the set of UEs
B	the set of BSs
S	the set of services
$z_{i,j}$	$z_{i,j} = 1$ means BS i host service j, otherwise $z_{i,j} = 0$
CRU	computing resource unit
$c_{i,j}$	the number of CRU allocated by BS i to service j
$U_i^{'}$	the set of UEs served by BS i
c_j^u	the number of CRU of service j needed to process the computing task offloaded by UE u
W_{sub}	the bandwidth of RRB
W_i	the uplink bandwidth of BS i
N_i	the maximum number of RRBs can allocated by BS i
R_i	the total number of RRBs that are allocated by BS i
w_u	the required data rate for UE u to get service
$e_{u,i}$	The reveived data rate for each RRB of BS i from UE u
$n_{u,i}$	the number of RRBs allocated by BS i to UE u
U_k	the set of UEs subscribing to SP k and task processed by BS nearby
$a_{u,i}$	$a_{u,i} = 1$ means task of UE u is served by BS i, otherwise $a_{u,i} = 0$
W_k	the revenue function of SP k at MEC laye
W_k^r	the total revenue that SP k receives from UE u for its service
W_k^S	the total other cost for SP k to serve UE
W_k^B	the total payment from SP k to all BSs
m_k	the price of unit CRU set by SP k
$p_{i,u}$	the price of unit CRU set by BS i to UE u
$d_{i,u}$	the distance between BS i and UE u
$J_{u,j}$	$J_{u,j} = 1$ means UE u request service j, otherwise $J_{u,j} = 0$

B. Computing Resources

An MEC server can provide computation services to multiple UEs with different service requests concurrently. However, as the capacity of an MEC server is limited, it can only provide a subset of the services in S. The offloaded computation task cannot be forwarded to an MEC server that does not provide the corresponding service. In our research, we use 'Computing Resource Unit (CRU)' as the unit to describe the computing resource allocation in an MEC server. Specifically, we use $c_{i,j}$ to denote the number of CRUs that BS $i \in B$ allocates to service $j \in S$. If $c_{i,j} > 0$, then MEC server i can provide service j. If MEC server i is designed not to handle service j, then $c_{i,j} = 0$. Furthermore, we use $U_i^{'}$ to denote the set of UEs served by BS $i \in B$ (obviously, $U_i^{'} \subseteq U$); and we use $c_j^u (c_j^u \geq 0)$ to denote the amount of CRUs required to process the computing task offloaded by UE u, which corresponds to service j (obviously, $u \in U_i^{'}$). At any time, the total amount of CRUs allocated by BS i to handle service-j-related computing tasks are limited by $c_{i,j}$. Namely, we have:

$$\sum_{u \in U_i^{'}} c_j^u \leq c_{i,j}, \forall i \in B, j \in S \tag{1}$$

C. Radio Resources

Each BS has limited radio resources that can be used to transmit data between UE and BS. In our research, we only consider the uplink radio resource consumption because the size of the task data is usually much larger than that of the result data. In addition, we consider a system with Orthogonal Frequency Division Multiple Access (OFDMA) being the access scheme. The basic unit of radio resource allocation is denoted as Radio Resource Block (RRB). The bandwidth of it is denoted as W_{sub}. Let W_i denote the uplink bandwidth of BS $i \in B$. The maximum number of RRBs in BS i that can be used to process the computing tasks offloaded by UEs is N_i.

The number of RRBs allocated by BSs to UEs is influenced by the data rate requested by UEs. We use w_u to denote the required data rate for UE u to get service. The Signal-to-Interference-plus-Noise-Ratio (SINR) from UE u to BS i is denoted as $\lambda_{u,i}$. The received data rate for each RRB of BS i from UE u is

$$e_{u,i} = W_{sub} log_2(1 + \lambda_{u,i}) \tag{2}$$

The number of RRBs that should be be allocated by BS i to UE u is

$$n_{u,i} = \lceil w_u / e_{u,i} \rceil \tag{3}$$

$e_{u,i}$ is determined by transmit power of UE u, interference power of the other signals in the network and some noise term. The interference power increases with the distance between UE u and BS i. When interference power increases, $e_{u,i}$ will decrease. So when w_u is fixed, the farther the distance between UE u and BS i, the more number of RRB in BS i is needed by UE u.

