Joint Radio and Computational Resource Allocation in IoT Fog Computing

Yunan Gu, *Student Member, IEEE*, Zheng Chang, *Senior Member, IEEE*, Miao Pan, *Member, IEEE*, Lingyang Song, *Senior Member, IEEE*, and Zhu Han, *Fellow, IEEE*

***Abstract*—The current cloud-based Internet-of-Things (IoT) model has revealed great potential in offering storage and computing services to the IoT users. Fog computing, as an emerging paradigm to complement the cloud computing platform, has been proposed to extend the IoT role to the edge of the network. With fog computing, service providers can exchange the control signals with the users for specific task requirements, and offload users' delay-sensitive tasks directly to the widely distributed fog nodes at the network edge, and thus improving user experience. So far, most existing works have focused on either the radio or computational resource allocation in the fog computing. In this work, we investigate a joint radio and computational resource allocation problem to optimize the system performance and improve user satisfaction. Important factors, such as service delay, link quality, mandatory benefit, and so on, are taken into consideration. Instead of the conventional centralized optimization, we propose to use a matching game framework, in particular, student project allocation (SPA) game, to provide a distributed solution for the formulated joint resource allocation problem. The efficient SPA-(S,P) algorithm is implemented to find a stable result for the SPA problem. In addition, the instability caused by the external effect, i.e., the interindependence between matching players, is removed by the proposed user-oriented cooperation (UOC) strategy. The system performance is also further improved by adopting the UOC strategy.**

***Index Terms*—Fog computing, IoT, resource allocation, matching theory, student project allocation.**

I. Introduction

INTERNET of things (IoT) which supports ubiquitous information exchange and content sharing among smart devices with little or no human intervention is a key enabler for various applications such as smart city, smart grid, smart health, intelligent transportation systems, and so on [1], [2]. Cloud computing is an Internet-based computing platform that provides shared processing resources and data to computers and other devices on demand [3]. In particular, mobile cloud computing (MCC), as a combination of cloud computing, mobile computing and wireless networks, has made it possible for the mobile users to access the cloud resources to offload the computational-intensive tasks [4]. By integrating the cloud into the IoT platform, information collected from end users can be exchanged and processed through cloud-based devices, thus enabling a wide range of new services such as connected vehicles, smart grid, wireless sensor networks and health system monitoring. However, moving the data generated at the IoT edges to the network core (i.e., cloud) has brought new issues, such as the data transferring expense, cloud storing cost, Internet access management and security issues.

On the other hand, to be implemented in the next generation wireless networks, the IoT platform is facing not only the volume, velocity and variety increase regarding the communication contents, but also the emerging of new communication specifications, such as quality of service (QoS), location awareness, real-time mobility support, and latency-sensitive requirements. Therefore, it requires a new designed cloud-based IoT framework to meet these critical requirements for the next generation communication network [5]. CISCO first proposed the idea of Fog Computing in 2014, as a platform that exists between the end devices and the cloud data centers, to provide compute, storage and communication resources to the close proximity of mobile users [6].

Fog computing bringing the cloud closer to the end users, processes and analyzes the most time-sensitive data at the network edge instead of sending them to the cloud [6]. Typically located at the network edge, the fog nodes (FNs), which provide storage, computation and communication capabilities, are characterized with low latency, wide-spread distribution, support for mobility, heterogeneity, interoperability and federation [5]. As the layered architecture shown in Fig. 1, the fog computing extends the cloud computing by introducing an intermediate fog layer between the mobile users/IoT layer and the cloud. A FN can be a cellular base station, Wi-Fi access point or femtocell router with upgraded CPU and memories in either fixed locations, such as a bus, a shopping mall and a road side unit, or being mobile. With communication ability, FNs can communicate with nearby users for both control signal communication and data transmission. However, this direct communication may cause the security issues without the surveillance and protection from the cloud security system, such as eavesdropping and data hijack. One

Manuscript received September 11, 2017; revised February 7, 2018; accepted March 23, 2018. Date of publication March 29, 2018; date of current version August 13, 2018. This work was supported in part by the U.S. National Science Foundation under Grants US CNS-1343361, CNS-1350230 (CAREER), CNS-1646607, CNS-1702850, CNS-1717454, CNS-1731424, and ECCS-1547201. The review of this paper was coordinated by S. He. *(Corresponding author: Zheng Chang.)*

Y. Gu is with the IP Technology Research Department, Huawei, Beijing 100031, China (e-mail: ygu6@uh.edu).

Z. Chang is with the Faculty of Information Technology, University of Jyväskylä, Jyväskylä FIN-40014, Finland (e-mail: zheng.chang@jyu.fi).

L. Song is with the School of Electrical Engineering and Computer Science, Peking University, Beijing 100080, China (e-mail: lingyang.song@pku.edu.cn).

M. Pan is with the University of Houston, Houston, TX 77004 USA (e-mail: mpan2@uh.edu).

Z. Han is with the University of Houston, Houston, TX 77004 USA, and also with the Department of Computer Science and Engineering, Kyung Hee University, Seoul 02447, South Korea (e-mail: zhan2@uh.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TVT.2018.2820838

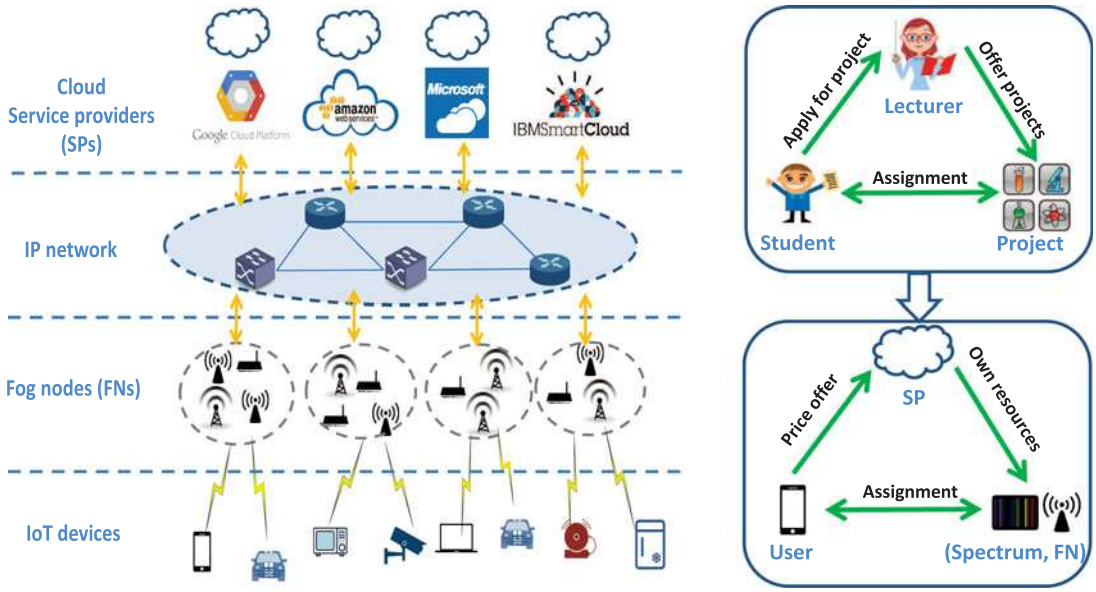


Fig. 1. System model. (a) Fog computing architecture. (b) Student project allocation modeling.

way to avoid the security issues is to transfer them from the FN side to the cloud system side. In other words, the cloud, as the centralized controller of all the FNs and the users, will be responsible for the security controls, including authentication, authorization and so on. Thus, the communication between FNs and users only involves the computation/storage data.

