Low Complexity Node Clustering in Cloud-RAN for Service Provisioning and Resource Allocation

Haining Wang, Priyesh Shetty, and Zhi Ding,

***Abstract*—Auction-based service provisioning and resource allocation have demonstrated strong potential in Cloud-RAN wireless network architecture and heterogeneous networks for effective resource sharing. One major technical challenge is the integration of interference constraints in auction-based solutions. In this work we transform the interference constraint requirement into a set of linear constraints on each cluster. We tackle the generally NP-hard clustering problem by developing a novel practical suboptimal solution that can meet our design requirement. Our novel algorithm utilizes the properties of chordal graphs and applies Lexicographic Breadth First Search (Lex-BFS) algorithm for cluster splitting. This polynomial time approximate algorithm searches for maximal cliques in a graph by generating strong performance in terms of subgraph density and probability of optimal clustering without suffering from the high complexity of the optimal solution.**

***Index Terms*—auction design, clustering algorithm, graph theory, chordal graphs, maximal clique, polynomial time algorithm**

I. Introduction

Heterogeneous Networks (HetNets), through coordinated coverage and resource sharing among macro cells and small cells of various sizes, have demonstrated strong promises in expanding network capacity of wireles networks including 4G and the futuristic 5G. However, one bottleneck lies in the limited backhaul speed for performing traffic balancing, radio resource management, and interference mitigation. Hence, in developing advanced wireless networks, the Cloud Radio Access Network (C-RAN) architecture is increasingly gaining traction [1]. In C-RAN, we have a centralized Baseband Unit (BBU) formed by pooling remote radio head (RRH) units from their respective base-stations (BS) for muliplexing gain [2]. We shall use RRH and BS interchangeabley since RRH can be viewed as a simple BS. In HetNets, the coordinated RRHs (BSs) are connected to the BBU for centralized signal processing and resource control [3].

Resource management for heterogeneous BSs is key to achieving the promised gains in practice. Existing works include a joint sub-channel assignment and power control scheme in an OFDMA-based network [4], a dynamic user association framework [5] that silences small-cellgroups to mitigate inter-cell interference. Network-utility maximization can also handle user association, power allocation, and data off-loading [6–9].

This material is based upon work supported by the National Science Foundation under Grants CNS-1443870 and CNS-1702752.

H. Wang, P. Shetty and Z. Ding are with Department of Electrical and Computer Engineering, University of California, Davis, California, 95616 (email:{hnwang, pshetty,zding}@ucdavis.edu)

Auction design in wireless networks can achieve joint downlink UE (use equipment) association and resource allocation for a centralized BBU. For example, in [10], a near optimal algorithm was presented. However, these frameworks must incorporate interference constraints within the auction design. In general, interference constrained auction design is a very challenging problem. To avoid strong inter-cell interference of auction based heterogeneous networks, our approach is to divide the associated BSs into clusters. BSs within a cluster are sensitive to interference constraints and cannot share spectrum resources. In this paper, we utilize graph theory [11] by convert these interference constraints into a set of linear constraints on each BS cluster, thereby facilitating subsequent auction-based UE association and resource allocation. Existing algorithms are based on finding maximal cliques in the graph. We also aim to approximate exponential time optimal algorithm for finding cliques in a graph with near-optimal polynomial time algorithm.

Practically, clique enumeration for any optimal algorithm has worst-case time complexity $O(3^{N/3})$ [12], which is too complex even for moderate sized network. Very few approximation algorithms are available in the literature without substantial performance loss. One example in [13] is the proposal of a ''super-cliques" method. However, since the interference graph in our problem is more restricted, the algorithm does not apply. The algorithms proposed in [14] can also be viewed as two polynomial-time approximation algorithms, yielding the *Node-ALL* interference constraints and *Node-L* interference constraints. As discussed in Section III, we propose improved polynomial time algorithms to approximately achieve suboptimal clustering algorithms to satisfy the interference constraints.

In this work, we propose a suboptimal clusteirng algorithm Alg.2 before cluster splitting with Alg.3. In our proposed approach, we utilize the properties of chordal graphs since most algorithms, though NP-hard for general graphs, can be solved in polynomial time for chordal graphs. We use lexicographic Breadth First Search (Lex-BFS) algorithm [15] [16] to determine the maximal clique for a chordal graph, since Lex-BFS returns maximal clique if the graph is chordal. The subsequent channel assignment from Alg.1 can henceforce take into account the interference constraints in auction design. Our results illustrate that our algorithm performs better than the existing algorithms in terms of optimal clustering probability and graph density of induced subgraphs. Furthermore, our algorithm has polynomial time complexity $O(N^3)$ which is similar to the complexity of several existing algorithms.