So the total number of RRBs that are allocated by BS i can be modeled as

$$R_i = \sum_{u \in U_i^{'}} n_{u,i} \tag{4}$$

The total number of RRBs allocated by BS i to UE $u \in U_i^{'}$ cannot exceed N_i.

D. SP Utility

For each SP, the cost of using the resources in the BSs deployed by itself is lower than that of using the resources in the BSs deployed by other SPs. Therefore, the scheme of an SP preferentially allocating resources in the BSs deployed by itself to its own UEs can not only improve the user experience of its UEs, but also increase its own revenue.

UEs need to pay SPs for the service, and SPs have to pay BSs and remote clouds for using their resources. Let U_k denotes the set of UEs subscribing to SP k and whose computing task processed by nearby BSs. We define the variable $a_{u,i} \in \{0, 1\}$ for BS $i \in B$, UE $u \in U$. If the offloaded task of UE u is served by BS i, then $a_{u,i} = 1$; otherwise $a_{u,i} = 0$. The utility function of SP $k \in \varsigma$ at MEC layer is defined as:

$$W_k = W_k^r - W_k^B - W_K^S \tag{5}$$

$$W_k^r = \sum_{u \in U_k} \sum_{j \in S} c_j^u m_k \tag{6}$$

$$W_k^B = \sum_{u \in U_k} \sum_{i \in B} \sum_{j \in S} a_{u,i} p_{i,u} c_j^u \tag{7}$$

$$W_K^S = \sum_{u \in U_k} \sum_{j \in S} c_j^u m_k^o \tag{8}$$

where W_k^r is the total revenue that SP k receives from UE $u \in U_k$ for its service; m_k denotes the CRU price set by SP k; W_k^S is the total other cost for SP k to serve UE $u \in U_k$; m_k^o denotes the price of CRU of other cost for SP k. m_k and m_k^o are two constants; W_k^B denotes the total payment from SP k to all BSs; $p_{i,u}$ is the CRU price set by BS i to UE u and it satisfies $m_k > p_{i,u} + m_k^o$. $p_{i,u}$ can be calculated using the following equation:

$$p_{i,u} = \begin{cases} b + d_{i,u}^{\sigma} b & u \text{ and } i \text{ from same SP} \qquad (9) \\ \iota b + d_{i,u}^{\sigma} b & u \text{ and } i \text{ from different SP} \quad (10) \end{cases}$$

Here, both computing resource price and transmission price are taken into consideration. When UE u and BS i belong to the same SP, the price of one CRU of computing resource is b; otherwise the price is ιb ($\iota > 1$). It is much cheaper for SPs to use resources of their own than those belonging to other SPs when the requested services are the same. Using resources in remote clouds is the most expensive option, because the cost of time and energy used to transmit the offloaded task of UEs to the clouds is much higher than that for nearby BSs. $d_{i,u}$ is the distance between BS i and UE u, which has increases with the transmission cost in a linear fashion. The farther the distance between UE and BS, the higher the energy consumption of the transmission (i.e. the higher the transmission price). Furthermore, ι and σ are two weight parameters.

IV. PROBLEM FORMULATION

With the proposed architecture, SPs prefer to assign the computing tasks offloaded by UEs to the EC layer rather than the remote cloud in order to avoid potential network congestion, ensure fast response time, and improve the provided Quality of Service (QoS) and Quality of Experience (QoE).

SPs, as a middle layer in the proposed architecture, controls the interaction behavior of UEs and BSs. As mentioned previously, each SP preferentially allocates the resources in the BSs deployed by itself to its own UEs to ensure the QoE of the UEs subscribing to itself. For SP k, the cost of using the resources in BS $i \in B$ deployed by itself to process the computing task offloaded by UE u who subscribes to it is lower than that of using the resources in BS $i^{'} \in B$ deployed by other SPs when the distance between UE u and BS i is equal to that between UE u and BS $i^{'}$.

For a batch of UEs with computing tasks, their location and requested services are known. In our research, we define the variable $J_{u,j} \in \{0, 1\}$ for UE $u \in U$ and service $j \in S$. If UE

u requests service j, then $J_{u,j} = 1$; otherwise $J_{u,j} = 0$. The scheme of allocating limited computing resources and radio resources of the BSs to UEs directly affects the revenue of SPs.