Currently, there are some major obstacles that can limit the deployment and performance of MCC and fog computing. A lot of research has been done on studying how to efficiently allocate the cloud/fog computational resources to various users with heterogeneous requirements, especially on the offloading problem [7], [8]. In [7], the authors investigate a multi-user computation offloading problem for mobile-edge cloud computing in a multi-channel wireless interference environment and propose a game theoretic approach for distributed computation offloading solution. A dynamic offloading framework for extending the lifetime of mobile users is discussed in [8]. The proposed algorithm, based on Lyapunov optimization, is able to extend the battery lifetime while satisfying the execution time requirement. The energy efficiency issue of mobile users during the cloud computing is also discussed [9], [10]. For example, [9] studies the optimal offloading problem in fog computing system to minimize the energy consumption and delay performance. By reconfiguring the offloading probability and varying the transmit power, the objective is to conserve energy for the mobile device and minimize the service delay. In [10], the wireless powered mobile device model is considered. A framework for energy efficient computing is proposed that comprises a set of policies for configuring CPU cycles of local computing and offloading probability. The above mentioned works are all solved in a centralized way. However, the large scale and high mobility features of IoT and the cellular network have made the centralized optimization less efficient, with respect to (w.r.t.) extremely high computation complexity and heavy communication overhead. In addition with the self-organizing feature of the next generation communications, distributive solutions have become more and more needed. Game theory [11], as a popular mathematical framework, has already been applied in the resource allocations of MCC. However, it is worth noticing that there are some shortcomings for using game-theoretic approaches. For example, some knowledge of the other players' actions are required in the classical game-theoretic algorithms, which is hard to be implemented in a distributed manner. Second, in some practical cases, the specific structure in the objective functions of the game-theoretic methods may not always be satisfied [13]. In [12], a joint radio and computation resource allocation in cloud computing is discussed, with user energy and delay requirements considered. The optimization problem is solved in a distributive way, however only one centralized cloud provider is considered without FN.

Considering the above mentioned research challenges in fog computing, we want to study a joint radio access and computational resources allocation when optimizing the system performance. The important factors, such as, transmit power, service latency, and transmission quality, can be jointly considered. To the best of our knowledge, this work is the first attempt to investigate the joint radio and computational resource allocation problem with multiple cloud providers in the fog computing. In addition, we advocate the matching theory framework, in particular, student project allocation (SPA) game, to model the problem and solve it in a distributive manner. The efficient SPA-(S,P) algorithm is implemented to find a stable result for the formulated SPA problem. Matching game is able to some aforementioned limitations of game-theoretic and centralized approaches. There are many benefits to apply the matching game, instead of traditional game theory, to address radio resource allocation problems [14], as it can provide a better model to characterize interactions between different players, define the preferences that can properly present the system requirements, and offers feasible solutions etc [15]. By applying the matching game to the resource allocation problem in IoT fog system, both the cloud provider and the IoT devices are able to express their preference when designing the resource allocation strategy. The considered scenario and proposed scheme can be applied to some typical IoT

applications, such as smart home and Industry 4.0, where fog computing and resource allocation play a significant role. The major contributions of this work are briefly summarized as follows.

- We propose to address a joint radio and computational resource allocation problem for fog computing. We allow users to express their needs, w.r.t. the delay requirement and data size, in the form of mandatory offer to the cloud providers. On the hand, by communicating with the users, cloud providers try to find suitable FNs for offloading users' computation tasks, together with the assignment of radio spectrum, to satisfy users' requirements.
- With the objective of optimizing the user satisfaction, we formulate this joint resource allocation as a mix integer nonlinear programming (MINLP) problem. In formulation, system constraints such as service delay, transmission quality, power control and so on are considered. We advocate the SPA matching game to model the optimization problem, where cloud providers (modeled as lecturers) own the radio/computation resources (modeled as the projects), and are responsible for the communications with users (modeled as students), as shown in Fig. 1(b).
- We adopt the SPA-(S,P) algorithm to find a stable matching result of the SPA game. In addition, the external effect, due to the inter-independence of matching players' preferences lists, is removed by the proposed user-oriented cooperation (UOC) strategy. After the UOC procedure, the network stability is guaranteed and the system performance is further improved.

The rest of this paper is organized as follows. In Section II, some related works in cloud computing and fog computing are discussed. In Section III, we provide the framework and system assumptions of the joint resource allocation problem. Then in Section IV, we formulate the proposed problem as an optimization problem aiming at maximizing the system cost performance. After that, the SPA matching approach is introduced to model the optimization problem, and the SPA-(S,P) algorithm is adopted as a distributed solution in Section V. Simulations results are analyzed in Section VI and conclusions are drawn in Section VII.

II. Related Works

An overview of fog computing and its role in IoT is provided in [5], ranging from conceptual visions to existing point solution prototypes. The opportunities and challenges of fog, focusing primarily on the networking context of IoT, have been discussed in [16]. [17] provided the insight on why the current compute and storage models confined to data centers are not suitable for some of the applications in the IoT scenarios, regarding three requirements: mobility, reliable control and actuation, and scalability. The analysis demonstrates that fog computing is the natural choice for the IoT development considering the large geographical distribution of fog devices and the real-time decision making requirements from users.

Some applications of fog computing framework in IoT have been proposed. For example, [18] explores the social connections of the IoT devices, and develop a relay selection mechanism based on the coalitional game solution to improve the communications among devices. [19] addresses the utility based pairing problem between the IoT devices for resource sharing in the fog computing paradigm. The Irving's stable roommate algorithm is proposed to find a stable matching between the IoT devices. [20] introduced a method for measuring the UV radiance through mobile phone cameras, and the measurements are gathered and amended, by utilizing fog computing, through the fog servers to provide more accurate results. In [21], the authors have identified the requirements in the fog computing application, such as device heterogeneity, support for Perception-Action cycles, mobility and scalability, and proposed a Distributed Dataflow (DDF) programming model. The proposed DDF framework is evaluated by implementing it on a visual programming tool, named Node-RED, that uses a flow-based model for building IoT applications. Some existing works have been proposed, using distributive game-theoretical approaches, to solve the resource allocations in the cloud computing networks. For example, [22] discusses the resource management problem in the fog computing network, which is modeled with a 3-layer architecture: FNs are in the upper layer, data center operators in the middle layer, the users in the bottom layer. A hierarchical Stackelberg game is proposed to find the network equilibrium.

III. System Model

As shown in Fig. 1, we assume a network comprised of a set of IoT devices, such as smart phones, surveillance cameras, vehicles, fire alarm and so on, denoted as the IoT users $\mathcal{U} = \{u_1, u_2, \ldots, u_M\}$. These IoT users may offload certain type of computing or storage tasks to the cloud service providers (SPs), which are denoted as $\mathcal{SP} = \{sp_1, sp_2, \ldots, sp_N\}$. These SPs can meet different users with specific computing requirements w.r.t. data size and service delay. For example, devices like fire alarms are typically more delay sensitive, while devices like freezers are typically more flexible regarding the service latency requirement. For those users who are not delay sensitive, the computing will be sent to the cloud, while for those users with strict delay requirements, the SPs will allocate one of the nearby FNs to offload the computation task. It's not hard to understand that FNs that are closer to the users typically result in smaller transmission latencies. However, the geography location is not the only factor that affects the whole service delay. In fact, the service delay consists of three time periods, which are transmitting time, CPU processing time and receiving time. The transmitting and receiving periods are defined as the time used for sending data to FNs for processing and the time used for receiving the processed results, respectively. Such communication latency is not only related to the channel conditions but also affected by the data size of the computing task. On the other hand, the CPU processing time is decided by the CPU rate of each FN. Thus, for any SP sp_j, when selecting the proper FN from the set $\mathcal{FN}^j = \{fn_1^j, fn_2^j, \ldots, fn_L^j\}$ for each user, it will jointly allocate its radio resources $\mathcal{W}^j = \{w_1^j, w_2^j, \ldots, w_K^j\}$ (channel bandwidth) and computational resources $\mathcal{C}^j = \{c_1^j, c_2^j, \ldots, c_L^j\}$ (CPU cycle rate).

