



A fog-driven dynamic resource allocation technique in ultra dense femtocell networks



Mohammad Goudarzi^a, Marimuthu Palaniswami^b, Rajkumar Buyya^{a,*}

^a The Cloud Computing and Distributed Systems (CLOUDS) Laboratory, School of Computing and Information Systems, The University of Melbourne, Australia

^b Department of Electrical and Electronic Engineering, The University of Melbourne, Australia

ARTICLE INFO

Keywords:

Femtocell networks
Distributed dynamic clustering
Fog servers
Policy aware resource allocation
3GPP long term evolution (LTE)

ABSTRACT

The highly dense small cell structure filled with large number of Femtocell Base Stations (FBSs) is expected to address the increasing data demand of end users in current and upcoming generation of wireless networks. However, the large and random deployment of such devices incur severe interference which leads to significant performance degradation. To overcome this issue in the Orthogonal Frequency Division Multiple Access (OFDMA)-based femtocell networks, we propose a hierarchical technique consisting of a dynamic distributed clustering and a fog-driven resource allocation to optimize the total throughput of the network while mitigating the interference. Our fully distributed clustering method is designed so that FBSs adaptively form clusters with dynamic size based on the current status of the network and end users. Moreover, we put forward a policy-aware resource allocation method to address the intra and inter-cluster interference, which are two potential types of interference in clustering-based resource allocation techniques. Since our technique carefully considers users' demands in cluster formation, there is always sufficient resources for end users in each cluster, so that each cluster head can find a resource allocation solution, by which no intra-cluster interference occurs. Besides, we employ local fog servers situated in the proximity of clusters for monitoring and assigning a set of policies to CHs for resource allocation, by which the number of inter-cluster interference can be significantly reduced. The extensive simulation results demonstrate that our proposed hierarchical technique significantly improves total throughput, interference, user satisfaction, and fairness compared to other proposed techniques in dense and ultra-dense femtocell networks.

1. Introduction

A rapid growth in deployment and the use of mobile devices such as smartphones, tablets, and sensors has resulted in rapid increase of data-streaming applications such as video streaming, online games, health-care, and Voice over Internet Protocol (VoIP). This leads to a significant amount of data to be transferred over cellular networks (Lee et al., 2014; Goudarzi et al., 2016, 2017). Considering the fact that the number of cellular network resources is restricted, the requested quality of service can be satisfied for only a limited number of users. Besides, recent studies have revealed that approximately 70 percent of the data is originated from indoor places where severe wall penetration loss and longer transmission distance incur poor received signal quality (Garcia-Morales et al., 2015). To address these issues, Femtocell Base Stations (FBSs), which are low-power, short-range, and low-cost edge devices are deployed over macrocell network to effectively improve the indoor

received signal quality and overall network throughput. This latter is obtained by reusing same frequency by several FBSs while the former one is satisfied by decreasing the distance between transmitter and end users (Fu et al., 2017).

However, in densely deployed femtocell networks, neighboring FBSs experience severe co-tier interference (i.e., interference between adjacent femtocells (Mhiri et al., 2013)) due to finite domain of shared spectrum unless an efficient interference management technique is used. The co-tier interference can be significantly reduced in down-link Orthogonal Frequency Division Multiple Access (OFDMA)-based femtocell networks by means of an efficient allocation of Resource Blocks (RBs) between interfering FBSs (Bu et al., 2015). To achieve this, researchers have proposed several Resource Allocation (RA) techniques including centralized and clustering. However, due to non-convex non-deterministic polynomial time (NP-hard) nature of this problem, centralized techniques are not efficiently practical and result in high com-

* Corresponding author.

E-mail address: rbuyya@unimelb.edu.au (R. Buyya).

<https://doi.org/10.1016/j.jnca.2019.102407>

Received 21 February 2019; Received in revised form 11 June 2019; Accepted 20 July 2019

Available online 22 July 2019

1084-8045/© 2019 Elsevier Ltd. All rights reserved.

plexity, signaling overhead, and single point of failure, specifically in dense and ultra-dense networks (Fu et al., 2017; Rohoden et al., 2019). To overcome this problem, the clustering-based RA techniques, which are partially decentralized, are introduced by which the complexity of RA problem is significantly reduced. In majority of these techniques, each cluster has access to entire set of RBs, while FBSs in one cluster cannot use the same RBs simultaneously. This latter enables RA technique to be performed in each cluster independently of other clusters (Lee et al., 2014).

In order to effectively utilize the benefits of clustering in RA, several issues should be carefully addressed. Clusters can be formed either by gateway (GW) centrally or by FBSs in a distributed manner (Qiu et al., 2016). Moreover, the maximum size and number of clusters can be statically determined or can be obtained dynamically by the GW or cluster heads (CHs) at the runtime. In addition, the RA in each cluster can be performed by a CH individually or all FBSs collaboratively. Besides, in the dense and ultra-dense femtocell networks, interference between clusters should be mitigated so that FBSs located at the edge of clusters (edge FBSs) do not suffer from decreased throughput, which apparently reduces total throughput and end users' quality of experience. Last but not least, it is worth mentioning that centralized and clustering-based RA techniques, in which GWs and CHs respectively perform the majority of responsibilities, suffer from the scalability issues in dense and ultra-dense femtocell networks, because the above-mentioned burdens are not proportionally distributed.

Considering the aforementioned issues, we propose a Distributed Dynamic Clustering (D^2C)-Fog-driven Resource Allocation Technique (D^2C -FORAT) to optimize the total throughput of the downlink OFDMA femtocell networks. The proposed solution is divided into two methods including distributed dynamic clustering and RA, so that we proportionally distribute responsibilities over the network entities including FBSs, CHs, GW, and local fog servers. The fog servers are local entities located in the proximity of end users, which have computing capabilities, and can be accessed by low latency (Zhou et al., 2016; Hu et al., 2017; Chang et al., 2019). In the D^2C -FORAT, FBSs make clusters in a distributed dynamic manner so that FBSs which have the highest co-tier interference on each other join to the same cluster, and select a CH. Afterward, the CH monitors the available resources and users' demands in its cluster, and dynamically control the size of its cluster in the runtime. In addition, the fog servers collect the edge FBSs' information of each cluster which is then used to form the edge FBSs' interference graph. Besides, the fog servers employ a graph-coloring-based technique to assign a set of policies for edge FBSs in each cluster to reduce the inter-cluster interference. These policies are then forwarded to respective CHs, by which the RA can be performed more efficiently, resulting in increased throughput and user satisfaction.

We summarize the main contributions of this paper as follows.

- 1) We propose a hierarchical RA technique, aiming at maximizing the total throughput while mitigating the interference, to satisfy the ever-increasing users' demands in dense and ultra-dense femtocell networks
- 2) We put forward a distributed dynamic clustering algorithm by which CHs adaptively control their cluster size based on requested demands of their end users. This results in better scalability so that our technique can be effectively adapted to dense and ultra-dense femtocell networks.
- 3) Considering the fact that sufficient resources are available in each cluster due to our clustering method, no intra-cluster interference occurs. To address the inter-cluster interference problem, we develop a fog-driven RA method by which the fog servers assign a set of policies to CHs to be considered in their RA. This latter leads to decreasing the inter-cluster interference which significantly improve the total throughput and user satisfaction.

- 4) We study current clustering-based RA techniques in femtocell networks to identify their key elements, and provide a comprehensive qualitative comparison.
- 5) The performance of our technique, D^2C -FORAT, is comprehensively evaluated in dense and ultra-dense femtocell networks, and we compared it by the state-of-the-art current techniques in terms of system throughput, interference, satisfaction rate, and fairness to precisely analyze its efficiency.

The rest of the paper is organized as follows. Section 2 reviews the current literature in clustering-based RA techniques in femtocell networks. The system model and problem formulations are presented in section 3. Our distributed clustering and RA methods are presented in section 4 and section 5, respectively. In section 6, we evaluate the system performance under our proposed solution, and compare it by the state-of-the-art RA techniques of the literature. Finally, section 7 concludes the paper and draws future works.

2. Related work

A significant number of studies has been focused on RA techniques in OFDMA-based femtocell networks to address the co-tier interference, among which we study the current literature in clustering-based RA. The proposed techniques are categorized into two groups of centralized and distributed based on their clustering approach. Besides, main elements of each technique are identified, by which we can qualitatively compare these techniques.

In the centralized clustering techniques, clustering is performed by the GW. The authors in Li et al. (2012) proposed a centralized clustering technique for the RA in femtocell networks, in which, after formation of interference graph, the GW obtains the minimum-interfered clusters by Max k-Cut algorithm. Then, a heuristic algorithm is used to assign available RBs to different clusters. Authors in Abdelnasser et al. (2014) proposed a hierarchical RA technique, in which the GW collects the channel gain between each pair of FBSs and builds the interference graph. The GW forms clusters based on the correlation clustering concept. Afterward, due to the NP-hard nature of correlation clustering, the problem is formulated as a semi-definite program (SDP) and solved by randomized rounding. Authors in Li and Zhou (2017) proposed a cluster-based solution for the RA, in which the FBSs are clustered together according to their geographical positions by the GW. Then, in each cluster, the FBS with the largest interference degree is selected as the CH, whose main task is the RA in its cluster. This latter is performed by solving an optimization problem via a sub-gradient iteration-based RA algorithm. In Fu et al. (2017), the GW collects the interference degree of each FBS and performs a predetermined clustering accordingly. Then, for each cluster derived in the predetermined clustering, the GW obtain specifies a set of best candidate sub-clusters. This process is repeated until the GW recognizes the best sub-clustering for each cluster. The authors in Li and Zhang (2018) proposed a RA technique in which GW centrally clusters FBSs by a modified k-mean clustering algorithm. Afterward, a greedy algorithm is used to distribute available resources to the FBSs.

Although the main goal of centralized clustering techniques is to find the best clustering, this is a time-consuming process, specifically in the dense and ultra-dense femtocell networks (Qiu et al., 2016). Moreover, considering the dynamic nature of femtocell networks, the GW requires to re-cluster all the FBSs whenever any change occurs in the network to find the best clustering solution.

In the distributed clustering techniques, FBSs collaboratively make clusters without participation of the GW. If any changes occur in the status of the network and end users, the change can be handled locally, and there is no need to repeat the clustering algorithm for all FBSs. Authors of Zhang et al. (2013) proposed a graph-based RA technique in which clusters of non-interfering FBSs are formed in a distributed manner by FBSs, and then, the GW performs the RA for each cluster

according to its average users' demands. Since the RA is performed centrally by the GW, this technique cannot be efficiently employed in dense and ultra-dense networks. Moreover, RBs are not fairly allocated to end users because this technique only considers average demand of each cluster. However, each user's demand can be more than the obtained cluster's average demand, resulting in less fairness and user satisfaction. In [Hatoum et al. \(2014\)](#), the authors proposed a Quality-of-service-based Femtocell Cluster-based RA (QFCRA). Initially, each FBS builds a neighboring list containing its one-hop FBSs, and sends it to all of its proximate neighbors. This latter assists each FBS to obtain interference degree of its one-hop neighbors. Afterward, the FBS with the largest interference degree among its neighbors announces itself as the CH, and other FBSs connect themselves to it. Each CH has the responsibility of RA, especially in the dense and ultra-dense femtocell networks. In [Qiu et al. \(2016\)](#), the authors proposed a learning-based scheme (LFCRA) to solve the inter-cluster interference of the QFCRA. Although the proposed technique is more efficient than QFCRA in handling inter-cluster interference, similar to QFCRA, its performance mainly depends on its cluster size which is not dynamically set.