Consequently, each SP attempts to maximize its profit through using the optimal resource allocation scheme to allocate resources to UEs with offloading computing tasks and subscribing to it. The problem we concern is determining the value of $a_{u,i}$. Our goal is to find an optimal UE-BS association scheme to maximize the total profit of all SPs at the MEC layer. The Total Profit Maximization problem (TPM) can be defined as follows.

Definition 1: The Total Profit Maximization problem (TPM) can be formulated as the following optimization problem:

$$\max_{a_{u,i}, \forall u \in U, \forall j \in S} \quad \sum_{k \in \varsigma} W_k \tag{11}$$

$$\text{s.t.} \quad \sum_{u \in U_i'} c_j^u \le c_{i,j}, \forall i \in B, j \in S \tag{12}$$

$$a_{u,i} \le z_{i,j} J_{u,j}, \forall i \in B, \forall j \in S, \forall u \in \delta \tag{13}$$

$$\sum_{u \in U_i'} n_{u,i} \le N_i \tag{14}$$

$$\sum_{i \in B} a_{u,i} \le 1, \forall u \in U \tag{15}$$

$$m_k > p_{i,u} + m_k^o, \forall j \in S, \forall u \in U_k, \forall k \in \varsigma \tag{16}$$

Technically, there are five constraints involved in the optimization problem. The constraint corresponding to Notation (12) shows that the amount of CRUs from BS i used to process computing tasks offloaded by UEs must satisfy the capacity constraint of BS i. The constraint corresponding to Notation (13) states that the premise that BS i can be associated with UE u is that BS i has the service j requested by UE u. The constraint corresponding to Notation (14) shows that the total number of RRBs allocated by BS cannot exceed its maximum capacity. The constraint corresponding to Notation (15) indicates that the computing task offloaded by each UE can be processed by at most one BS. The constraint corresponding to Notation (16) shows that it is profitable for each SP to provide service to its users.

V. DMRA: A DECENTRALIZED SCHEME

The key to the TPM problem is to find out the best association scheme for UEs and BSs. To find the best association, we need to try all possible combinations of UEs, BSs and remote cloud, which is impractical for large-scale distributed networks without a centralized control center. Another obstacle is that each SP needs to adjust its resource allocation strategy in real time to adapt its network to the changing environment. Namely, the best association changes over time.

The association between UEs and BSs can be regarded as a matching problem. In this paper, we propose an improved matching algorithm, Decentralized Multi-SP Resource Allocation (DMRA), to solve the TPM problem. The matching problem for UEs and BSs is similar to the classic Stable Marriage Problem (SMP) in mathematics although there are a couple of differences. The first difference between them is that the preference list of men and women in SMP is fixed while the preference list of UEs and BSs vary over time. The second difference is that each UE only needs to pay attention to the BSs that are reachable and can provide the requested service. It does not need to consider all BSs.

An overview of DMRA is presented as follows. With DMRA, SPs, as the middle layer, receive the computing tasks offloaded by UEs and help each UE be associated with an appropriate BS or remote cloud via multiple iterations. In each iteration, unserved UE first proposes its most preferred BS. Each BS builds a preference list for each provided service. The list includes the UEs that have proposed to the BS as the most preferred one. Once the list is available, the BS associates itself with its most preferred UE in this list. It is profitable for BSs to provide service to UEs. The cost of using resources in remote cloud is much more expensive than that of using resources in BSs. In addition, the distance between UE and BS determines the transmission delay and user experience. Therefore, SPs should forward the computing tasks offloaded by UEs to nearby BSs whenever possible. We consider two factors when we determine the preference of UEs for BSs. The first factor is the price of CRU set by BS. The second factor is the remaining radio resources and computing resources corresponding to the service requested by UE. Let $v_{u,i}$ denote the preference of UE u for BS i, then we have:

$$v_{u,i} = p_{i,u} + \rho / [(c_{i,j} - \sum_{u \in U_i'} c_j^u) + (N_i - \sum_{u \in U_i'} n_{u,i})] \tag{17}$$

where ρ is a parameter that determine the choice of BS. The more the amount of remaining computing and radio resources in BSs, the greater the probability that UE's task will be processed by nearby BSs. In each iteration, UE u proposes to BS i with the smallest $v_{u,i}$. Each service in a BS preferentially selects one UE belonging to the same SP. Let f_u denote the number of BSs which can cover UE u and have available computing and radio resources. When there are multiple UEs that satisfy the condition, we select the UE with smallest f_u. If there are more than one UE that can be selected, we choose the UE u with smallest $(n_{u,i} + c_j^u)$.