From the users' perspective, who have delay sensitive contents to process, will offer prices to the SPs to compete for better resources (both radio and computational resources). Intuitively, users who are requiring less latencies tend to offer a higher price. In addition, users will take the data sizes into consideration, since typically more data asks for longer transmission period as well as longer CPU processing time. Notice here, the CPU cycles for the processing tasks are related to the data size but not exactly equal to it. Thus, we assume each user u_i carries D_i (bits) data, and the corresponding processing task requires

DC_i CPU cycles. Without loss of generality, we simply assume a linear relation between the DC_i and D_i [12].

So far, we can see the joint radio and computation resource allocation can be treated as the mapping between the user sets $\mathcal{U}$ and the (radio,computation) resource pair sets $\mathcal{RP}^j = \{(w_k^j, c_l^j) | \forall w_k^j \in \mathcal{W}^j, c_l^j \in \mathcal{C}^j\}$ owned by each SP $sp_j, sp_j \in \mathcal{SP}$. In the rest of this work, we may use $rp_{l,k}^j$ to denote (w_k^j, c_l^j) for simplicity. We represent such mapping relation with the binary value $\rho_{k,l}^{i,j}$, where $\rho_{k,l}^{i,j} = 1$ if u_i is offloaded to FN fn_l^j using the channel w_k^j owned by sp_j, and $\rho_{k,l}^{i,j} = 0$ otherwise. In order to optimize the joint resource allocation, we consider the profits of both users' and SPs', which are discussed in the following two sections respectively.

A. User Satisfaction

One of the most important metrics that all SPs concern about is the user experience or user satisfaction. As we mentioned previously, we are discussing a set of users with sensitive delay requirements, so service latency is used as the user satisfaction measurement. However, before talking about the delay, we should first guarantee that the transmission quality between the users and FNs can meet the requirement. In other words, the signal to interference noise ratio (SINR) should be higher than a threshold $\Gamma_{\min}$ in order to deliver the correct/complete data. We define the received SINR from u_i at fn_l^j using w_k^j as follows.

$$\Gamma_{k,l}^{i,j} = \frac{P_i g_{k,l}^{i,j}}{\sum_{u_{i'} \in \mathcal{U}, i' \neq i} \rho_{k,l}^{i',j} P_{i'} h_{k,l}^{i',j} + \sigma_N^2}, \tag{1}$$

where P_i and $g_{k,l}^{i,j}$ are the transmission power and channel gain between user u_i and fog node fn_l^j using channel w_k^j, respectively. $h_{k,l}^{i',j}$ represents the interference channel gain from any other mobile user $u_{i'}$ at fn_l^j due to channel reuse. We consider orthogonal radio resources are used among SPs and the radio resources within SPs can be coordinated to avoid the interference. σ_N^2 represents the channel noise. It is required that $\Gamma_{k,l}^{i,j} \geq \Gamma_{\min}$ in order to ensure a successful transmission.

The transmission rate from u_i at fn_l^j using w_k^j, if satisfying the SINR requirement, can be represented as follows.

$$r_{k,l}^{i,j} = w_k^j \log\left(1 + \Gamma_{k,l}^{i,j}\right), \tag{2}$$

As we discussed previously, the service delay consists of three time periods: the transmitting time t_{trans}, the CPU processing time t_{proc}, and the receiving time t_{recv}. Generally speaking, the received result from the FN after processing is typically in relatively trivial size compared to the original unprocessed data. In addition, with no knowledge of the result after processing, we cannot predict the exact size of the returned data although pretty small. Thus, the receiving time period should be sufficiently short, and we assume a random variable $\delta t, \delta t \in [0, 1]$ to represent t_{recv} for any user. When defining t_{trans} and t_{proc}, we should consider the channel reuse and CPU sharing among multiple users. We allow each channel to be shared among more than one user within it capacity q_R, and also allow each FN to accommodate more than one user to share its CPU within its capacity q_C. We also denote q_{SP} as the maximum number of users that one SP can serve. Thus, the transmission rate for each user can be affected by the interference from the co-channel users, as represented in (1). In addition, the CPU processing rate for each user is affected by the co-CPU users. For simplicity, we assume each co-CPU user will be allocated an equal share of the total CPU rate, denoted as $c_{k,l}^{i,j} = \frac{1}{\sum_{u_i \in \mathcal{U}} \rho_{k,l}^{i,j}} c_l^j$. Now, we can define the service delay of u_i when using the resource pair (w_k^j, c_l^j) as follows.

$$t_{k,l}^{i,j} = t_{\text{trans}} + t_{\text{proc}} + t_{\text{recv}} = \frac{D_i}{r_{k,l}^{i,j}} + \frac{DC_i}{c_{k,l}^{i,j}} + \delta t. \tag{3}$$

B. SP Revenue

The mandatory revenue is the incentive that makes SPs provide better service to its subscribed users. Also as another factor to measure the system performance, we adopt the price offers from the users as the benefit/revenue of the SPs. As we have discussed, the price that each user offers is not only related to its delay requirement T_i but also its data size D_i. Without loss of generality, we assume a linear relation between the price and the data size, as well as the inverse of the delay requirement. Thus, the offer from each user can be represented as follows.

$$O_i = f(D_i, T_i), \tag{4}$$

where $f(\cdot)$ should be a monotonic increasing function for D_i and monotonic decreasing function for T_i. For simplicity, we advocate following function to define $f(D_i, T_i)$.

$$O_i = a\frac{D_i}{T_i}, \tag{5}$$

where a is a parameter with unit dollar/Mbps, and O_i is the price that u_i is willing to pay for any SP if matched.

Each SP serves more than one user, and thus receiving more than one offer. We define each SP sp_j's revenue as the summation of the mandatory offer collected from all its matched users. In this work, we consider the cost of each SP related to power consumption of transmission and maintenance. For the sake of simplicity, we assume it is fixed in this work. When considering the SP's revenue, we ignore the effect of the fixed service cost. As a result, the total revenue for each SP is represented as follows.

$$Rev_j = \sum_{u_i \in \mathcal{U}} \rho_{k,l}^{i,j} O_i. \tag{6}$$

IV. Problem Formulation

In the previous section, we discuss two performance metrics, which are both essential for a good resource allocation in fog computing. The system objective in this work is designed as a combination of both metrics, and is named as the cost-performance (CP). The CP is defined as the ratio between each user's average data rate and its price cost, with the unit of Mbps/sec/dollar. The date rate, instead of pure delay, is considered because that the actual delay value is strongly related to the user's data size to be transmitted and processed. Thus, the actual data rate is a more fair measurement than the delay value if comparing horizontally with other users. Then, for the cost factor, it's also reasonable to use the monetary payment/offer of the user for the corresponding fog computing service it acquires. As a result, to combine both factors in one metric, we have defined the cost-performance function for each user, which physically represents the service quality that the user pays for. The system CP CP_{sys} is the average of all users' CP value $CP(i)$, which is

represented as follows.

$$CP_{sys} = \frac{\sum_{u_i \in \mathcal{U}} CP(i)}{M}, u_i \in \mathcal{U}, \tag{7}$$

where $CP(i)$ is the CP value for user u_i, and is defined as follows.

$$CP(i) = \rho_{k,l}^{i,j} \frac{\frac{D_i}{t_{k,l}^{i,j}}}{O_i}. \tag{8}$$

Next, we are ready to formulate the optimization problem, which is shown below.