[Table 1](#) identifies and compares key elements of current works in terms of architectural parameters, clustering parameters, interference management, and evaluation parameters. The clustering parameters include approach describing whether that proposal is centralized or distributed, dynamicity showing whether that cluster updates itself whenever a new FBS is added or removed, or even when the users' demands are changed. The cluster size can be set by GW or CHs in a predefined manner statically or according to the current status of FBSs dynamically. The clustering criteria show the main factors used for creation of clusters, which can be interference degree, interference intensity, and users' demands. The architectural parameters contain hierarchical components which describe what entities participate in clustering and RA, and introduce specific roles of each entity. Furthermore, although all proposals consider intra-cluster interference management, only some of them address the inter-cluster interference which has a significant effect on network throughput, specifically in the densely deployed femtocell networks. Finally, the evaluation parameters depict main parameters by which the performance of each proposal is evaluated.

To address the above-mentioned issues, we propose a hierarchical technique, called D^2C -FORAT, containing clustering and RA methods. The clustering is performed in a distributed manner, where FBSs collaboratively form clusters based on interference intensity, and cluster members (CMs) identify a CH for each cluster. Each CH dynamically controls the size of its cluster in the runtime based on available resources and interference intensity. Besides, CHs, based on a proposed routine, dynamically update the cluster parameters whenever users' demands change or a FBS send join/disjoin message to the CHs. In addition, the RA method concurrently addresses the intra and inter-cluster interference problem. Since each CH guarantees that there is always sufficient RBs for FBSs within a cluster, the intra-cluster interference never occurs. Moreover, we employ local fog servers, that are aware of location and demands of edge FBSs, to run a graph-coloring-based algorithm to address the inter-cluster interference.

3. System model and formulation

In this section, we describe the system model and formulate the RA as an optimization problem to maximize the network throughput.

3.1. System model

We consider an OFDMA-based femtocell network in which FBSs are densely deployed. In such networks, FBSs may face to two types of interferences including cross-tier interference (i.e., interference between femtocell and macrocells ([Mhiri et al., 2013](#))), and co-tier interference.

Table 1
Qualitative comparison of clustering-based resource allocation techniques.

Proposal	Clustering Parameters			Architectural Parameters			Interference Management			Evaluation Parameters				
	Approach	Dynamicity	Size	Criteria	Hierarchical Components	GW Role	CH Role	CM Role	Intra Cluster	Inter Cluster	Throughput	Interference	Fairness	Satisfaction
Li et al. (2012)	Centralized	D	D	Intf I	GW, CM	Clustering, RA	-	-	✓	×	✓	×	×	×
Abdelhasser et al. (2014)		S	D	Intf I	GW, CH, CM	Initial Clustering, CH Sel	RA, Improve Clustering	-	✓	×	✓	✓	×	×
Li and Zhou (2017)		S	S	Intf D	GW, CH, CM	Clustering	RA, PA	CH Sel	✓	×	✓	×	×	×
Fu et al. (2017)		S	S	Intf D	GW, CH, CM	Initial Clustering, CH Sel	RA, PA	-	✓	✓	✓	×	✓	×
Li and Zhang (2018)		S	D	Intf D	GW, CH, CM	Clustering, CH Sel	RA	-	✓	×	✓	×	×	✓
Zhang et al. (2013)	Distributed	S	S	Intf D	GW, CM	RA	-	-	✓	×	×	×	✓	×
Hatoum et al. (2014)		S	S	Intf D	CH, CM	-	RA	CH Sel	✓	✓	×	×	✓	✓
Qiu et al. (2016)		S	S	Intf D	CH, CM	-	RA, Power Adj	CH Sel	✓	✓	✓	×	✓	×
D^2C -FORAT		D	D	Intf I, Intf D, users' demands	GW, Fog Servers, CH, CM	Set Policies for RA	RA, Notify fog Server, Cluster size Control	CH Sel	✓	✓	✓	✓	✓	✓

The abbreviated terms are as follows: Clustering Parameters (D:Dynamic, S:Static, Intf I=Interference Intensity, Intf D = Interference Degree), Architectural Parameters (GW:Gateway, CH:Cluster Head, CM:Cluster Member, RA:Resource Allocation, CH Sel:CH Selection, PA:Power Allocation, Power Adj:Power Adjustment).

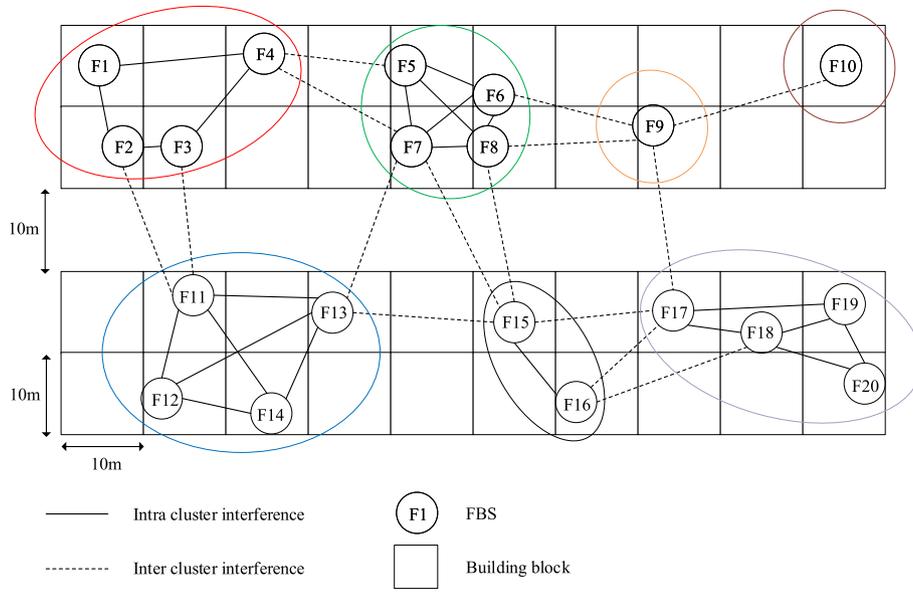


Fig. 1. The 3GPP dual-strip residential apartment model.

This latter interference, which is significantly aggravated in dense and ultra-dense networks, can be reduced by forming clusters of FBSs, and coordinating among them through their X2 interfaces (Pratap et al., 2018). Besides, the interference between macrocell and FBSs can be regarded as Additive White Gaussian Noise (AWGN) (Kim and Cho, 2010).

We use the 3GPP dual-strip residential apartment model (3GPP, 2010) to represent how FBSs are deployed in our network. In this model, we have two strips of one-floor buildings, so that strips are separated by a 10m-wide street, and each one contains 20 buildings (10 m × 10 m). Each building comprises a FBS, so that its location is set based on uniform distribution model. In addition, since the owners can turn on/off their FBSs randomly, which change the network topology, we set the activation status of each FBS, $S_a \in \{0, 1\}$, and FBS density, $\lambda \in [0, 1]$. This latter represents the ratio of active FBSs to all FBSs in the network (Lee et al., 2017). Each active FBS can support up to maximum of four end users that are uniformly distributed in its proximity. Moreover, the user demand of each FBS is defined as the number of requested RBs by that user. Fig. 1 represents an example of FBSs' deployment in 3GPP dual-strip apartment model, where FBSs are grouped into seven different clusters. Moreover, this Figure demonstrates the concept of intra and inter-cluster interference.

In our model, we use LTE specification for downlink, in which the available 5 MHz bandwidth is divided into a number of RBs so that each RB contains 12 consecutive subcarriers with 15 KHz of spacing between adjacent subcarriers, and 7 OFDMA symbols with the time duration of 0.5 ms (Capozzi et al., 2013).

3.1.1. Hierarchical architecture

We propose a hierarchical architecture in which responsibilities are proportionally distributed over the network entities including GW, fog servers, CHs, and CMs, in order to meet the requirements of dense and ultra-dense networks. Fig. 2 depicts an overview of our hierarchical architecture. In what follows, we briefly illustrate responsibilities of each entity, and define how they collaborate.

The GW. The main responsibility of this entity is managing Operation and Maintenance (OAM) information, including FBSs' location, identification, authentication, aggregating and validating signaling traffic (Li et al., 2012).

The Fog Server. This entity is located between the GW and CHs,

and its main responsibility is forwarding policies to its in-range CHs so that they can efficiently allocate RBs to their FBSs. It receives the clusters' configuration of FBSs and forms an edge FBSs' interference graph. Afterward, by means of its computing capacity, the fog server attempts to solve the inter-cluster interference and forward specific policies to CHs.

The CH. The CH is placed between CMs and the fog server, and its main duties include notifying the fog server of its edge FBSs' configuration, and RA runtime. This feature helps the clusters to dynamically change and adapt themselves to the current state of the network. Furthermore, it periodically notifies the fog server about its edge FBSs' configuration. This latter is largely because each CH has a local view of its CMs' configuration, and is not aware of other clusters' configurations, which results in less-precise RA. Finally, it allocates available resources to its CMs while considers the received policies from the fog servers. This leads to more efficient RA which improves the total throughput and satisfaction of end users.

The FBS. The main responsibilities of FBSs in this architecture, alongside with satisfying their users, are cluster formation and CH selection which are obtained in a distributed dynamic manner.

3.2. Problem formulation

We define a set of FBSs as $\mathcal{F} = \{f_1, f_2, f_3, \dots, f_M\}$, so that each FBS is a member of disjoint cluster set $\mathcal{C} = \{c_1, c_2, c_3, \dots, c_L\}$. Hence, each FBS $f_i \in c_l$ can be represented by $f_{i,l}$. Moreover, we define the in-range neighbors of FBS f_i as a set of FBSs shown by \mathcal{N}_{f_i} . It is important to note that \mathcal{N}_{f_i} contains members that are not necessarily in the same cluster as f_i . Moreover, the set of end users of f_i are defined as \mathcal{U}_{f_i} .