The details of the proposed decentralized algorithm are summarized as Alg. 1. Each UE $u \in U$ initializes a set B_u, which includes all BSs which it can reach and host the service requested by it. If B_u is empty, it means there is no BS that can be associated with the UE, and the request of the UE will be forwarded to remote cloud. In each iteration, the UEs whose B_u is not empty and that have not been associated with BSs in the previous iteration will firstly choose its most preferred BS i' in set B_u (Lines 3-10). If the chosen BS has enough computing and radio resources to process the task offloaded by UE u, then UE u will propose to BS i' and send its service request, which includes the information about the location, service demands of u, the number of BSs which can cover u and the SP that u subscribes to. Otherwise, delete BS i' from set B_u and choose another BS in set B_u, as the resources in

BS cannot increase, the BS $i^{'}$ cannot be associated with UE u in subsequent iterations. If the appropriate BS is not found until B_u is empty, the computing task offloaded by UE u will be forwarded to remote clouds for processing.

Each service in each BS maintains a preference list. These lists are set to be empty at first, because no UE has chosen a BS at that time. When BSs receive new service requests in each iteration, they will build their candidate UEs-set $U_{i,j}^c$ after all UEs $u \in U$ have sent their service requests. Service j in BS i prefers to make a proposal to UEs $u \in U_{i,j}^c$ belonging to the same SP. When there are more than one UE maintain this, we will choose the UE with the smallest f_u. If there are more than one UE in set $U^{'}$, we will choose UE u with the smallest $(n_{u,i} + c_j^u)$ (refer to line 13-21 of Alg. 1). After all services in BS have selected their most preferred UE, then add the selected UEs into UE set U^*. We need to check whether the sum of radio resources requested by UE $u \in U^*$ exceeds the remaining amount of radio resources of the BS i. If not, associate these UEs with BS i and update the remaining amount of resources in each service of BS i; otherwise, calculate the preferences of BS i for UEs $u \in U^*$ and sort them in descending order. Select the first few UEs meeting radio resource constraint of BS i and associate these UEs with BS i (refer to line 22-25 of Alg. 1). Finally, BS i broadcasts the connection information to the UEs covered by itself. The iteration continues until there is no UE that sends service request.

Let $|U|$, $|B|$, $|S|$ denote the number of elements in set U, B, S respectively. The complexity for UEs to propose to BSs is $O(|U|)$, the complexity for services in BSs to propose to UEs is $O(|B * S|)$. In terms of the interaction between BSs and UEs, the complexity is $O(|B * U|)$. So the complexity of algorithm DMRA is $O(|u|^2 * |B| + |B|^2 * |U| * |S|)$.