$$\max_{\rho_{k,l}^{i,j}} : \quad \frac{\sum_{u_i \in \mathcal{U}} CP(i)}{M} \tag{9}$$

$$\text{s.t.} \quad : \quad \rho_{k,l}^{i,j} t_{k,l}^{i,j} \leq T_i, \quad \forall u_i \in \mathcal{U}, rp_{l,k}^{j} \in \mathcal{RP}^j, sp_j \in \mathcal{SP}, \tag{10}$$

$$\rho_{k,l}^{i,j} \Gamma_{k,l}^{i,j} \geq \Gamma_{min}, \quad \forall u_i \in \mathcal{U}, rp_{l,k}^{j} \in \mathcal{RP}^j, sp_j \in \mathcal{SP}, \tag{11}$$

$$\sum_{u_i \in \mathcal{U}, fn_l^j \in \mathcal{FN}^j} \rho_{k,l}^{i,j} \leq q_R, \forall w_k^j \in \mathcal{W}^j, sp_j \in \mathcal{SP}, \tag{12}$$

$$\sum_{u_i \in \mathcal{U}, w_k^j \in \mathcal{W}^j} \rho_{k,l}^{i,j} \leq q_C, \forall fn_l^j \in \mathcal{FN}^j, sp_j \in \mathcal{SP}, \tag{13}$$

$$\sum_{u_i \in \mathcal{U}, rp_{l,k}^j \in \mathcal{RP}^j} \rho_{k,l}^{i,j} \leq q_{SP}, \forall sp_j \in \mathcal{SP}, \tag{14}$$

$$\rho_{k,l}^{i,j} \in \{0, 1\}, \tag{15}$$

where (9) is the system objective, representing the overall cost performance for users. (10) represents the delay requirement for each user. (11) defines the minimum SINR requirement for each user. (12), (13) and (14) satisfy the capacity constraints for each channel, FN and SP, respectively.

Obviously, this optimization problem is a MINLP problem, which is generally NP-hard to solve [23]. Therefore, it motivates us to find a feasible suboptimal solution. Thus, we introduce the matching-theory based distributed approach: the student project allocation game, which will be discussed in the next section.

V. Student-Project Matching Game

In the previous section we have formulated the joint radio and computational resource allocation as a MINLP problem. Due to the NP-hardness, as well as the new trend of 5G resource management that shifts from the traditional centralized optimization to distributive or semi-distributive approaches, we are proposing a semi-distributive matching-based solution in this section. The fact that assignment of the radio and computation resources are coupled has motivated us to treat the (radio, computation) pair as one individual entity. We can enumerate all possible combinations of the two types of resources and try to map the user sets to the resource pair sets. Apparently, this process of enumerating and mapping should be under the assistance of the SPs, who are responsible for the control signal communication with the users and both resources.

A suitable matching model that exactly offers such structure is the Student Project Allocation (SPA) problem [24], where various students will be assigned different projects (owned by different lecturers) under the assistance of the lecturers. In this section, we first introduce how to model our proposed problem using the SPA model, and then implement the SPA-(S,P) algorithm to find a stable matching solution in Section V-A. However, to deal with the externality that appears during the matching, we propose the inter-channel cooperative strategy to remove the external effect and ensure the system stability in Section V-B.

A. Student-Project Allocation Modeling

In many university departments, students seek to undertake a project (e.g., senior design) from lecturers. Typically each lecturer will offer a variety of projects. Each student has preferences over the available projects that he/she finds acceptable, whilst a lecturer normally have some form of preferences over his/her projects and/or the students who find them acceptable. There may also be upper bounds on the number of students that can be assigned to a particular project, and also the number of students that a given lecturer is willing to supervise. One variant is the SPA problem with lecturer preferences over student-project pairs, referred to as SPA-(S,P), in which each lecturer has a preference list that depends on not only the students who find his/her projects acceptable, but also the particular projects that these students would undertake [25].

Inspired by the SPA problem, we model the resource allocation problem in fog computing as the SPA game, in which we assume the SPs, the (radio, computation) resource pairs and the users as the lecturers, the projects and the students, respectively. In the SPA model, lecturers offer different projects, and students can apply for these projects. Similarly in our work, SPs offer available radio and CPU resource bundles, and users propose to the SPs for acceptable resource bundles. SPs make decisions based on the revenue that can be collected from the users by offering the resource bundles. The stability notion here implies the robustness to deviations that can benefit both the users and the resources. An unstable matching can lead to cases in which two SPs can swap their matched users if this swap is beneficial to both of them. Having such network-wide deviations ultimately leads to an undesirable and unstable network operation. The formal stability definition is provided in Definition 1.

Definition 1. Stability: A matching $\mathcal{M}$ is said be stable, if there's no blocking pair (BP). A pair $(u_i, rp_{l,k}^j)$ is defined as a BP if all of the following conditions are satisfied:

(1) u_i finds $rp_{l,k}^j$ acceptable;

(2) either u_i is unmatched in $\mathcal{M}$, or u_i prefers $rp_{l,k}^j$ to $\mathcal{M}(u_i)$;

(3) either

(3.1) $rp_{l,k}^j$ is under subscribed and either of the following three conditions is satisfied:

- *a)* $\mathcal{M}(u_i) \in \mathcal{RP}^j$, and sp_j prefers $(u_i, rp_{l,k}^j)$ to $(u_i, \mathcal{M}(u_i))$; or
- *b)* $\mathcal{M}(u_i) \notin \mathcal{RP}^j$ and sp_j is under-subscribed; or
- *c)* $\mathcal{M}(u_i) \notin \mathcal{RP}^j$ and sp_j is full and sp_j prefers $(u_i, rp_{l,k}^j)$ to its current worst pair $(u_{\text{wst}}, rp_{\text{wst}}^j)$;

(3.2) $rp^j_{l,k}$ is full and sp_j prefers $(u_i, rp^j_{l,k})$ to the its current worst pair $(u_{\text{wst}}, rp^j_{\text{wst}})$, and either of the following two conditions is satisfied:

a) $\mathcal{M}(u_i) \notin \mathcal{RP}^j$;

b) $\mathcal{M}(u_i) \in \mathcal{RP}^j$ and sp_j prefers $(u_i, rp^j_{l,k})$ to $(u_i, \mathcal{M}(u_i))$.

In Definition 1, $\mathcal{M}(x)$ represents the partner/matching of the player x in matching $\mathcal{M}$. More precisely, $\mathcal{M}(u_i) = rp^j_{l,k}, (w^j_k, c^j_l) \in \mathcal{RP}^j$.

In order to find a stable matching, the preference lists of both users' and SPs', denoted as $\mathcal{PL}^{user}$ and $\mathcal{PL}^{SP}$, need to be established first. During this procedure, the constraints (10) and (11) should be satisfied from both users' perspective, w.r.t. delay and SINR requirement. In other words, when setting up the preference lists for users, each user needs to first check the two constraints, and include those resource pairs that satisfy them. These sets of resource pairs are called the acceptable sets. After finding all users' acceptable sets, we rank these resource pairs in descending/ascending orders for each user according to their preferences. Intuitively, users prefer resources that can offer the computation offloading with the least delay. However, since we allow each resource pair to accommodate more than one user, then the multi-user coexistence will affect both the radio and the CPU performances. For simplification, we assume these coexisting users share the frequency band as well as the CPU rate equally. Thus, it is not who the user will share resources with that matters but how many of them. Before the matching is finalized, this number is unknown to any user nor SP, although each SP and each radio and CPU resource do have quotas, q^{SP} q^R and q^C, that limit the maximum number of users. In order to calculate the potential service delay, each user will assume a $\frac{1}{Q}$ share of the radio and CPU resource depending on the exact quota Q. The true performance actually may deviate from this evaluation, which causes the external effect during the matching (We'll address this issue in the Section V-B). Thus, the preference of any user u_i over the $rp^j_{l,k}$ is based on the potential service delay $t'^{i,j}_{k,l}$, and is represented as follows.

$$PL^{user}_i(j,k,l) = t'^{i,j}_{k,l} = t'_{\text{trans}} + t'_{\text{proc}} + t'_{\text{recv}} = \frac{D_i}{\frac{1}{q^R} r'^{i,j}_{k,l}} + \frac{DC_i}{\frac{1}{q^R} c^j_l} + \delta t', \tag{16}$$

where $r'^{i,j}_{k,l}$ the data rate from u_i to FN fn^j_l when only u_i is using the channel w^j_k, and is represented as $r'^{i,j}_{k,l} = w^j_k \log(1 + \frac{P_i g^{i,j}_{k,l}}{\sigma^2_N})$. $\delta t'$ is an another random value within $[0, 1]$ that represents the possible time period for returning the result.