We denote the set of RBs as Δ , and hence, the received amount of signal to interference plus noise (SINR) of each $u \in \mathcal{U}_{f_i}$ on the RB $k \in \Delta$ is defined as follows (Le et al., 2018; Ha and Le, 2014):

$$\gamma_{u,k}^{f_{i,l}} = \frac{P_k^{f_{i,l}} \times H_{u,k}^{f_{i,l}}}{\sigma^2 + \sum_{f_{j,l'} \in \mathcal{F}, j \neq i, l' \neq l} P_k^{f_{j,l'}} \times H_{u,k}^{f_{j,l'}}} \quad (1)$$

where $P_k^{f_{i,l}}$ and $H_{u,k}^{f_{i,l}}$ are the transmission power of $f_{i,l}$ and the channel gain between u and $f_{i,l}$ on RB k , respectively. Moreover,

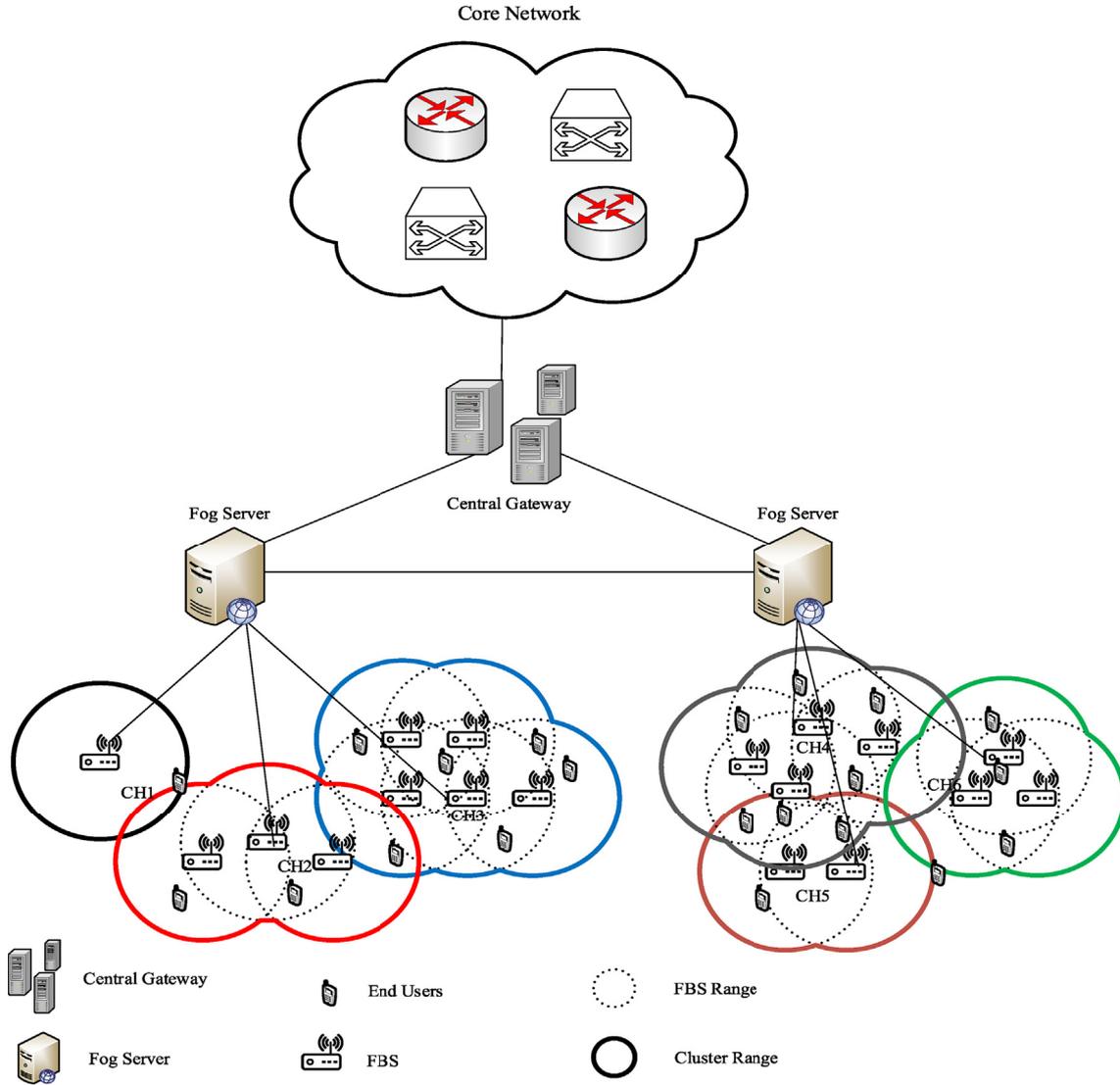


Fig. 2. An overview of proposed hierarchical architecture.

$\sum_{f_{j,l'} \in \mathcal{F}, j \neq i, l' \neq l} p_k^{f_{j,l'}} \times H_{u,k}^{f_{j,l'}}$ is the interference generated by other adjacent FBSs belonging to other clusters, called inter-cluster interference, and σ^2 is noise power density.

According to the amount of $\gamma_{u,k}^{f_{i,l}}$ on RB k , the user u can select an appropriate Modulation-and-Coding Scheme (MCS) to achieve the highest-possible data rate while guarantee reliability. We represent the achievable data rate on the RB of this user by $R_{u,k}^{f_{i,l}}$ so that this should be always between the maximum and minimum achievable data rate, which is calculated based on MCS and SINR threshold used in Mach and Becvar (2011), as follows:

$$R_{\max} = (7\text{symbol} \times 4.8\text{bit/symbol}) \times .5\text{ms} \times 12 = 806.4\text{kbs} \quad (2)$$

$$R_{\min} = (7\text{symbol} \times .66\text{bit/symbol}) \times .5\text{ms} \times 12 = 110.88\text{kbs} \quad (3)$$

To calculate R_{\max} and R_{\min} , the QAM-64 with code rate 4/5 and the QPSK with code rate 1/3 are used, respectively.

The principal goal of this work is to maximize the total throughput of the network while mitigating the severe interference through allocating appropriate RBs to each FBS. Thus, according to the above-mentioned goal, the clustering-based RA problem can be formulated as

follows:

$$\text{maximize} \sum_{c_l \in \mathcal{C}} \sum_{u \in \mathcal{U}_{f_i}} \sum_{k \in \Delta} d_{u,k}^{f_{i,l}} \times R_{u,k}^{f_{i,l}} \quad (4)$$

s.t.

$$C1 : d_{u,k}^{f_{i,l}} \in \{0, 1\}, \quad \forall u \in \mathcal{U}_{f_i}, \forall f_i \in c_l, \forall c_l \in \mathcal{C}, \forall k \in \Delta$$

$$C2 : \sum_{f_i \in c_l} \sum_{u \in \mathcal{U}_{f_i}} \sum_{k \in \Delta} d_{u,k}^{f_{i,l}} \leq |\Delta|, \quad \forall c_l \in \mathcal{C}$$

$$C3 : \sum_{f_i \in c_l} \sum_{u \in \mathcal{U}_{f_i}} d_{u,k}^{f_{i,l}} \leq 1, \quad \forall c_l \in \mathcal{C}, \forall k \in \Delta$$

$$C4 : \sum_{k \in \Delta} d_{u,k}^{f_{i,l}} \times R_{u,k}^{f_{i,l}} \geq \hat{R}_u^{f_{i,l}}, \quad \forall u \in \mathcal{U}_{f_i}, \forall f_i \in c_l, \forall c_l \in \mathcal{C}, \forall k \in \Delta$$

$$C5 : d_{u,k}^{f_{i,l}} \times R_{u,k}^{f_{i,l}} \geq d_{u,k}^{f_{i',l'}} \times R_{\min}, \quad \forall u \in \mathcal{U}_{f_i}, \forall f_i \in c_l, \forall c_l \in \mathcal{C}, \forall k \in \Delta$$

$$C6 : \bigcup_{c_l \in \mathcal{C}} c_l = \mathcal{F}$$

$$C7 : c_l \cap c_{l'} = \emptyset, \quad \forall c_l, c_{l'} \in \mathcal{C}, l \neq l'$$

Table 2
Parameters and respective definitions.

Parameter	Definition	Parameter	Definition
S_a	Activation status of each FBS	λ	FBSs' density
\mathcal{F}, \mathcal{M}	Set of all FBSs, Number of all FBSs	f_i	The i th FBS
$f_{i,l}$	The FBS f_i in cluster c_l	C, L	Set of clusters, Number of clusters
$\mathcal{N}_{f_i}^*$	Set of one hop neighbors of FBS f_i	$\mathcal{U}_{f_i}^*$	Set of end users of FBS f_i
$\gamma_{u,k}^{f_{i,l}}$	The SINR that end user u receives from FBS $f_{i,l}$ on RB k	$R_{u,k}^{f_{i,l}}$	Data rate of end user u belongs to $f_{i,l}$ on RB k
Δ	Set of all RBs	σ^2	Noise power density
$P_k^{f_{i,l}}$	Transmission power of FBS $f_{i,l}$ on RB k	$\hat{R}_u^{f_{i,l}}$	Minimum data rate required by end user u
$H_{u,k}^{f_{i,l}}$	Channel gain between the FBS $f_{i,l}$ and the end user u on the RB k	$I(f_i, c_l)$	Relative sum interference of FBS f_i on cluster c_l
R_{\max}	Maximum throughput on each RB	R_{\min}	Minimum data rate on each RB
FC_{c_l}	Free capacity of cluster c_l	$f_{c_l}^*$	The worst CM in the cluster c_l
$\mathcal{N}(f_i, c_l)$	Set of one-hop neighbors of FBS f_i belonging to cluster c_l	$I(f_i, f_j)$	Interference between FBS f_i and FBS f_j
$deg(f_i, c_l)$	Relative interference degree of FBS f_i on cluster c_l	\mathcal{P}_{c_l}	Set of all possible partitions for cluster c_l
$policy_{c_l}$	The set of policies enacted for the cluster c_l	$range(f_{i,l})$	The set of authorized RBs for $(f_{i,l})$
G	The interference graph of edge FBSs	K_{\max}	The maximum number of colors
$d_{u,k}^{f_{i,l}}$	Exclusion factor indicating whether RB k is assigned to end user u of FBS $f_{i,l}$, or not	$FC_{c_l}^*$	Free capacity of cluster c_l without considering $f_{c_l}^*$
$I^*(f_i, c_l)$	Relative sum interference of FBS f_i on cluster c_l without considering the $f_{c_l}^*$	$demand_u$	The demand of end user u in terms of number of RBs

In the optimization problem (4), the $d_{u,k}^{f_{i,l}}$ is an exclusion factor to represent whether the RB k is assigned to the user u of FBS $f_{i,l}$ or not, as depicted in the constraint C1. The second constraint, C2, expresses that each cluster can use all available network's RBs, while the C3 denotes that each RB k in each cluster can be assigned only to one user. Thus, there is no intra-cluster interference in our problem. The C4 indicates that each user u of FBS $f_{i,l}$ should at least receive its minimum requested data rate, depicted as $\hat{R}_u^{f_{i,l}}$. The C5 expresses the fact that the RB k should not be assigned by FBS f_i to its end users whenever a severe interference is existed on that RB from. The C6 and C7 denote that each FBS f_i is a member of one cluster, and the set of clusters are disjoint. Table 2 summarizes the parameters used in this paper and their respective definitions.