Algorithm 1: Decentralized Multi-SP Resource Allocation (DMRA)

```
   Input: $U$, $B$, $S$, $\delta$, $c_{i,j}$, $c_j^u$, $J_{u,j}$, $z_{i,j}$, $N_i$, $w_u$, $\lambda_{u,i}$,
          $\forall u \in U$, $\forall i \in B$, $\forall j \in S$, $\forall k \in \delta$
   Output: $a_{u,i}$, $\forall u \in U$, $\forall i \in B$
1  initialize $a_{u,i} = 0$, $flag_i = 0$, $U^* = \phi$, compute the set
   $B_u$ of BSs which can cover UE $u$ and provide the
   service UE requests, $\forall u \in U$, $\forall i \in B$;
2  repeat
3      for $u \in U$ do
4          while $B_u \neq \phi$ and $\sum_{i \in B_u} a_{u,i} = 0$ do
5              select $i^{'} = argmin\ v_{u,i}, i \in B_u, u \in U$ ;
6              if $c_{i^{'},j} \geq c_j^u$ and $N_{i^{'}} \geq w_u$ then
7                  send service request to BS $i^{'}$;
8                  break;
9              else
10                 $B_u = B_u - \left\{i^{'}\right\}$;
11     for $i \in B$ do
12         if there is any new incoming service request then
13             summarize the service set $S_i^{'}$ of BS $i$
               requested by UEs and the set $U_{i,j}^c$ of
               candidate UEs that send service requests to
               BS $i$ for service $j$;
14         for $j \in S_i^{'}$ do
15             Divide $u \in U_{i,j}^c$ into two sets $U_1$ and $U_2$,
               $u \in U_1$ with BS $i$ belong to the same SP;
               $u \in U_2$ with BS $i$ belong to different SP;
16             if $U_1 \neq \phi$ then
17                 compute the set $U^{'} \subseteq U_1$ of UEs with
                   $argminf_u, u \in U_1$;
                   $u^{'} = argmin(n_{u,i} + c_j^u), u \in U^{'}$;
18             else
19                 compute the set $U^{'} \subseteq U_2$ of UEs with
                   $argminf_u, u \in U_2$ ;
20                 $u^{'} = argmin(n_{u,i} + c_j^u), u \in U^{'}$;
21             $U^* = U^* + \left\{u^{'}\right\}$; $w = w + w_{u^{'}}$; $c_j^{'} = c_j^{u^{'}}$ ;
22         if $w \leq N_i$ then
23             $N_i = N_i - w$; $c_{i,j} = c_{i,j} - c_j^{'}$, $\forall j \in S_i$;
24         else
25             rank UE $u \in U^*$ according to the preference
               of BS $i$ to the UE $u$ in ascending order and
               remove $u \in U^*$ in order until
               $\sum_{u \in U^*} w_u \leq N_i$; $N_i = N_i - \sum_{u \in U^*} w_u$; get
               the service set $S^*$ requested by UE $u \in U^*$;
               $c_{i,j} = c_{i,j} - c_j^{'}, \forall j \in S^*$;
26         $U^* = \phi$; set $a_{u,i} = 1$ and send the message to
           UEs $u$, $u \in U^*$; Broadcast the remaining
           resousrces $c_{i,j}, \forall j \in S$ and $N_i$ of BS $i$ to UEs
           covered by it;
27 until No UE send service request;
```

VI. EXPERIMENTAL RESULTS

In this section, we present the configuration of our simulations and the details of the experimental results.

A. Simulation Setup

In our simulations, we consider a densely-deployed network. Specifically, there are 5 SPs in the experimental network. Each SP deploys 5 BSs, each of which provides six services. Two different BS placement methods are considered in our simulation. With the first placement method, BSs are placed regularly, with the inter-site distance being 300 meters. With the second placement method, BSs are placed randomly in a 1200m x 1200m rectangle. UEs with a variety of different service requests are distributed randomly in the network. The number of CRBs allocated by BS i to service j is set to a number in the range of 100 to 150. The number of CRBs used to process the computing request offloaded by UE u varies from 3 to 5. The required data rate w_u for UE u is in the range of 2Mbps to 6Mbps. The bandwidth of the uplink channel for each BS is 10MHz. The bandwidth of each RRB is 180kHz. The transmission power of a UE is 10dBm and

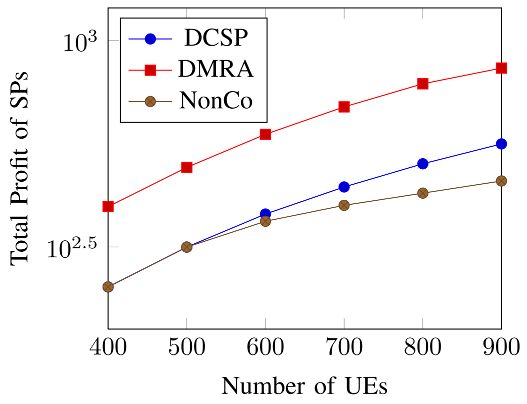

Fig. 2. Total profit of SPs vs. number of UEs ($\iota = 2$, regular BS placement)

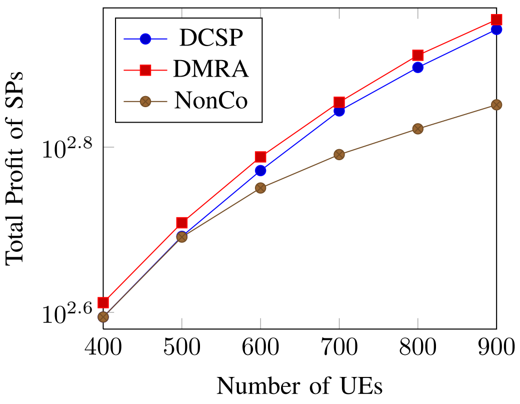

Fig. 4. Total profit of SPs vs. number of UEs ($\iota = 1.1$, regular BS placement)