On the other hand, when selecting the users to match its resource pairs, SPs not only consider the mandatory benefit that is related to the data size, but also the potential service delay. The delay factor works likewise for users and SPs, since users expect faster service and SPs pursuit short service times in each user so that to serve more users in the long term consideration. Thus, the preferences of SPs over the users are based on the ratio of price over delay (same as the potential delay evaluation for users), and is represented as follows.

$$PL^{SP}_{j,k,l}(i) = \frac{O_i}{t'^{i,j}_{k,l}}. \tag{17}$$

Algorithm 1: SPA-(S,P) Algorithm.

Input: $\mathcal{U}, \mathcal{SP}, \mathcal{W}, \mathcal{FN}, \mathcal{PL}^{user}, \mathcal{PL}^{SP}$;
Output: Matching $\mathcal{M}$;
Initialization: set $\mathcal{M}$ empty, set all users free;
while some user u_i is free and u_i has a non-empty preference list **do**
  for all $u_i \in \mathcal{U}$ **do**
    u_i proposes to the first entity $rp^j_{l,k}$ in $\mathcal{PL}^{user}_i$, and then remove $rp^j_{l,k}$ from $\mathcal{PL}^{user}_i$;
    $\mathcal{M} \leftarrow \mathcal{M} \cup (u_i, rp^j_{l,k})$;
  end for
  for all $rp^j_{l,k}, rp^j_{l,k} \in \mathcal{RP}^j, sp_j \in \mathcal{SP}$ **do**
    while $rp^j_{l,k}$ is over-subscribed **do**
      Find the worst pair $(u_{\text{wst}}, rp_{\text{wst}})$ assigned to $rp^j_{l,k}$ in sp_j's list;
      $\mathcal{M} \leftarrow \mathcal{M}/(u_{\text{wst}}, rp_{\text{wst}})$;
    end while
  end for
  for all $sp_j \in \mathcal{SP}$ **do**
    while sp_j is over-subscribed **do**
      Find the worst pair $(u_{\text{wst}}, rp_{\text{wst}})$ in sp_j's list;
      $\mathcal{M} \leftarrow \mathcal{M}/(u_{\text{wst}}, rp_{\text{wst}})$;
    end while
  end for
end while
Terminate with a matching $\mathcal{M}$.

With the preference lists set up, we can apply the SPA-(S,P) algorithm, as illustrated in Algorithm 1, to find an efficient matching between users and resources. The key idea of the SPA-(S,P) algorithm is developed from the classical Gale-Shapley algorithm [26]. It consists of sequential proposing and accepting/rejecting operations by users and SPs. The convergence of the SPA-(S,P) algorithm is guaranteed and the existence of a stable matching is proven in [24]. As it is Gale-Shapley algorithm-based, the overall computation complexity is $\mathcal{O}(m)$ where m is the total length of the preferences lists.

It can be noticed that a stable matching is ensured under the condition of Canonical matching. It implies that the preference of any players don't depend on the choices/actions of other players, but on the local information about the other type of players. While this assumption is no longer true in this work, since with more users sharing the same radio/CPU resources, their performances will be degraded. Thus, the resulting matching after running the SPA-(S,P) algorithm is not necessarily stable, and calls for further actions to reach stability. In the next subsection, we propose a cooperative procedure to transform the current matching into being stable again.

B. User-Oriented Cooperation Strategy

Due to the inter-dependence of the preferences of users and resources (i.e., they are influenced by the existing matching), the matching yielded by SPA-(S,P) algorithm is not necessarily stable. We call the matching framework with such inter-dependence as matching games with externality [27]. For example, a previously good resource pair may be over evaluated with many users sharing it, while a not so good one may become better

Algorithm 2: User-Oriented Cooperation (UOC) Strategy.

Input: Existing matching $\mathcal{M}_0$;
Output: Pareto optimal matching $\mathcal{M}_s$.
$\mathcal{M}_t = \mathcal{M}_0$;
while $\mathcal{M}_t$ is "unstable" (user, user) pairs $\mathcal{BP}$ **do**
  for all $(u_{i1}, u_{i2}) \in \mathcal{BP}$ **do**
    if $\exists u \in \mathcal{M}_t(rp_{i1}) \cup \mathcal{M}_t(rp_{i2})$, $\Delta U(u) < 0$ **then**
      (u_{i1}, u_{i2}) are not allowed to switch partners;
    else
      (u_{i1}, u_{i2}) are allowed to switch partners;
    end if
  end for
  Find the optimal BP $(u^*_{i1}, u^*_{i2}) \in \mathcal{BP}$;
  u^*_{i1} and u^*_{i2} switch partners;
  $\mathcal{M}_{t+1} \leftarrow \mathcal{M}_t / \{(u^*_{i1}, \mathcal{M}_t(u^*_{i1})), (u^*_{i2}, \mathcal{M}_t(u^*_{i2}))\}$;
  $\mathcal{M}_{t+1} \leftarrow \mathcal{M}_t \cup \{(u^*_{i1}, \mathcal{M}_t(u^*_{i2})), (u^*_{i2}, \mathcal{M}_t(u^*_{i1}))\}$;
  Update $\mathcal{PL}^{user}$ based on $\mathcal{M}_t$;
end while
$\mathcal{M}_s = \mathcal{M}_t$.

with very few users sharing it. There may be incentives for users to swap to other resources, which become the BPs in the matching. We can design algorithms to remove those BPs; however, it is also reasonable to think more from the users' point of view. With our system objective evaluated through the average users' cost performance, we believe that it's workable to begin to value the stability notion solely from the user side at this time point. In other words, we assume that only users have the incentive to make changes. Thus, a new "stability" notation should be defined among the users. This new "stability", different from Definition 1, relies on the equilibrium among all users. Cooperation between users are needed to transform the existing matching into a stable one. We call such one-sided "stability" as "Pareto Optimality" in matching theory [25]. The definition of Pareto optimal is provided as follows.

Definition 2: Pareto Optimal: A matching is said to be Pareto Optimal if there is no other matching in which some player is better off, whilst no player is worse off.

Accordingly, the new BP definition for the one-sided matching problem in given in Definition 3.

Definition 3: A BP in the one-sided matching: A user pair (u_i, u_j) is defined as a BP, if both u_i and u_j are better off after exchanging their parters.

To find such Pareto optimal matching, users again requires assistance from the SPs for utility evaluation. The stability/Pareto optimality is achieved through finite partner switch operations between user pairs. As stated in Definition 2, the stability/Pareto optimality is reached when no player/user is better off without other player(s) being worse off. In other words, every swap operation should be beneficial to some user(s) while being no harm to the rest users. Through finite such swaps, we can finally reach a swap-free system, which means a stable system. We call such procedure as the User Cooperation (UOC) Strategy, and the details are illustrated in Algorithm 2.

In Algorithm 2, $rp_{i1} = \mathcal{M}_t(u_{i1}), rp_{i2} = \mathcal{M}_t(u_{i2})$. We define $U(x)$ as the utility function of user x, and is equal to its service delay. Thus, we define $\Delta U(x) = U(x)' - U(x)$, where $U(x)'$ is the utility after exchanging partners. In other words, $\Delta U(x)$ represents user x's performance change, and is improved if $\Delta U(x) > 0$ or decreased if $\Delta U(x) < 0$. A user pair is allowed to switch partners if and only if $\Delta U(x) \geq 0$ for any user x that is affected in this switch (e.g., $\forall x \in \mathcal{M}_t(rp_{i1}) \cup \mathcal{M}_t(rp_{i2})$). Then to find the optimal BP among all the BPs allowed to switch partners, we search for a BP which provides the highest the performance improvement. The performance here refers to the overal time delay for all users. We define the optimal BP as follows.

$$(u^*_{i1}, u^*_{i2}) = \underset{(u_{i1}, u_{i2})}{\text{argmax}} \sum_{u \in \{u_{i1} \cup u_{i2} \cup \mathcal{M}_t(rp_{i1}) \cup \mathcal{M}_t(rp_{i2})\}} \Delta U(u), \tag{18}$$

where the user pair (u_{i1}, u_{i2}) should be allowed to exchange partners.