4. Distributed dynamic clustering method

In this section, we propose a Distributed Dynamic Clustering (D^2C) method, in which FBSs with highest relative interference form different clusters. Moreover, FBSs in each cluster select one FBS as their CH. The CH dynamically controls the cluster size based on requested demands of its end users and makes the decision whether a new FBS can join the cluster or not, accordingly. The D^2C has three principal functions including new FBS arrival (NFA), update clustering parameters (UCP), and cluster migration possibility (CMP), as discussed in this section.

4.1. New FBS arrival (NFA)

Whenever a new FBS f_i joins the network, it creates its neighbor list $\mathcal{N}_{f_i}^*$, in which each $f_j \in \mathcal{N}_{f_i}^*$ is either a CM or CH. If $\mathcal{N}_{f_i}^*$ does not have any CH member, the f_i creates the cluster c_l , set itself as the CH, and calculates the free capacity of the cluster as follows:

$$FC_{c_l} = |\Delta| - \sum_{f_j \in c_l} \sum_{u \in \mathcal{U}_{f_j}^*} demand_u \quad \forall c_l \in C \quad (5)$$

where $demand_u$ depicts the number of RBs requested by the user $u \in \mathcal{U}_{f_i}^*$.

If the $\mathcal{N}_{f_i}^*$ contains CH members, the f_i sends message to those CHs and requests their free capacities, FC s. Afterward, clusters whose FC s are greater than or equal to $\sum_{u \in \mathcal{U}_{f_i}^*} demand_u$ are considered as candidate clusters by f_i . To select the best candidate cluster to join, the f_i calculates the relative sum interference of itself on all members of each candidate

cluster as follows:

$$I(f_i, c_l) = \sum_{f_j \in c_l, i \neq j} I(f_i, f_j) \quad (6)$$

where $I(f_i, f_j)$ represents the interference between f_i and f_j as calculated in Tan et al. (2011).

$$I(f_i, f_j) = P_i^j \times H_{f_i, f_j}^j \quad (7)$$

To simplify the problem, we assume the mutual interference between two FBSs is symmetric, as shown in the following:

$$I(f_i, f_j) = I(f_j, f_i) \quad (8)$$

Among all candidate clusters, f_i selects the candidate cluster by which it has the highest relative sum interference, $I(f_i, c_l)$, and sends the soft-join request to that cluster's CH. The soft-join indicates that the candidate cluster has enough FC to accept the join-request of f_i while the CH guarantees to allocate sufficient requested RBs to its current end users.

If there is no candidate clusters with sufficient FC s to support the required demand of f_i , the hard-join is considered as a potential solution. In this latter, the candidate cluster c_l should substitute its worst FBS, called $f_{c_l}^*$, for the f_i . To identify the $f_{c_l}^*$, each $f_{j,l}$ creates the $\mathcal{N}(f_j, c_l) \subseteq \mathcal{N}_{f_j}^*$ where $\mathcal{N}(f_j, c_l)$ denotes one-hop neighbors of f_j that are in the cluster c_l . Moreover, the FBS f_j calculates its relative interference degree on cluster c_l as $deg(f_j, c_l) = |\mathcal{N}(f_j, c_l)|$. Hence, the CH selects the member with the lowest relative interference degree as $f_{c_l}^*$ on its cluster. In a case that there are several members with the lowest $deg(f_j, c_l)$, the member whose interference with other members is the lowest, is selected as the $f_{c_l}^*$. Considering the fact that any cluster c_l has its $f_{c_l}^*$, the f_i sends message to each neighboring CH to obtain its free capacity while worst FBS $f_{c_l}^*$ is excluded, as shown in the following:

$$FC_{c_l}^* = FC_{c_l} + \sum_{u \in \mathcal{U}_{f_{c_l}^*}^*} demand_u, \quad \forall c_l \in C \quad (9)$$

Afterward, each cluster whose $FC_{c_l}^*$ is greater than or equal to $\sum_{u \in \mathcal{U}_{f_i}^*} demand_u$ is considered as candidate cluster for hard-join. Among these candidate clusters, the f_i sends the hard-join request to the CH of the cluster by which it has the highest relative sum interference $I^*(f_i, c_l)$.

$$I^*(f_i, c_l) = I(f_i, c_l) - I(f_i, f_{c_l}^*) \quad (10)$$

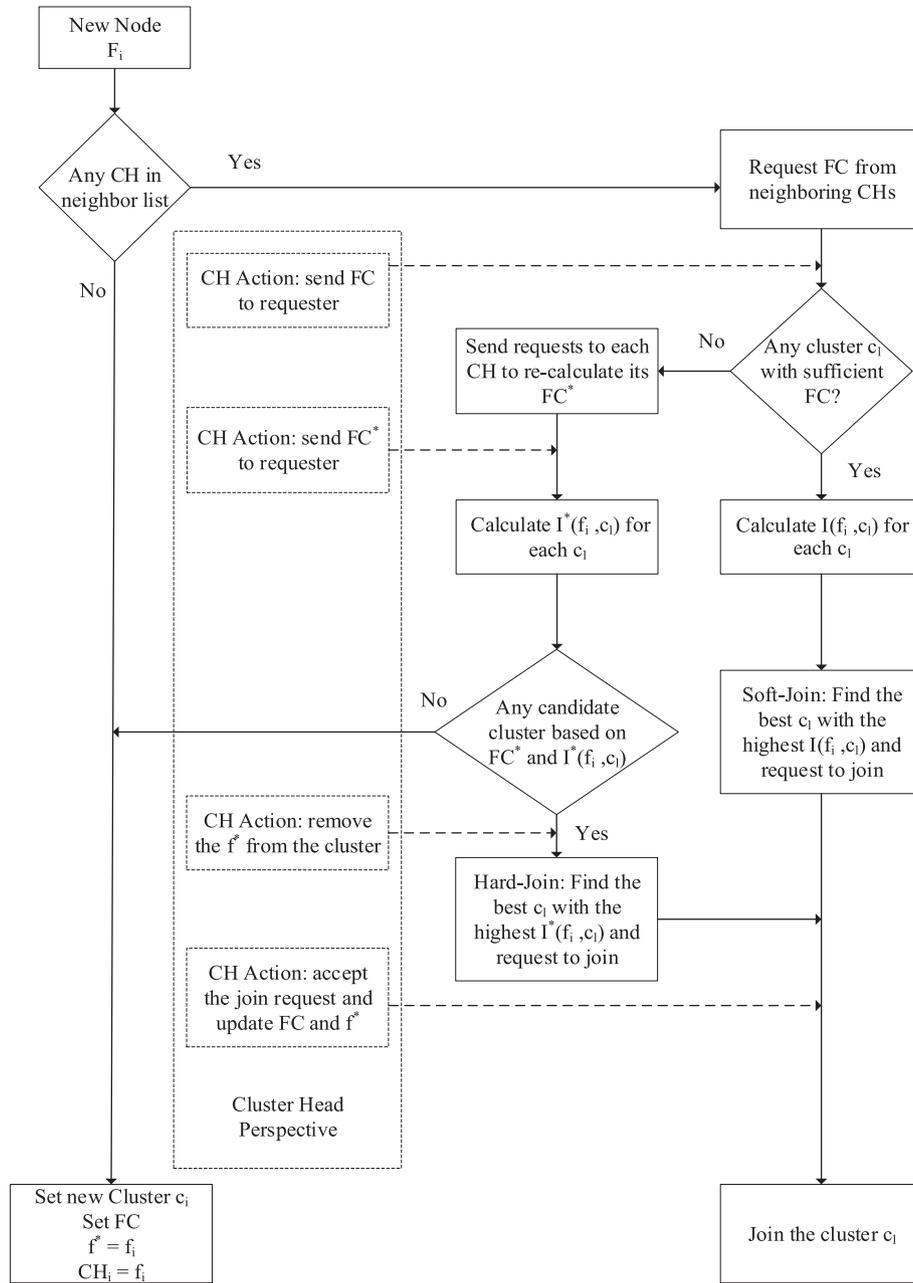


Fig. 3. New FBS Arrival (NFA) flowchart.

In a case that there is no possibility to perform soft-join and hard-join, the f_i acts exactly the same as the condition that there is no CHs in its \mathcal{N}_{f_i} , as discussed earlier, and forms a new cluster. Fig. 3 represents the process of NFA function.

4.2. Update clustering parameters (UCP)

When the configuration of a cluster c_l changes (e.g. joining or removing a member), its CH requests all its members $f_{x,l}$ to calculate their $deg(f_{x,l}, c_l)$ and $I(f_{x,l}, c_l)$, and send them back. Then, it makes a priority list of its members according to these parameters so that members with the largest relative interference degree receive higher priority. If there are several members with the same relative interference degree, those members are sorted in terms of the relative sum interference. Finally, the member with the highest priority is selected as the CH, and the member with the lowest priority is selected as the $f_{c_l}^*$.

4.3. Cluster migration possibility (CMP)

The main goal of this function is to check whether any CM can migrate to other clusters so that the quality of clustering improves or not. This function is called by the CMs that at least have one another CH in their neighbor lists belonging to another cluster. To illustrate, we consider $f_{i,l}$ and $f_{j,l'}$, $f_j \in \mathcal{N}_{f_i}$ and the fact that $f_{j,l'}$ is the CH of $c_{l'}$. Then, the $f_{i,l}$ should request the $FC_{c_{l'}}$ and periodically calculate the $I(f_i, c_{l'})$. If $I(f_i, c_l) < I(f_i, c_{l'})$, then f_i sends the soft-join request to the f_j to join the cluster $c_{l'}$. This guarantees that each FBS always attempts to join a cluster to which it has the highest relative interference. This latter helps to improve clusters as the network configuration changes, and consistently attempts to maintain FBSs with highest relative sum interference as a cluster, which finally leads to less inter-cluster interference.

Algorithm 1 General view of policy identification algorithm.

```

%Graph Formation Phase:
1 create  $G(V, E, W_v, W_e)$ 
  /* The graph  $G$  is comprised of
  several connected graph  $g_i$  as
   $G = \{g_1, g_2, \dots, g_Z\}$  */
%Graph Coloring Phase:
2 for  $z = 1$  to  $Z$  do
3   calculate  $K_{max}$  from eq. 17
  /* The  $K_{max}$  is the maximum number
  of colors */
4   for  $k=2$  to  $K_{max}$  do
5      $g_z^* = \text{graph-simplification}(g_z, k)$ 
6     if  $\text{graph-coloring}(g_z^*, k) == \text{false}$  then
7       if  $k + 1 \leq K_{max}$  then
8         continue
9       else
10        %Graph Relaxation Phase:
11         $g_z = \text{graph-relaxation}(g_z)$ 
12        go to line 3
13      end
14    else
15      send policies to CHs
16      break
17    end
18  end

```

5. A new resource allocation method

In this section, we propose a RA method in which fog servers and CHs collaborate to mitigate the interference and improve the network throughput.