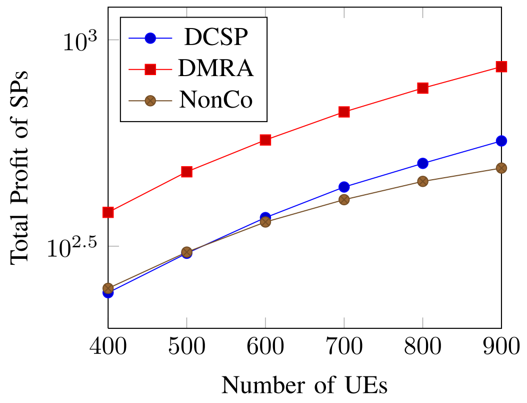

Fig. 3. Total profit of SPs vs. number of UEs ($\iota = 2$, random BS placement)

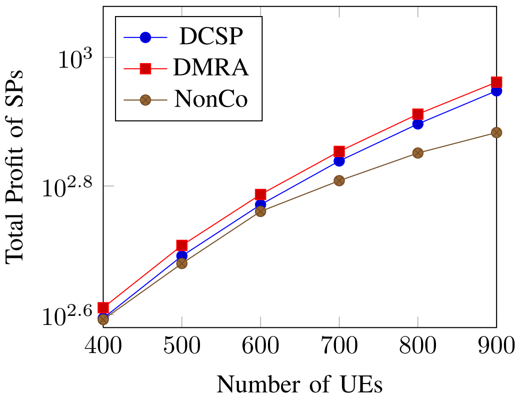

Fig. 5. Total profit of SPs vs. number of UEs ($\iota = 1.1$, random BS placement)

the uplink channel follows the distance-dependent pass-loss model, which is:

$$140.7 + 36.7log_{10}d_{i,u}(km) \tag{18}$$

The noise in the uplink channel is -170dBm. In addition, we set the weight parameter $\sigma = 0.01$.

B. Details of Experimental Results

In our research, we compared the proposed scheme, DMRA, with two state-of-the-art resource allocation methods: Decentralized Collaboration Service Placement (DCSP) [26] and Non-Collaboration (NonCo) algorithm. Technically, DCSP jointly considers service placement and UE association. Each time, UE proposes to BS with the lowest resource occupation, and BS proposes to UE with the smallest number of BSs that can cover it. If more than one UE satisfy the condition, BS chooses the UE which consumes the least amount of radio resources. The iteration is repeated until no UE sends service requests any more. With NonCo, each UE proposes to BS with the maximum SINR in the uplink channel. Each BS prefers to be associated with the UE consuming the least number of RRBs. The collaboration of BSs is not taken into consideration in this algorism.

Fig. 2-5 shows the performance of the schemes under investigation from the perspective of the total profit of SPs vs. the number of UEs, by using algorithm DMRA, NonCo and DCSP respectively. The BS placement method used in the scenarios corresponding to Fig. 2 and 4 are the regular approach, while those corresponding to Fig. 3 and 5 are the random approach. Our experimental results indicate that, for all the schemes under investigation, the total profit of SPs increases with the number of UEs requesting computing services. As the number of UEs requesting services goes up from 400 to 900, the increase rate of the total profit of SPs becomes smaller. That is because, with the increase of the number of computing tasks offloaded by UE, the amount of resources available in nearby BSs decreases gradually. As a result, more and more requests are forwarded to remote clouds. When the resources in BSs are used up, the profit of SP remains unchanged. The weight parameter ι influences the price of using the resources in BSs. The greater the parameter ι, the less $p_{i,u}$ is determined by the distance between BSs and UEs, which means more SPs prefer to choose BSs deployed by themselves. When the weight parameter $\iota = 1$, $p_{i,u}$ is only determined by the distance between BSs and UEs. Note that, in all the scenarios under investigation, DMRA leads to the