We summarize the steps of the UOC strategy as follows: firstly search all "unstable" user-user pairs (who have the exchange incentive) regarding the current matching; secondly, check whether the exchange/switch between such a pair is allowed (beneficial to related users); thirdly, find the allowed pair, that provides the greatest throughput improvement, to switch their partners, and update the current matching; then keep searching such BPs until we reach a trade-in-free network. The convergence of the UOC process is guaranteed by the irreversibility of each switch. Finally, UOC terminates with a Pareto optimal matching, and simultaneously improves the system throughput. The total iterations of BP searches or swaps are bounded by N^2. Thus, the worst case complexity of terminating the algorithm is $\mathcal{O}(N^3 M)$.

VI. Performance Evaluation

In this Section, we first evaluate both SPA-(S,P) algorithm and the UOC strategy w.r.t. users' service latency, SPs' profit and the system cost performance. In addition, the convergence of UOC will be analyzed.

We consider a network with $N = 2$ SPs, each equipped with $L = 5$ FNs randomly distributed within the network, with a radius of $R = 1$ km. Assume a number of IoT users $M, M \in [45, 210]$, also randomly distributed within the network. Each SP owns $K = 5$ channel bands for users to share, and the bandwidth is set to $w = 5$ MHz. The SINR requirement Γ_{min} for users is a uniform random distribution within $[20, 30]$ dB. We set equal capacity requirement for each channel and each FN, which is $q_R = q_C = 10$, and the SP's capacity is set as $q_{SP} = 80$ for each. Users' delay requirement and data size, as well as the corresponding CPU cycles, are determined by the specific IoT device types. The service delay includes both transmission latency and CPU processing latency, and total delay requirement T_d for each user is uniformly distributed within $[6, 7]$ sec. The users' data size D is set as a uniform distribution, $[2, 8]$ Mb, and corresponding CPU cycles is set as $DC_i = D_i * 10^4$ cycles. The CPU processing rate for each FN is set as a uniform distribution within $[5, 6] * 10^{10}$ cycles/sec. For the propagation gain g, we set the pass loss constant C as 10^{-2}, the path loss exponent α as 4, the multipath fading gain as the exponential distribution with unit mean, and the shadowing gain as the log-normal distribution with 4 dB deviation.

In Figs. 2 and 3, we evaluate the performance of users and SPs. For comparison purposes, we use the Random method as the victim strategy, which refers to a random resource allocation between users and resource pairs. In addition, we also modify the one proposed in [9], which consider a joint optimization

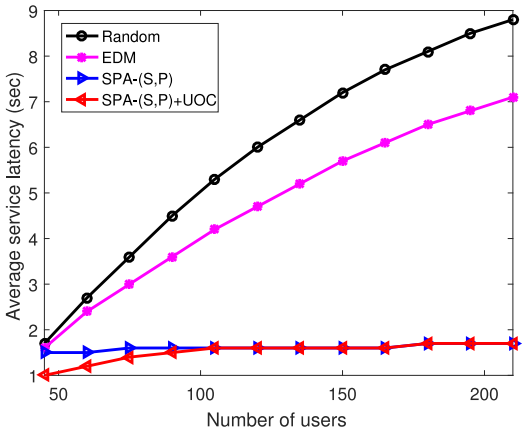


Fig. 2. Users' average service latency.

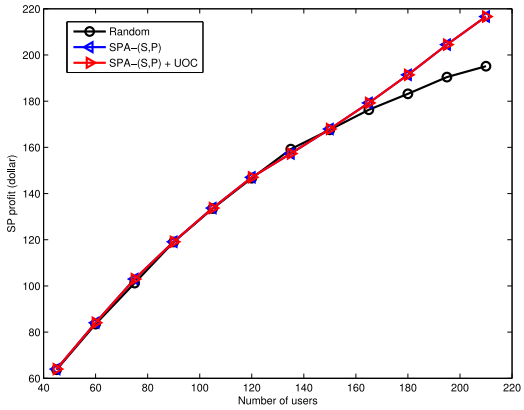


Fig. 3. SPs' profit.

of energy consumption and delay performance (EDM). Fig. 2 shows the average service delay evaluation under the comparison of four methods: the Random method, the EDM algorithm, the SPA-(S,P) algorithm and the SPA-(S,P) with the UOC strategy. We increase the number of users from 45 to 210 by the step of 15 to show the change of latencies. Apparently, the service latency for all four strategies increase with the number of users. It is understandable since more users means less resource share for each averagely, which thus leading to higher delay. Among the four methods, the Random curve gives the highest average latency, and is much higher than the others. The EDM also has a worse latency performance comparing with the proposed ones. For the rest two matching curves, SPA with UOC is slightly better than the SPA-(S,P) when the user number $M < 150$, and is almost the same as SPA-(S,P) when $M > 150$. It tells two things, one is that UOC can further improve users' performance while guarantee network stability, and second thing is that the improvement is less apparent when the user number is close to or has reached the network capacity $M = 160$. The network capacity refers to the maximum number of users that the SPs can accommodate without any user left unmatched. The reason that UOC can further improve users' performance is the user switching rules designed for UOC. Only when a switch is beneficial to both of the users and does no harm to the performance of the rest users can this swap be allowed. From the SPs' perspective, it gains mandatory profit from matched users. As shown in Fig. 3, the average profit gained by SPs are almost the same for all three methods when $M < 150$, and after $M > 150$ both SPA-(S,P) and SPA-(S,P) with UOC outperform the Random method. In

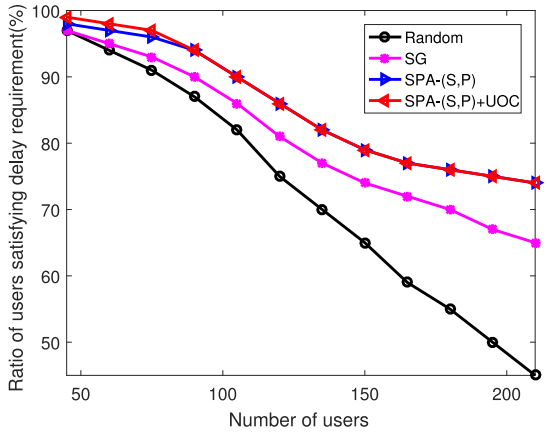


Fig. 4. Ratio of users satisfying delay requirement.

fact, the SPs' benefit is decided by the number of users who get matched as well as who are matched and who are not. Before the user number reaches the network capacity, almost all the users can be matched to a resource pair under different methods, good or not. Thus, SPs can still gain all the money. However, when the users are more than the network capacity, then users need to compete for a share. Thus, which users are kicked off and which ones stay? As we discussed in Section III, users who have more strict latency requirement offer higher prices, thus making them more likely to be selected by the SPs. In turn, users with higher offers make the SPs gain more profit. That's why both matching curves beat the the other curves when $M > 150$.

The user satisfaction is evaluated in Fig. 4, w.r.t. the ratio of users whose actual service latencies meet their requirements. We have modified the one in [22] and applied the Stackelberg game to model the interactions between SPs and users. Apparently, the ratio of satisfied users decrease with the increase of users for all four methods. The starting points of all four methods are almost 100%, and after that the Random method drops faster than the other three algorithms. The two matching curves decrease in similar speeds, and the decrease become slower after the user number $M > 150$. At the end point when $M = 210$, both the SPA-(S,P) with UOC method and SPA-(S,P) method reach almost 75%, while the Random curve falls below 50%. In other words, more than 75% of users are satisfied with their performances with the allocated fog and radio resources under the proposed matching methods when $M < 150$. Fig. 4, together with the average delay evaluation shown in Fig. 2, shows that our proposed matching algorithms not only think from the users' and SPs' point of view in an average way, but also takes each individual user's performance into consideration.