Each CH is responsible to allocate the RBs so that no intra-cluster interference occurs in its cluster. Since each CH is unaware of adjacent clusters' RAs, there is high probability of inter-cluster interference on edge FBSs. This problem is aggravated in dense and ultra-dense networks to the point that the network throughput is severely dropped (Qiu et al., 2016). Hence, we concurrently consider both intra and inter-cluster interference for the RA so that our method can be applied to femtocell networks ranging from sparse to ultra-dense. In this method, the fog servers are responsible to provide a set of policies to CHs in order to minimize the inter-cluster interference. The CHs consider these policies and their users' demands, and aim at maximizing clusters' throughput alongside with decreasing the interference by proper allocation of resources.

5.1. Policy identification

We divide the FBSs of each cluster into two categories containing central and edge FBSs. The $f_{i,l}$ is considered as a central member whenever \mathcal{N}_{f_i} only contains neighbors from c_l , while it is considered as edge member if the \mathcal{N}_{f_i} contains any member from other clusters, as noted in equation (11). Apparently, the central nodes never experience inter-cluster interference.

$$f_{i,l} \text{ is } \begin{cases} \text{Central,} & \text{if } \mathcal{N}_{f_i} - \mathcal{N}(f_i, c_l) = \emptyset \\ \text{Edge,} & \text{if } \mathcal{N}_{f_i} - \mathcal{N}(f_i, c_l) \neq \emptyset \end{cases} \quad (11)$$

The fog servers provide a set of policies for each cluster c_l , in which each element contains the specific FBS on which that policy should be applied, and specific subset of RBs, called *range* representing the RBs which can be assigned to that FBS, as shown in the following.

$$\text{policy}_{c_l} = \{(f_{i,l}, \text{range}(f_{i,l})) \mid f_{i,l} \text{ is Edge, } \text{range}(f_{i,l}) \subseteq \Delta\} \quad (12)$$

The policies to mitigate the inter-cluster interference are provided in three phases including graph formation, graph coloring, and graph relaxation, as discussed in the following. The Algorithm 1 represents a general view of policy identification through these phases.

5.1.1. Graph formation

The main goal of this phase is to form an interference graph of edge FBSs. To achieve this, CHs send their neighbor lists of edge FBSs and their respective demands to the fog servers. Afterward, the fog servers create the weighted graph of edge FBSs, $G(V, E, W_v, W_e)$, based on the information received from their corresponding CHs where V represents the set of vertices, so that each vertex denotes an edge FBS. The E explains the set of edges, so that each edge represents the interference between two edge FBSs, as follows:

$$e_{v_i, v_j} \text{ is } \begin{cases} 1, & \text{iff } f_i \in \mathcal{N}_{f_j} \\ 0, & \text{iff } f_i \notin \mathcal{N}_{f_j} \end{cases} \quad (13)$$

It is important to note that based on equation (13), edge FBSs that are in the same cluster and in range of each other are connected by an edge in the G . Moreover, the W_v and W_e depict the set of weights for vertices and edges respectively. The weight of each vertex v_i is calculated as the total users' demands of each edge FBS as follows:

$$W_{v_i} = \sum_{u \in U_{f_i}} \text{demand}_u \quad (14)$$

In addition, the weight of each edge e_{v_i, v_j} is the amount of interference between those vertices, as depicted in the following:

$$\text{if } e_{v_i, v_j} \in E \text{ then } w_{v_i, v_j} = I(f_i, f_j) \quad (15)$$

Finally, the weighted graph G is not necessarily a connected graph, and it can be comprised of several connected graphs $G = \{g_1, g_2, \dots, g_Z\}$, with the following condition:

$$g_z \cap g_{z'} = \emptyset, \quad \forall g_z, g_{z'} \in G, \quad z \neq z' \quad (16)$$

5.1.2. Graph coloring

In this phase, we address the inter-cluster interference by a graph coloring method for every connected graph g_z in the G , created in the graph formation phase. The main goal of graph coloring method is to find a set of different colors (so that each color represents a set of RBs) for edge FBSs and assign a set of respective policies to CHs for the RA, so that the inter-cluster interference reduces.

Considering the fact that each assigned color to an edge FBS should contain sufficient RBs to support demands of all users belonging to that

FBS, the maximum number of colors K_{\max} is restricted and is calculated as follows:

$$K_{\max} = \lfloor \frac{\Delta}{W_x} \rfloor, \quad x = \arg \max (W_{v_i}), \quad \forall v_i \in V_{g_z} \quad (17)$$

where the W_x represents the weight of the heaviest vertex $x \in V_{g_z}$.

Thus, the graph coloring problem is changed to the k-coloring problem, so that we should iterate from $k = 2$ to K_{\max} to find the least-possible number of colors, which is a time-consuming problem. In this phase, we also propose a greedy simplification algorithm to simplify the connected graph g_z , so that we can color this graph in a timely manner. The Algorithm 2 demonstrates the graph simplification.

Algorithm 2 Graph simplification.

```

input      :  $g_z$ : An instance of subgraph,
             $k$ : The number of colors
/*  $\mathcal{P}_{c_l}$ : Set of all possible
partitions of  $c_l$ ,  $\Pi$ : The
partition with the lowest number
of classes */
1 initialize  $g_z^* = g_z$ 
2 for  $l = 1$  to find-size-cluster( $g_z$ ) do
/* The method
find-size-cluster( $g_z$ )
determines the number of
clusters belongs to  $g_z$  */
3  $\mathcal{P}_{c_l} = \text{make-partition}(c_l)$ 
/* The make-partition ( $c_l$ ) method
creates ascending sorted list
of all partitions, and its size
is obtained from Dobinsky
Formula */
4  $\Pi = \text{find-partition}(\mathcal{P}_{c_l})$ 
/* The find-partition( $\mathcal{P}_{c_l}$ ) method
returns the partition with the
least number of classes,
 $\Pi = \{A_1, A_2, \dots, A_t\}$ ,  $t = \text{number of
classes in } \Pi$  */
5 for  $j = 1$  to  $t$  do
6   if  $\sum_{v_i \in A_j} W_{v_i} > \frac{|\Delta|}{k}$  then
7      $\mathcal{P}_{c_l} = \mathcal{P}_{c_l} - \Pi$ 
8     go to line 4
9   end
10 end
11 for  $j = 1$  to  $t$  do
12   combine( $A_j$ )
/* The combine( $A_j$ ) method
merges all vertices in  $A_j$  */
13 end
14 end

```

The simplification algorithm starts from the least possible number of colors $k = 2$, and tries to combine the edge FBSs of each cluster together, if possible, and creates the g_z^* . We can combine any two edge FBSs $v_i, v_j \in g_z$ if they belong to same cluster, and an edge $e_{v_i, v_j} \in E_{g_z}$ exists between them, and their aggregate vertices' weights is not higher than the capacity of colors ($\frac{|\Delta|}{k}$). According to this latter, we can make a new vertex v_{ij} in the g_z^* , whose weight is the aggregate weights of v_i, v_j , and v_{ij} connects to any vertices to which v_i or v_j was connected previously.

if $v_i, v_j \in c_l$ & $e_{v_i, v_j} \in E_{g_z}$ & $w_{v_i} + w_{v_j}$

$$\leq \frac{|\Delta|}{k} \text{ then combine}(v_i, v_j) \text{ as } v_{ij}, \quad \forall v_i, v_j \in V_{g_z}, \forall c_l \in C \quad (18)$$

It is crystal clear that if the number of edge FBSs within the c_l is equal to n , $n > 2$, several configurations for combination of any number of

edge FBSs in c_l can be considered, which is obtained by the Dobinsky's formula, as follows (Pitman, 1997).

$$|\mathcal{P}(c_l)| = \sum_{m=1}^n \frac{m^n}{m!} \sum_{j=0}^{n-m} \frac{(-1)^j}{j!} \quad (19)$$

where we denote the set of all possible partitions of cluster c_l as $\mathcal{P}(c_l)$, in which each partition is a set of classes $\{A_1, A_2, \dots, A_t\}$. Each class A_j contains one or more edge FBSs of cluster c_l that should be evaluated whether they can be considered as one vertex or not. The *make-partition()* method creates an ascending sorted list of all partitions based on the number of classes, t , of the partitions. In the next step, the *find-partition()* method is responsible to return the partition with the least number of classes, defined as Π , from the partition set $\mathcal{P}(c_l)$. For each class A_j in Π , it is examined whether the aggregate weights of that class is less than $\frac{|\Delta|}{k}$ or not. If there is even one class in the partition Π that does not satisfy this condition, the partition is removed from the $\mathcal{P}(c_l)$ and the algorithm searches for the next candidate partition. However, if all classes satisfy the condition, FBSs of the same class can be combined together and new simplified graph g_z^* is made.

If g_z^* is colored by k colors, the range of RBs for each color can be specified and respective policies will be sent to the CHs. But, in a case that the g_z^* cannot be colored by k colors, the k increases and the *graph-simplification()* method is invoked to simplify the graph. Fig. 4 depicts an example of graph coloring phase. Fig. 4a represents the graph of edge FBSs derived from FBSs' configuration of Fig. 1. Fig. 4b demonstrates a candidate simplified graph g_z^* , in which several nodes in each cluster are combined. Afterward, the g_z^* is colored by three colors as it can be seen in Fig. 4c, and Fig. 4d denotes how these colors are represented in the g_z .

In an ultra-dense network that even simplification cannot help to color the graph by K_{\max} colors, we provide a backup plan as graph relaxation phase.

5.1.3. Graph relaxation

This phase is the backup phase for the graph coloring, and is invoked if graph coloring cannot find any solution to color the g_z up to K_{\max} colors. The graph relaxation phase attempts to decrease the maximum vertex degree of sub-graph g_z by ignoring the weak interferences leading to creation of new g_z with less constraints. The *relaxation()* method receives the subgraph g_z , and finds the vertex or vertices with the largest interference degree because they incur the strongest constraints on the problem. Afterward, the lightest edge of those vertices is omitted to reduce their interference degree, as shown in the following:

$$\text{if } \text{deg}(v_x) = \text{deg}(g_z) \text{ then } E_{g_z} = E_{g_z} - e_{v_x, v_y}, \quad (20)$$

$$y = \arg \min (W_{v_x, v_k}), \forall v_k \in V_{g_z}, e_{v_x, v_k} \in E_{g_z}, \forall v_x \in V_{g_z}$$

The new subgraph g_z is then created and sent to the graph coloring phase.