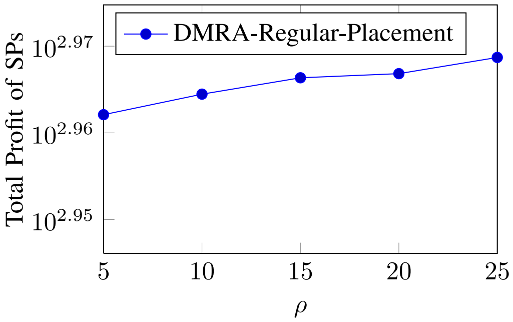

Fig. 6. Total profit of SPs vs. ρ ($\iota = 2$, number of UEs=1000)

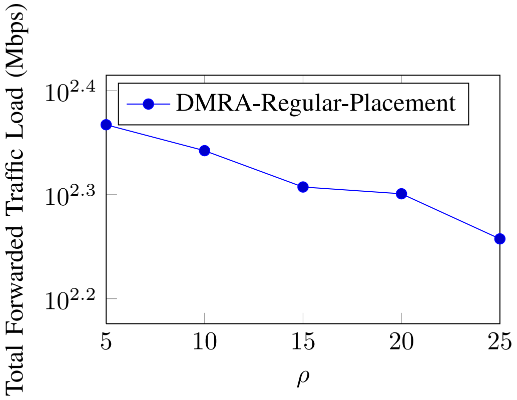

Fig. 7. Total forwarded traffic load vs. ρ ($\iota = 1.1$, number of UEs=1000)

highest total profit of SPs, which clearly shows the advantage of DMRA over DCSP and NonCo.

Fig. 6 and Fig. 7 shows the performance of DMRA in terms of the total profit of SPs and the total forwarded traffic load respectively. The BS placement method adopted in these two simulations are the regular approach. The total number of tasks offloaded by UE is 1000. As there are many UEs requesting resources at the same time, the resources in nearby BSs are not enough. Consequently, part of computation requests offloaded by UEs are forwarded to remote clouds. The weight factor ρ influences the preference of UEs. The greater the weight factor ρ, the higher the number of UEs that prefer to propose to the BSs with more available resources and pay less attention to the price of using resources in BS. As a result, more tasks will be processed by nearby BSs; the total amount of forwarded traffic load will decrease; and the total profit of SPs will go up, which is consistent with the results in Fig. 6 and 7.

VII. Conclusion

In this paper, we study the resource allocation problem in a densely-deployed MEC network. Our goal is to maximize the total profit of all SPs in EC layer. Technically, we transform the resource allocation problem into a UE-BS matching problem, then a decentralized algorithm, DMRA, is proposed to solve the problem. DMRA can continuously adjust the resource allocation scheme according to the demands of the UEs and the remaining amount of resources in the BSs through recalculating the preference relationship between UEs and BSs during each iteration. Our experimental results indicate that the proposed scheme outperforms the existing resource allocation algorithms for MEC.

VIII. Acknowledgment

This study was supported by National Natural Science Foundation of China (No. 61772154). It was also supported by National Sciences and Engineering Research Council of Canada (No. RGPIN-2017-05853).

References

[1] Q. Zhang, L. Cheng, and R. Boutaba. Cloud computing: state-of-the-art and research challenges. *Journal of Internet Services and Applications*, 1(1):7–18, May 2010.

[2] T. Verbelen, P. Simoens, F. De Turck, and B. Dhoedt. Leveraging cloudlets for immersive collaborative applications. *IEEE Pervasive Computing*, 12(4):30–38, Oct 2013.

[3] G. Ananthanarayanan, P. Bahl, P. Bodk, K. Chintalapudi, M. Philipose, L. Ravindranath, and S. Sinha. Real-time video analytics: The killer app for edge computing. *Computer*, 50(10):58–67, 2017.

[4] X. Sun and N. Ansari. Edgeiot: Mobile edge computing for the internet of things. *IEEE Communications Magazine*, 54(12):22–29, December 2016.

[5] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash. Internet of things: A survey on enabling technologies, protocols, and applications. *IEEE Communications Surveys Tutorials*, 17(4):2347–2376, Fourthquarter 2015.

[6] Y. Hu, M. Patel, D. Sabella, N. Sprecher, and V. Young. Mobile edge computinga key technology towards 5g. *ETSI white paper*, 11(11):1–16, 2015.

[7] H. Zhang, Y. Xiao, S. Bu, D. Niyato, F. R. Yu, and Z. Han. Computing resource allocation in three-tier iot fog networks: A joint optimization approach combining stackelberg game and matching. *IEEE Internet of Things Journal*, 4(5):1204–1215, Oct 2017.