In Fig. 5, we evaluate the system cost performance under three methods. Intuitively, the cost performance is a measurement of how much service one can buy at what price. It's an joint consideration of both users' and SPs' benefits, with the objective to allocate the best resources to users who want them most (i.e., who offer the highest prices). As shown in Fig. 5, when $M < 75$, SPA-(S,P) with UOC outperforms the other two, while the Random allocation beats the SPA-(S,P). This happens because in SPA-(S,P), users first propose to their favorite resources, and thus some good resources may receive many more proposals than the rest resources. Thus, when user number is relatively small and there are sufficient spare resources, the good resources, who are matched with users to their full capacities, may be not so good as those resource who have sufficient spare

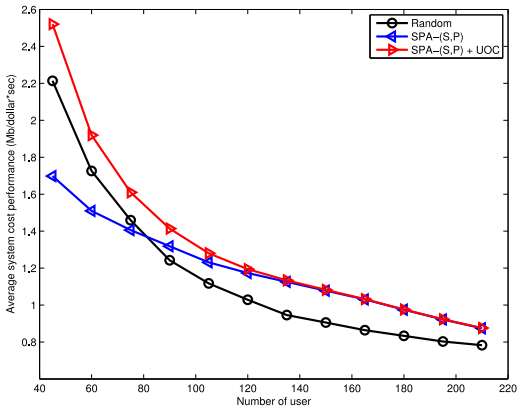


Fig. 5. System cost performance.

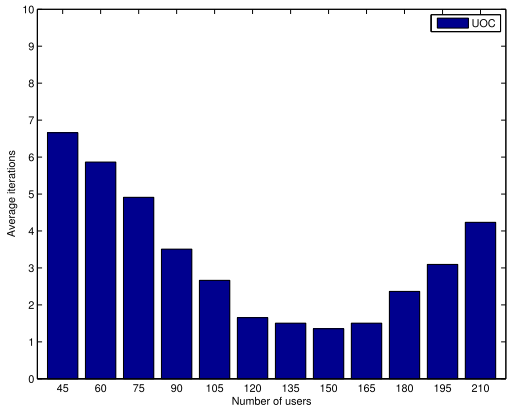


Fig. 6. UOC convergence analysis.

rooms. On the other hand, the Random allocation method is designed as a uniform random allocation in our simulation, which allocates users more distributively than the SPA-(S,P). Thus, when user number is small, the SPA-(S,P) is worse than the Random. After $M > 80$, both matching algorithms are better than the Random one. SPA-(S,P) with UOC outperforms SPA-(S,P) when $M < 150$. The performance of all three curves are decreasing with the increase of M. It is reasonable since the offers keep unchanged, but more users lead to less resource share for each, thus making the average cost performance decrease.

Lastly, the convergence of the proposed UOC strategy is analyzed in Fig. 6. The iteration of users swaps/switches during UOC is taken as the measurement of its convergence, which is calculated under averaging 200 times of simulation. As we discussed in Section V, the convergence of UOC to a Pareto optimal matching is guaranteed since each switch is not revertible. By each switch, some users can switch to currently better resource pairs, which are preciously under evaluated. With so many switch options, our proposed UOC selects the currently best user pair to switch. It's not hard to understand, such pair selection procedure can greatly reduce the number of switching times if switch under no pre-selection. We can see a decrease of iterations with the increase of user number M when $M < 150$, and then begin to increase with the increase of user number when $M > 150$. Notice here, the network capacity is $M = 160$ and the user increase step is 15, which means when $M > 150$ the user number exceeds the network capacity. So before the user number exceeds the capacity, SPs have some resource pairs who have spare rooms for more users. Thus with more such rooms, users have more chances to improve their performance by switching. Thus, it explains that with the decrease of spare network capacity (i.e., when $M < 160$), the switching times decrease. However after the user number has reached the network capacity, there are more users who can not get any resources. Thus the competition between these unmatched users and the matched users will bring more switches. Thus, after $M > 160$, with more unmatched users in the network, the switch times start to increase. No matter decreasing or increasing, the total swapping time is limited by 10 in this network setting, which is in fact a trivial number.

VII. Conclusion

In this work, we have studied the joint radio and computational resource allocation problem in fog computing. Considering the distributive features of the IoT framework, we have proposed matching theory, as a semi-distributive solution approach, to find a stable matching between the users and resources. With the proposed SPA framework, we have modeled the interaction between the IoT users, SPs and FNs. System requirements, such as the transmission quality, service latency, and maximum power requirement have been addressed through the representation of the preference lists. The proposed SPA-(S,P) algorithm together with the UOC procedure can guarantee a stable matching. The simulations results have demonstrated that our proposed framework can provide distributive, close-to-optimal performance from both the users' perspective and the system's view.

References

[1] M. Barcelo, A. Correa, J. Llorca, A. Tulino, J. L. Vicario, and A. Morell, "IoT-cloud service optimization in next generation smart environments," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 12, pp. 4077–4090, Dec. 2016.

[2] Y. Sun, R. Bie, P. Thomas, and X. Cheng, "New advances in data, information, and knowledge in the Internet of Things," *Pers. Ubiquitous Comput.*, vol. 20, no. 5, pp. 653–655, 2016.

[3] "Cloud Computing," [Online]. Available: https://en.wikipedia.org/wiki/Cloud_computing

[4] N. Fernando, S. W. Loke, and W. Rahayu, "Mobile cloud computing: A survey," *Future Gen. Comput. Syst.*, vol. 29, no. 1, pp. 84–106, Jan. 2013.

[5] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the internet of things," in *Proc. 1st ed. MCC Workshop Mobile Cloud Comput.*, Helsinki, Finland, Aug. 2012, pp. 13–16.

[6] "Fog Computing and the Internet of Things: Extend the Cloud to Where the Things Are," 2015, white paper. [Online]. Available: https://www.cisco.com/c/dam/en_us/solutions/trends/iot/docs/computing-overview.pdf

[7] X. Chen, L. Jiao, W. Li, and X. Fu, "Efficient multi-user computation offloading for mobile-edge cloud computing," *IEEE/ACM Trans. Netw.*, vol. 24, no. 5, pp. 2795–2808, Oct. 2016.

[8] D. Huang, P. Wang, and D. Niyato, "A dynamic offloading algorithm for mobile computing," *IEEE Trans. Wireless Commun.*, vol. 11, no. 6, pp. 1991–1995, Jun. 2012.

[9] L. Liu, Z. Chang, X. Guo, S. Mao, and T. Ristaniemi, "Multi-objective optimization for computation offloading in fog computing," *IEEE Internet Things J.*, vol. 5, no. 1, pp. 283–294, Feb. 2018.

[10] C. You, K. Huang, and H. Chae, "Energy efficient mobile cloud computing powered by wireless energy transfer," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 5, pp. 1757–1771, May 2016.

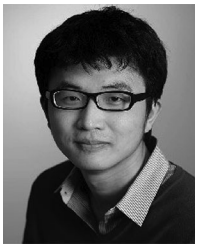
[11] Z. Han, D. Niyato, W. Saad, T. Basar, and A. Hjorungnes, *Game Theory in Wireless and Communication Networks: Theory, Models and Applications*. Cambridge, U.K.: Cambridge Univ. Press, 2011.

[12] G. Scutari, S. Sardellitti, and S. Barbarossa, "Joint optimization of radio and computational resources for multicell mobile-edge computing," *IEEE Trans. Signal Inf. Process. Over Netw.*, vol. 1, no. 2, pp. 89–103, Jun. 2015.

[13] Y. Gu, W. Saad, M. Bennis, M. Debbah, and Z. Han, "Matching theory for future wireless networks: Fundamentals and applications," *IEEE Commun. Mag.*, vol. 53, no. 5, pp. 52–59, Apr. 2015.