5.2. Policy aware resource allocation

Considering the fact that CHs receive policies for their edge FBSs from fog servers, they should apply those policies in their RA. The Algorithm 3 indicates an overview of policy aware RA. To achieve the aforementioned goals, the FBSs that are not assigned any RBs, are divided into two sets including S_1, S_2 by each CH. The first set S_1 belongs to FBSs for which the fog server sends some policies to their respective CHs, while the second set S_2 contains the FBSs for which no policies are assigned. Because more restrictions are applied on the S_1 , it has higher priority compared to S_2 whenever CH assigns resources. These lists are sorted based on total users' demands of FBSs, so that FBSs with the highest required RBs that are not satisfied yet are placed in front of the

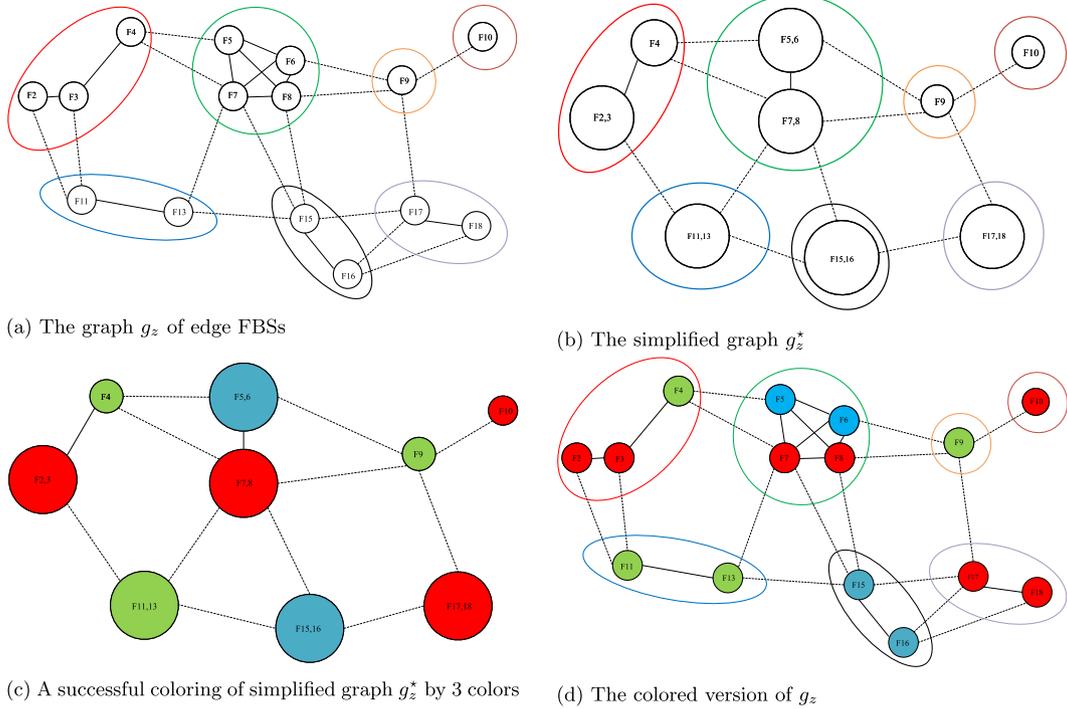


Fig. 4. An example demonstrating graph coloring phase based on FBS configuration depicted in Fig. 1.

lists. In each iteration, each CH selects an RB from the unallocated RBs, shown as Δ' , and assigns that RB to a FBS f_i , existing in S_1 , if that RB is in $range(f_i)$. In the case that there exists no FBS in the S_1 or the $range$ of that FBS does not comply with the selected FBS, the CH searches the S_2 to find a proper FBS to which it can assign that RB. The algorithm is finished whenever there is no FBSs in the sets S_1, S_2 , demonstrating that all FBSs are satisfied.

Algorithm 3 Policy aware resource allocation of each cluster c_l .

```

input      :  $\Delta$ : Set of RBs,
             $\Delta'$ : Set of unallocated RBs,
             $S_1$ : Unsatisfied sorted list of
            FBSs with assigned policies,
             $S_2$ : Unsatisfied sorted list of
            FBSs without assigned policies
1 initialize  $\Delta' = \Delta$ 
2 while  $S_1 \neq \emptyset, S_2 \neq \emptyset$  do
3    $r = \text{select RB from } \Delta'$ 
4   if  $\exists f_i \in S_1$  so that  $r \in (\Delta' \cap range(f_{i,t}))$ 
5     then
6       assign RB  $r$  to  $f_i$ 
7       update set  $S_1$ 
8       sort set  $S_1$ 
9     end
10  if  $\exists f_j \in S_2$  then
11    assign RB  $r$  to  $f_j$ 
12    update set  $S_2$ 
13    sort set  $S_2$ 
14  end
15   $\Delta' = \Delta' - r$ 
16 end

```

6. Performance evaluation

In this section, we evaluate the performance of our proposed solution through extensive simulations under different scenarios and compare it by the state-of-the-art RA techniques in femtocell net-

works to understand its efficiency. We discuss the employed system parameters and study the obtained results in the performance study subsection.

6.1. System setup and parameters

With regard to the simulation study, all algorithms are implemented in the MATLAB version R2018b on a machine with 2.2 GHz Intel core i7 CPU and 16 GB of RAM.

We assume an environment in which FBSs are located according to the dual-strip model, discussed in section 3, so that each FBS has two users in its proximity. The channel model of 3GPP (2010) is used for the propagation environment so that the channel gain includes path-loss and shadowing. The transmission power and range of each FBS is supposed to be 13 dBm and 30 m, respectively. Besides, the FBS density is assumed to be $\lambda = 0.5$ and $\lambda = 1$, of which the first one represents a dense FBS network while the second one illustrates an ultra-dense FBS network. Moreover, the total number of available RBs in the network equals to 25, and the users' demands for different experiment scenarios vary between 1 and 4 RBs. We used MCS table of Mach and Becvar (2011), which has 12 different steps for three modulations including QPSK, 16-QAM, and 64-QAM. Table 3 summarizes evaluation parameters and their respective values.

6.2. Performance study

We employed four quantitative parameters including throughput, interference, fairness, and throughput satisfaction to comprehensively study the behavior of our proposed solution, called D^2C -FORAT, and to compare its efficiency with other solutions in the literature. We implemented QFCRA (Hatoum et al., 2014) and LFCRA (Qiu et al., 2016) which are distributed clustering proposals discussed in related work, and distributed random access (DRA) proposal (Sundaresan and Rangarajan, 2009). The DRA works based on a random selection of resources by each FBS and re-selection of interfered RBs by randomized hashing function. Each experiment is conducted for two different values of λ , and the outcomes are the average of 200 runs.

Table 3
Evaluation parameters.

Evaluation Parameters	Value
Carrier frequency	2 GHz
Bandwidth	5 MHz
No. of available RBs	25
No. of sub-carrier per RB	12
Bandwidth per sub-carrier	15 KHz
Bandwidth per RB	180 KHz
Path loss model	3GPP TR 36.814
FBS transmitted power	13 dBm
Apartment dimension	10 m × 10 m
FBS radius	30 m
Minimum separation between end users and FBS	2 m
User demand	1-4 RBs
FBS density	$\lambda = 0.5$ (20 active FBS), $\lambda = 1$ (40 active FBS)
Number of end users per FBS	2
MCS	QPSK (1/3,1/2,2/3,3/4), 16-QAM (1/2,2/3,3/4,4/5), 64-QAM (2/3,3/4,4/5)
Variance of AWGN	$\sigma^2 = -174$ dBm/Hz

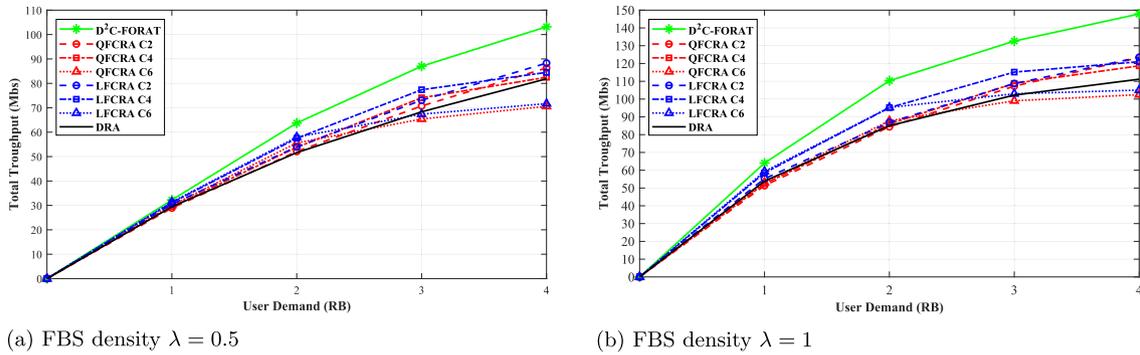


Fig. 5. Total throughput analysis using different FBS density λ and users' demands.

6.2.1. Throughput analysis

The total throughput of each technique is calculated based on equation (4) which represents the total throughput of all users in the network. Fig. 5 illustrates the total throughput of D^2C -FORAT and its counterparts for different values of λ . As it can be seen from Fig. 5a and b, the throughput of all techniques increases as the users' demands grow, while the growth rate decreases in higher demands due to increased interference. Besides, the D^2C -FORAT outperforms its counterparts by the maximum of 17% ($\lambda = 0.5$) and 21% ($\lambda = 1$) compared to the second-best technique. This improvement is the result of policies enacted for edge FBSs and our dynamic cluster size, resulting in better RA. The throughput of the QFCRA and the LFCRA heavily

depends on their cluster size, to the extent that their throughput falls below the DRA when their cluster size is 6.

6.2.2. Interference analysis

The interference between FBSs occurs whenever the overlapping FBSs use the same RBs simultaneously. Considering the SINR in the interfered RBs, the interference can be so weak, by which the throughput on those RBs does not decrease, or it can be so high, which results in unusable RBs. As the FBS density and users' demands increase, this problem occurs more often which has significant negative impact on the total network throughput. In this work, we consider RBs on which the throughput is less than R_{max} as interfered RBs. Based on the interfer-

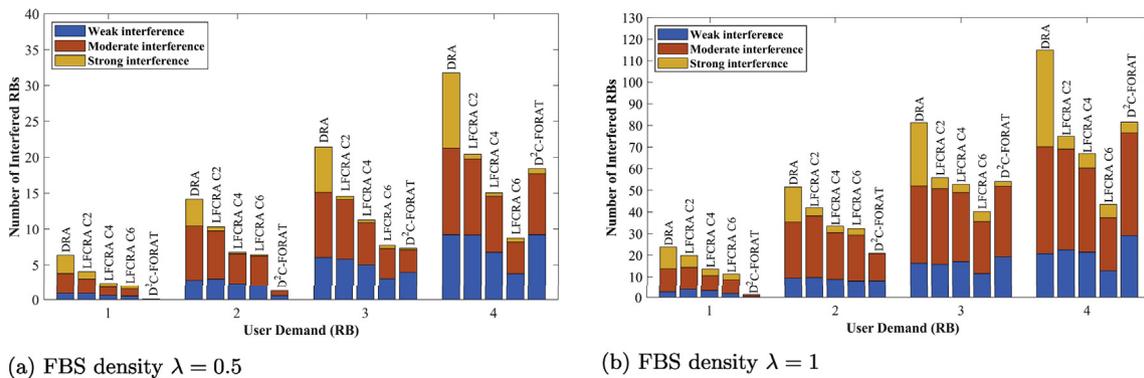


Fig. 6. Interference analysis using different FBS density λ and users' demands with three different interference levels including Weak, Moderate, and Strong.