[8] D. Gale and L. Shapley. College admissions and the stability of marriage. *The American Mathematical Monthly*, 69(1):9–15, 1962.

[9] Y. Gu, W. Saad, M. Bennis, M. Debbah, and Z. Han. Matching theory for future wireless networks: fundamentals and applications. *IEEE Communications Magazine*, 53(5):52–59, May 2015.

[10] S. Islam and J. Grégoire. Network edge intelligence for the emerging next-generation internet. *Future Internet*, 2(4):603–623, 2010.

[11] A. Ceselli, M. Premoli, and S. Secci. Mobile edge cloud network design optimization. *IEEE/ACM Transactions on Networking (TON)*, 25(3):1818–1831, 2017.

[12] L. Tong, Y. Li, and W. Gao. A hierarchical edge cloud architecture for mobile computing. In *IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications*, pages 1–9, April 2016.

[13] L. Mu, N. Yu, H. Huang, H. Du, and X. Jia. Distributed real-time pricing scheme for local power supplier in smart community. In *2016 IEEE 22nd International Conference on Parallel and Distributed Systems (ICPADS)*, pages 40–47, Dec 2016.

[14] X. Chen. Decentralized computation offloading game for mobile cloud computing. *IEEE Transactions on Parallel and Distributed Systems*, 26(4):974–983, April 2015.

[15] Y. Mao, J. Zhang, and K. B. Letaief. Dynamic computation offloading for mobile-edge computing with energy harvesting devices. *IEEE Journal on Selected Areas in Communications*, 34(12):3590–3605, Dec 2016.

[16] S. Wang, R. Urgaonkar, T. He, K. Chan, M. Zafer, and K. K. Leung. Dynamic service placement for mobile micro-clouds with predicted future costs. *IEEE Transactions on Parallel and Distributed Systems*, 28(4):1002–1016, April 2017.

[17] A. Ksentini, T. Taleb, and M. Chen. A markov decision process-based service migration procedure for follow me cloud. In *2014 IEEE International Conference on Communications (ICC)*, pages 1350–1354, June 2014.

[18] L. Chen and J. Xu. Collaborative service caching for edge computing in dense small cell networks. *arXiv preprint arXiv:1709.08662*, 2017.

[19] L. Yang, J. Cao, G. Liang, and X. Han. Cost aware service placement and load dispatching in mobile cloud systems. *IEEE Transactions on Computers*, 65(5):1440–1452, May 2016.

[20] S. Sardellitti, G. Scutari, and S. Barbarossa. Joint optimization of radio and computational resources for multicell mobile-edge computing. *IEEE Transactions on Signal and Information Processing over Networks*, 1(2):89–103, June 2015.

[21] C. You, K. Huang, H. Chae, and B. Kim. Energy-efficient resource allocation for mobile-edge computation offloading. *IEEE Transactions on Wireless Communications*, 16(3):1397–1411, March 2017.

[22] L. Tianze, W. Muqing, Z. Min, and L. Wenxing. An overhead-optimizing task scheduling strategy for ad-hoc based mobile edge computing. *IEEE Access*, 5:5609–5622, 2017.

[23] K. Xie, X. Wang, G. Xie, D. Xie, J. Cao, Y. Ji, and J. Wen. Distributed multi-dimensional pricing for efficient application offloading in mobile cloud computing. *IEEE Transactions on Services Computing*, pages 1–1, 2018.

[24] H. Zhang, Y. Xiao, L. X. Cai, D. Niyato, L. Song, and Z. Han. A hierarchical game approach for multi-operator spectrum sharing in lte unlicensed. In *2015 IEEE Global Communications Conference (GLOBECOM)*, pages 1–6, Dec 2015.

[25] H. Zhang, Y. Xiao, S. Bu, D. Niyato, F. R. Yu, and Z. Han. Computing resource allocation in three-tier iot fog networks: A joint optimization approach combining stackelberg game and matching. *IEEE Internet of Things Journal*, 4(5):1204–1215, Oct 2017.

[26] N. Yu, Q. Xie, Q. Wang, H. Du, H. Huang, and X. Jia. Collaborative service placement for mobile edge computing applications. In *2018 IEEE Global Communications Conference (GLOBECOM)*, pages 1–6, Dec 2018.