[14] Y. Gu, "Matching theory framework for 5G wireless communications," Ph.D. dessertation, Univ. Houston, Houston, TX, USA, Dec. 2016.
[15] Z. Han, Y. Gu, and W. Saad, *Matching Theory for Next-Generation Wireless Communications*. New York, NY, USA: Springer.
[16] M. Chiang and T. Zhang, "Fog and IoT: An overview of research opportunities," *IEEE Internet Things J.*, vol. 3, no. 6, pp. 854–864, Dec. 2016.
[17] M. Yannuzzi, R. Milito, R. Serral-Gracia, D. Montero, and M. Nemirovsky, "Key ingredients in an IoT recipe: Fog computing, cloud computing, and more fog computing," in *Proc. IEEE 19th Int. Workshop Comput. Aided Modeling Des. Commun. Links Netw.*, Athens, Greek, Dec. 2014, pp. 325–329.
[18] X. Chen, B. Proulx, X. Gong, and J. Zhang, "Exploiting social ties for cooperative D2D communications: A mobile social networking case," *IEEE/ACM Trans. Netw.*, vol. 23, no. 5, pp. 1471–1484, Oct. 2015.
[19] S. F. Abedin, M. G. R. Alam, N. H. Tran, and C. S. Hong, "A fog based system model for cooperative IoT node pairing using matching theory," in *Proc. 17th Asia-Pacific Netw. Operations Manag. Symp.*, Busan, Korea, Aug. 2015, pp. 309–314.
[20] B. Mei, W. Cheng, and X. Cheng, "Fog computing based ultraviolet radiation measurement via smartphones," in *Proc. 3rd IEEE Workshop Hot Topics Web Syst. Technol.*, Washington D.C., USA, Nov. 2015, pp. 79–84.
[21] N. K. Giang, M. Blackstock, R. Lea, and V. C. M. Leung, "Developing IoT applications in the fog: A distributed dataflow approach," *5th Int. Conf. Internet Things*, Seoul, Korea, Oct. 2015, pp. 155–162.
[22] 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 Things J.*, vol. 4, no. 5, pp. 1204–1215, Oct. 2017.
[23] D. Bertsimas and J. N. Tsitsiklis, *Introduction to Linear Optimization*. Be[a]lmont, MA, USA: Athena Sci., 1997.
[24] A. H. A. El-Atta and M. I. Moussa, "Student project allocation with preference lists over (student, project) pairs," in *Proc. 2nd Int. Conf. Comput. Electric. Eng.*, Dubai, Dec. 2009, pp. 375–379.
[25] D. F. Manlove, *Algorithmics of Matching Under Preferences*. Singapore: World Sci., 2013.
[26] D. Gale and L. S. Shapley, "College admissions and the stability of marriage," *Amer. Math. Monthly*, vol. 69, no. 1, pp. 9–15, Jan. 1962.
[27] A. Roth and M. A. O. Sotomayor, *Two-Sided Matching: A Study in Game-Theoretic Modeling and Analysis*. Cambridge, U.K.: Cambridge Univ. Press, 1992.

Yunan Gu received the B.S. degree in electronic engineering from Nanjing University of Science and Technology, Nanjing, China, in June 2011, the M.S. degree in computer science from Texas Southern University, Houston, TX, USA, in 2013, and the Ph.D. degree from the University of Houston, Houston, TX, USA, in 2016. She is currently with Huawei, Beijing, China. Her research interest include wireless communications and matching theory.

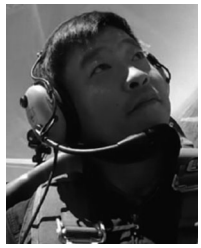


Zheng Chang (S'10–M'13–SM'17) received the B.Eng. degree from Jilin University, Changchun, China, in 2007, the M.Sc. (Tech.) degree from Helsinki University of Technology (Now Aalto University), Espoo, Finland, in 2009, and the Ph.D. degree from the University of Jyväskylä, Jyväskylä, Finland, in 2013. Since 2008, he has held various research positions with Helsinki University of Technology, University of Jyväskylä, and Magister Solutions Ltd. in Finland. He was a visiting Researcher with Tsinghua University, Beijing, China, from June to August in 2013, and with the University of Houston, Houston, TX, USA, from April to May in 2015. He is currently an Assistant Professor with the University of Jyväskylä. His research interests include IoT, cloud/edge computing, security and privacy, vehicular networks, and green communications. He has been awarded by the Ulla Tuominen Foundation, the Nokia Foundation, and the Riitta and Jorma J. Takanen Foundation for his research excellence.

He serves as an Editor of IEEE ACCESS, *Springer Wireless Networks*, and *IEEE MMTC Communications Frontier*, and Guest Editor for IEEE ACCESS, IEEE COMMUNICATIONS MAGAZINE, IEEE INTERNET OF THINGS JOURNALS, and *Wireless Communications and Mobile Computing*. He also served as TPC member for many IEEE major conferences. He has received Best Conference Paper Awards from IEEE Technical Committee on Green Communications and Computing and 23rd Asia-Pacific Conference on Communications in 2017.

Miao Pan (S'07–M'12) received the B.Sc. degree in electrical engineering from Dalian University of Technology, Dalian, China, in 2004, the MA.Sc. degree in electrical and computer engineering from Beijing University of Posts and Telecommunications, Beijing, China, in 2007, and the Ph.D. degree in electrical and computer engineering from the University of Florida, Gainesville, FL, USA, in 2012, respectively. He is currently an Assistant Professor with the Department of Electrical and Computer Engineering, University of Houston, Houston, TX, USA. He was an Assistant Professor in computer science with Texas Southern University from 2012 to 2015. His research interests include cognitive radio networks, cyber-physical systems, and cybersecurity. His work won Best Paper Awards in Globecom 2017 and Globecom 2015, respectively. He is currently an Associate Editor for IEEE INTERNET OF THINGS.

Lingyang Song (SM'11) received the Ph.D. degree from the University of York, York, U.K., in 2007. He was a Research Fellow with the University of Oslo, Oslo, Norway, until rejoining Philips Research U.K. in March 2008. In May 2009, he joined the School of Electronics Engineering and Computer Science, Peking University, Beijing, China, as a full Professor. His main research interests include MIMO, cognitive and cooperative communications, security, and big data. He wrote two text books, *Wireless Device-to-Device Communications and Networks* (Cambridge University Press, 2015) and *Full-Duplex Communications and Networks* (Cambridge University Press, 2017). He received the K.M. Stott Prize for excellent research from the University of York. He is the recipient of IEEE Leonard G. Abraham Prize in 2016 and IEEE Asia Pacific Young Researcher Award in 2012. He is currently on the Editorial Board of IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS. He has been an IEEE Distinguished Lecturer since 2015.



Zhu Han (S'01–M'04–SM'09–F'14) received the B.S. degree in electronic engineering from Tsinghua University, Beijing, China, in 1997, and the M.S. and Ph.D. degrees in electrical and computer engineering from the University of Maryland, College Park, MD, USA, in 1999 and 2003, respectively.

From 2000 to 2002, he was an R&D Engineer of JDSU, Germantown, MD, USA. From 2003 to 2006, he was a Research Associate with the University of Maryland. From 2006 to 2008, he was an Assistant Professor with Boise State University, Boise, ID, USA. He is currently a Professor with the Electrical and Computer Engineering Department as well as with the Computer Science Department, University of Houston, Houston, TX, USA. His research interests include wireless resource allocation and management, wireless communications and networking, game theory, big data analysis, security, and smart grid. He received an NSF Career Award in 2010, the Fred W. Ellersick Prize of the IEEE Communication Society in 2011, the EURASIP Best Paper Award for the Journal on Advances in Signal Processing in 2015, IEEE Leonard G. Abraham Prize in the field of Communications Systems (Best Paper Award in IEEE JSAC) in 2016, and several best paper awards in IEEE conferences. He is currently an IEEE Communications Society Distinguished Lecturer.