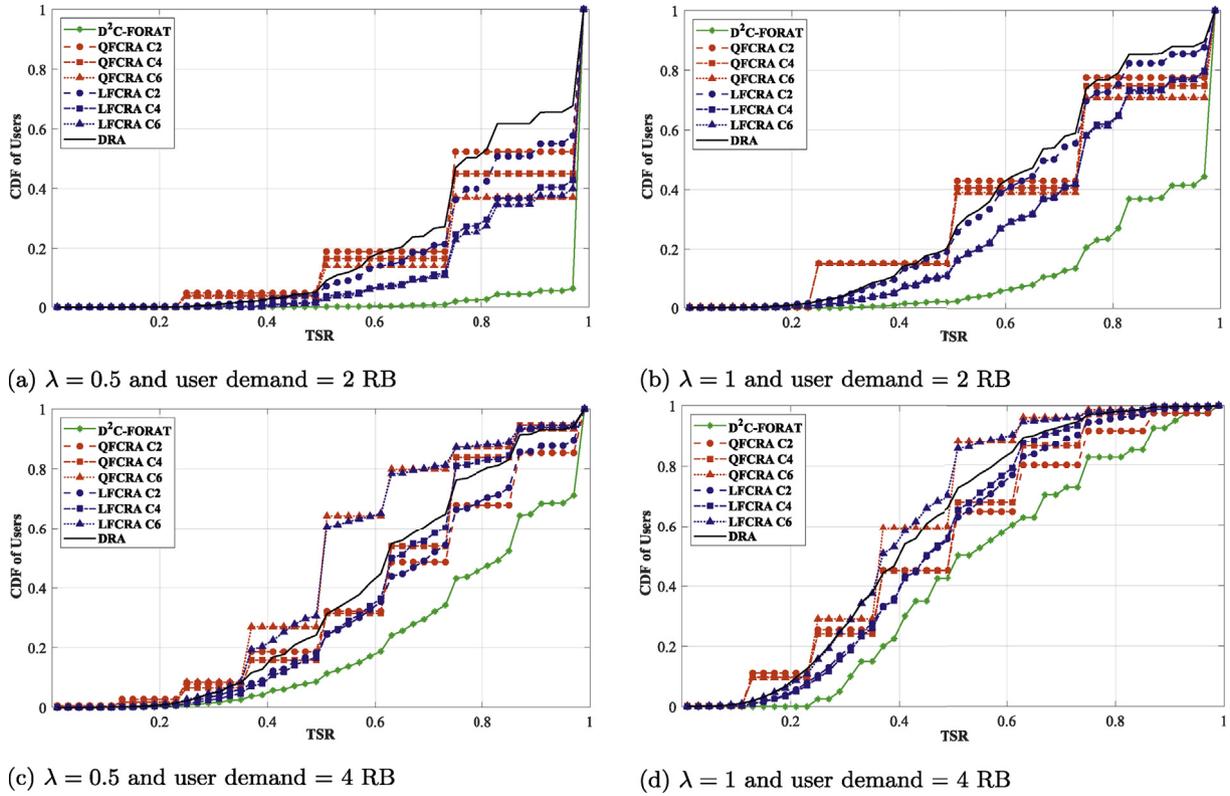


Fig. 7. Throughput Satisfaction Rate (TSR) using different FBS density λ and users' demands.

ence intensity, the interfered RBs can be divided into three categories including strong, moderate, and weak, for which the QPSK, 16-QAM, and 64-QAM are respectively selected as the MCS.

Fig. 6 represents obtained results for the number of interfered RBs with their corresponding categories. The QFCRA follows a conservative approach, and if any interference occurs between two FBSs, one of them is prevented of using that RB which results in no interference whenever the system is converged. However, this brings about several issues such as less throughput due to smaller RB reuse, as shown in the throughput analysis. As it can be seen from Fig. 6a and b, as the users' demands and FBS density increase, the number of interfered RBs increases. These results show that the DRA, due to its intrinsic random behavior and lack of coordination between FBSs, suffers from high number of interfered RBs, so that number of RBs that experiences strong interference is also more than its counterparts. The performance of LFCRA heavily depends on its cluster size, so that smaller cluster size incurs more interference and larger cluster size results in reduced number of interfered RBs. In contrary to LFCRA, as long as unallocated RBs are available, the D^2C -FORAT achieves minimum number of interfered RBs compared to its counterparts (Fig. 6a: users' demands 1 to 3 RB, and Fig. 6b: users' demands: 1 to 2 RBs), while it accepts some interference if throughput is satisfying. Besides, it is worth mentioning that the D^2C -FORAT obtains minimum number of strong interference which has the most negative effect on the total throughput.

6.2.3. Throughput satisfaction analysis

The Throughput Satisfaction Rate (TSR) is a quantitative parameter demonstrating the satisfaction degree of each user. The TSR is defined as the ratio of actual data rate of one user to its requested data rate, as depicted in the following.

$$TSR(u) = \frac{\sum_{k \in \Delta} a_{u,k}^f \times R_{u,k}^f}{\text{demand}_u \times R_{\max}}, \quad \forall u \in \mathcal{U}_f, \forall f_i \in \mathcal{F} \quad (21)$$

Fig. 7 represents the Cumulative Distributed Function (CDF) of the TSR for different values of λ and users' demands. As it can be observed, increasing users' demands in all techniques leads to less satisfaction, however, the D^2C -FORAT still outperforms its counterparts. This latter is because our technique dynamically controls cluster size, so that requested RBs can be completely assigned to end users. Also, it mitigates the interference on those RBs by the policy aware RA technique. In scenarios in which the number of RBs is greater than users' demands, the clustering techniques with larger cluster size incur less interference and better performance, however, as the users' demands increase the CHs are obliged to distribute the RBs between more users, and hence, the TSR significantly decreases. This latter can be observed in Fig. 7 when users' demands increase from 2 RB (Fig. 7a and b), in which techniques with larger cluster size are more efficient in terms of the TSR, to 4 RB (Fig. 7c and d) in which techniques with smaller cluster size can better satisfy the end users. Fig. 7a denotes that more than 95% of the end users have their TSR greater than 0.95 for D^2C -FORAT, which achieves 51% improvement compared to second-best technique. Fig. 7b represents that 59% of end users have the TSR greater than 0.95, which improves the second-best technique by 96%. Fig. 7c and d depict the TSR results whenever users' demands are 4 RB, in which the D^2C -FORAT achieves 0.8 user satisfaction for more than 55% and 18% of end users, respectively. These results demonstrate that our technique improves second-best techniques in Fig. 7c and d by 67% and 97%, respectively.

6.2.4. Fairness analysis

The Jain fairness index (Jain et al., 1984) is used to evaluate how fairly RBs are allocated between different end users, as expressed in the following:

$$\text{Fairness} = \frac{(\sum_{u=1}^N TSR(u))^2}{N \times \sum_{u=1}^N TSR(u)^2} \quad (22)$$

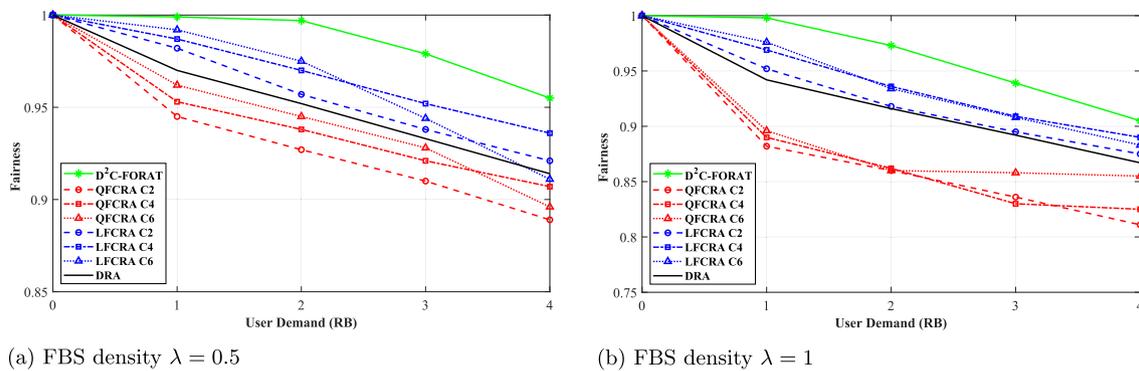


Fig. 8. Fairness analysis using different FBS density λ and users' demands.

where N represents the total number of end users in the system, and the maximum value for fairness is equal to 1 when all RBs are fairly allocated between end users.

As it can be seen from Fig. 8, as the users' demands and the FBS density increase, the fairness decreases due to increased number of interfered RBs. However, the D^2C -FORAT outperforms other techniques due to the policies enacted for edge FBSs and dynamic clustering that considers users' demands for controlling cluster size. Besides, the LFCRA obtains better results compared to the QFCRA because it improves the management of inter-cluster interference, which results in less interference for the edge FBSs. Furthermore, the DRA outperforms the QFCRA, because all end users in the DRA receive RBs either interfered or non-interfered ones, while the QFCRA attempts to assign only non-interfered RBs to end users which results in unsatisfied end users, specifically end users of the edge FBSs.

7. Conclusions and future work

In this work, we proposed a distributed dynamic clustering-fog driven RA technique, called D^2C -FORAT, to address the interference problem of femtocell networks, and to increase the total network throughput. Moreover, we used a hierarchical architecture, including the GW, fog servers, CHs, and CMs, among which the clustering and RA responsibilities are distributed. This latter results in better scalability, helping our technique to be efficiently run in sparse, dense, and ultra-dense networks. We proposed a distributed dynamic clustering method, in which FBSs select CHs, which are responsible to manage their corresponding cluster size based on the total demands of their CMs and available RBs. Moreover, each CH is responsible to allocating the cluster RBs and notifying its corresponding fog server of the cluster's parameters. Considering the fact that the inter-cluster interference is a big issue in clustering techniques, which decreases the total throughput of network, we proposed a policy aware fog-driven RA method to reduce such interferences. This method has three phases including graph formation, simplification, and relaxation which are performed on the fog servers located at the proximity of clusters. The outcome of these phases is a set of policies for edge FBSs of each cluster, by which the CH can assign the RBs more efficiently and prevent the severe inter-cluster interference. The effectiveness of the D^2C -FORAT is analyzed through extensive experiments and comparison by the state-of-the-art techniques in the literature. The obtained results demonstrate that our proposed solution outperforms other existing techniques in terms of total network throughput, interference, user satisfaction, and fairness.

In future, we aim to add virtualization concept to our technique as it helps to use resources more efficiently. Moreover, our plan is to exploit the computing capabilities of FBSs in the proposed hierarchical architecture, so that it jointly optimizes allocation of radio and computing resources.

Conflict of interest

No conflict of interest.

Acknowledgments

We thank Sara Kardani Moghaddam for her comments on improving this paper. We also thank anonymous reviewers and editors for their valuable feedback in enhancing the quality of our paper.

References

- 3GPP, Jan 22, 2015. Further advancements for e-utra physical layer aspects Evolved universal terrestrial radio access (e-utra): Further advancements for e-utra physical layer aspects. TR 36.814.
- Abdelnasser, A., Hossain, E., Kim, D.I., 2014. Clustering and resource allocation for dense femtocells in a two-tier cellular ofdma network. *IEEE Trans. Wirel. Commun.* 13 (3), 1628–1641.
- Bu, S., Yu, F.R., Yanikomeroglu, H., 2015. Interference-aware energy-efficient resource allocation for ofdma-based heterogeneous networks with incomplete channel state information. *IEEE Trans. Veh. Technol.* 64 (3), 1036–1050.
- Capozzi, F., Piro, G., Grieco, L.A., Boggia, G., Camarda, P., 2013. Downlink packet scheduling in lte cellular networks: key design issues and a survey. *IEEE Commun. Surv. Tutor.* 15 (2), 678–700.
- Chang, C., Srirama, S.N., Buyya, R., 2019. Internet of Things (IoT) and New Computing Paradigms, Fog and Edge Computing: Principles and Paradigms, pp. 1–23.
- Fu, F., Lu, Z., Xie, Y., Jing, W., Wen, X., 2017. Clustering-based low-complexity resource allocation in two-tier femtocell networks with qos provisioning. *Int. J. Commun. Syst.* 30 (4), e3005.
- Garcia-Morales, J., Femenias, G., Riera-Palou, F., 2015. Analytical performance evaluation of ofdma-based heterogeneous cellular networks using ffr. In: *Proceedings of 81th IEEE Vehicular Technology Conference (VTC Spring)*. IEEE, pp. 1–7.
- Goudarzi, M., Zamani, M., Toroghi Haghghat, A., 2016. A genetic based decision algorithm for multisite computation offloading in mobile cloud computing. *Int. J. Commun. Syst.*, <https://doi.org/10.1002/dac.3241>.
- Goudarzi, M., Zamani, M., Haghghat, A.T., 2017. A fast hybrid multi-site computation offloading for mobile cloud computing. *J. Netw. Comput. Appl.* 80, 219–231.
- Ha, V.N., Le, L.B., 2014. Fair resource allocation for ofdma femtocell networks with macrocell protection. *IEEE Trans. Veh. Technol.* 63 (3), 1388–1401.
- Hatoum, A., Langar, R., Aitsaadi, N., Boutaba, R., Pujolle, G., 2014. Cluster-based resource management in ofdma femtocell networks with qos guarantees. *IEEE Trans. Veh. Technol.* 63 (5), 2378–2391.
- Hu, P., Dhelim, S., Ning, H., Qiu, T., 2017. Survey on fog computing: architecture, key technologies, applications and open issues. *J. Netw. Comput. Appl.* 98, 27–42.
- Jain, R.K., Chiu, D.-M.W., Hawe, W.R., 1984. A Quantitative Measure of Fairness and Discrimination. Eastern Research Laboratory, Digital Equipment Corporation, Hudson, MA, USA, pp. 2–7.
- Kim, J., Cho, D.-H., 2010. A joint power and subchannel allocation scheme maximizing system capacity in indoor dense mobile communication systems. *IEEE Trans. Veh. Technol.* 59 (9), 4340–4353.
- Le, N.-T., Jayalath, D., Coetzee, J., 2018. Spectral-efficient resource allocation for mixed services in ofdma-based 5g heterogeneous networks. *Trans. Emerg. Telecommun. Technol.* 29 (1), e3267.
- Lee, Y.L., Chuah, T.C., Loo, J., Vinel, A., 2014. Recent advances in radio resource management for heterogeneous lte/lte-a networks. *IEEE Commun. Surv. Tutor.* 16 (4), 2142–2180.
- Lee, Y.L., Loo, J., Chuah, T.C., El-Saleh, A.A., 2017. Multi-objective resource allocation for lte/lte-a femtocell/henb networks using ant colony optimization. *Wirel. Pers. Commun.* 92 (2), 565–586.
- Li, W., Zhang, J., 2018. Cluster-based resource allocation scheme with qos guarantee in ultra-dense networks. *IET Commun.* 12 (7), 861–867.

- Li, L., Zhou, Z., 2017. Cluster based resource allocation in two-tier hetnets with hierarchical throughput constraints. *Int. J. Commun. Syst.* 30 (14), e3292.
- Li, W., Zheng, W., Wen, X., Su, T., 2012. Dynamic clustering based sub-band allocation in dense femtocell environments. In: *Proceedings of 75th IEEE Vehicular Technology Conference (VTC Spring)*. IEEE, pp. 1–5.
- Mach, P., Becvar, Z., 2011. Qos-guaranteed power control mechanism based on the frame utilization for femtocells. *EURASIP J. Wirel. Commun. Netw.* 2011 (1), 253–259.
- Mhiri, F., Sethom, K., Bouallegue, R., 2013. A survey on interference management techniques in femtocell self-organizing networks. *J. Netw. Comput. Appl.* 36 (1), 58–65.
- Pitman, J., 1997. Some probabilistic aspects of set partitions. *Am. Math. Mon.* 104 (3), 201–209.
- Pratap, A., Singhal, R., Misra, R., Das, S.K., 2018. Distributed randomized k -clustering based pcid assignment for ultra-dense femtocellular networks. *IEEE Trans. Parallel Distrib. Syst.* 29 (6), 1247–1260.
- Qiu, J., Ding, G., Wu, Q., Qian, Z., Tsiftsis, T.A., Du, Z., Sun, Y., 2016. Hierarchical resource allocation framework for hyper-dense small cell networks. *IEEE Access* 4, 8657–8669.
- Rohoden, K., Estrada, R., Otrók, H., Dziong, Z., 2019. Stable femtocells cluster formation and resource allocation based on cooperative game theory. *Comput. Commun.* 134, 30–41.
- Sundaresan, K., Rangarajan, S., 2009. Efficient resource management in ofdma femto cells. In: *Proceedings of the Tenth ACM International Symposium on Mobile Ad Hoc Networking and Computing*. ACM, pp. 33–42.
- Tan, L., Feng, Z., Li, W., Jing, Z., Gulliver, T.A., 2011. Graph coloring based spectrum allocation for femtocell downlink interference mitigation. In: *Proceedings of Wireless Communications and Networking Conference (WCNC)*. IEEE, pp. 1248–1252.
- Zhang, Q., Zhu, X., Wu, L., Sandrasegaran, K., 2013. A coloring-based resource allocation for ofdma femtocell networks. In: *Proceeding of Wireless Communications and Networking Conference (WCNC)*. IEEE, pp. 673–678.
- Zhou, L., Hu, X., Ngai, E.C.-H., Zhao, H., Wang, S., Wei, J., Leung, V.C., 2016. A dynamic graph-based scheduling and interference coordination approach in heterogeneous cellular networks. *IEEE Trans. Veh. Technol.* 65 (5), 3735–3748.



Mohammad Goudarzi graduated from Iran University of Science and Technology (IUST), Tehran, Iran, with First-Class Honors degree in M.Sc. in Information Technology (Computer Networking), where he was selected as the exceptional talented student as well. Due to his academic achievements, he was awarded to become a member of Iranian National Elites Foundation, a prestigious organization for recognition and support of Iranian national elites, from which he received a prestigious research Grant. He is working towards the Ph.D. degree at the Cloud Computing and Distributed Systems (CLOUDS) Laboratory, Department of Computing and Information Systems, the University of Melbourne, Australia. He was awarded the Melbourne International Research Scholarship (MIRS) supporting his studies. Besides, he was awarded the Rowden White Scholarship, a prestigious scholarship provided by the University of Melbourne to talented, high quality PhD students. His research interests include Internet of Things (IoT), Fog Computing, Wireless Networks, and Optimization.



Marimuthu Palaniswami (F'12) received the M.E. degree in electrical, electronic, and control engineering from the Indian Institute of Science, Bengaluru, India, in 1979, the M.Eng.Sc. degree in electrical, electronic, and control engineering from The University of Melbourne, Melbourne, VIC, Australia, in 1983, and the Ph.D. degree from The University of Newcastle, Callaghan, NSW, Australia, in 1987. He is currently a Professor in electrical engineering and the Director/Convener of a large ARC Research Network on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP) with about 100 researchers on various interdisciplinary projects. He is representing Australia as a core partner in EU FP7 projects, such as SENSEI, SmartSantander, Internet of Things Initiative, and SocloTal. He is the author or coauthor of more than 480 refereed research papers and leads one of the largest funded ARC Research Network on ISSNIP. His research interests include smart sensors and sensor networks, machine learning, IoT, and biomedical engineering and control.



Rajkumar Buyya is a Redmond Barry Distinguished Professor and Director of the Cloud Computing and Distributed Systems (CLOUDS) Laboratory at the University of Melbourne, Australia. He is also serving as the founding CEO of Manjrasoft, a spin-off company of the University, commercializing its innovations in Cloud Computing. He has authored over 650 publications and seven text books including “Mastering Cloud Computing” published by McGraw Hill, China Machine Press, and Morgan Kaufmann for Indian, Chinese and international markets respectively. Dr. Buyya is one of the highly cited authors in computer science and software engineering worldwide (h-index = 124, g-index = 271, 79,100 + citations). He is named in the recent Clarivate Analytics’ (formerly Thomson Reuters) Highly Cited Researchers and “World’s Most Influential Scientific Minds” for three consecutive years since 2016. Dr. Buyya is recognized as Scopus Researcher of the Year 2017 with Excellence in Innovative Research Award by Elsevier for his outstanding contributions to Cloud computing. Software technologies for Grid, Cloud, and Fog computing developed under Dr. Buyya’s leadership have gained rapid acceptance and are in use at several academic institutions and commercial enterprises in 40 countries around the world. Manjrasoft’s Aneka Cloud technology developed under his leadership has received “2010 Frost & Sullivan New Product Innovation Award”. He served as founding Editor-in-Chief of the IEEE Transactions on Cloud Computing. He is currently serving as Editor-in-Chief of Software: Practice and Experience, a long standing journal in the field established ~50 years ago. For further information on Dr. Buyya, please visit his cyberhome: www.buyya.com.