978-1-5090-5019-2/17/$31.00 ©2017 IEEE

Our manuscript is organized as follows. Section II presents the problem formulation of interference constraints and establishes the equivalence to the linear constraints on station clusters. Section III describes the fundamentals of existing polynomial time algorithms and proposes a new low-complexity suboptimal algorithm for clustering based on chordal cliques. We demonstrate important properties of our approximate algorithm in Section IV and test the performance of our algorithm and its complexity in Section V-A. Section VI concludes this paper.

II. Problem Formulation

We now formulate the BS clustering problem by transforming the important interference constraints in resource allocation to a set of linear constraints on each BS cluster. To begin, we list the following notations and symbols that are needed to capture the heterogeneous resource allocation problem and the associated BS clustering problem.

TABLE I: Notations for Problem Formulation

M	Number of UEs (User Equipments)
N	Number of BSs
$\mathcal{B}$	Set of BSs
$\mathcal{M}$	Set of UEs
N_b	Number of available channels
N_U	Maximal number of channels that can be allocated to one UE
r_m	Rate requirement for UE-m
p_{mi}	Transmission power of BS-i for UE-m.
v_{mi}	Valuation of the link between BS-i and UE-m
v_i	Valuation of the ith BS
a_{mi}	Link association indicator between BS-i and UE-m
b_{mi}	The bid for the link between BS-i and UE-m
n_m	Number of channel required for UE-m.
$\mathcal{C}_l$	lth BS cluster
$\mathcal{L}$	Set of all clusters
L	Number of clusters obtained by executing suboptimal clustering algorithm

A. Interference Constraints and Clustering

First, within a heterogenous network, UE-m with rate requirement r_m can be served by BS-i if there is link association ($a_{mi} = 1$) using power P_{mi}. By bidding price b_{mi}, BS-i serves UE-m and receives a value v_{mi}. The important constraints in such spectrum auction is to enforce the (low) interference constraints. Based on the submitted bids, the BBU has the following objective:

$$\max_{\boldsymbol{a}\in\mathcal{A}} \quad \sum_{m\in\mathcal{M}}\sum_{i\in\mathcal{B}} a_{mi}b_{mi} \tag{1a}$$

subject to interference constraints (1b)

$$\sum_{i,j\in\mathcal{B},i\neq j}\ \sum_{m\in\mathcal{M}} (a_{mi}+a_{mj})n_m \leq N_b,$$

$$\forall i,\ j \text{ with Distance}(B_i, B_j) \leq r_i + r_j \tag{1c}$$

$$\text{and} \sum_{i\in\mathcal{B}} a_{mi} \leq 1 \quad a_{mi}\in\{0,1\}. \tag{1d}$$

The interference constraints state that, when the distance between B_i and B_j is smaller than the sum of their interference radii τ_i and τ_j, then the two BSs must not share any channels when serving UE-m.

In this work, we study the problem of finding the set of clusters such that the interference constraints can be converted to a series of linear constraints. For better description of our problem, we apply the following definitions in graph theory [11].

Definition 1 (Clique)**.** A clique in an undirected graph is a subset of its vertices such that every two vertices in the subset are connected pairwise by an edge.

Definition 2 (Maximal Clique)**.** A maximal clique is a clique that is not fully contained in a larger clique.

The goal of clustering BSs is to identify BS clusters, denoted as $\{\mathcal{C}_l, l\in\mathcal{L}\}$ with $\mathcal{L}$ being the set of indices of all clusters such that BSs within the same cluster can not share any spectral components, also knwon as resource blocks (or RBs) in OFDMA. In terms of constraints, this is equivalent to

$$\sum_{i\in\mathcal{C}_l}\sum_{m\in\mathcal{M}} n_m a_{mi} \leq N_b, \qquad \forall l\in\mathcal{L}, \tag{2}$$

where N_b is the number of channels available at the cloud and a_{mi} is some feasible assignment satisfying $\sum_{i\in\mathcal{B}} a_{mi} \leq 1$ and $a_{mi}\in\{0,1\}$.

The need for an effective algorithm is illustrated by the two trivial cases: A single cluster of all BSs versus each BS forming its own cluster. If there is a single cluster of all base stations, then no two BSs can share any RB, leading to excessive requirement of bandwidth that is likely to exceed the overall availability N_b. If each BS forms a cluster on its own, then interference among any interference BSs is neglected. As a result, common RB assigned to interference BSs within close proximity will lead to strong interference and failed network links. Thus an optimal clustering strategy is to achieve sufficient number of clusters to utilize N_b resource blocks without mutual interference among BSs Dist(B_i, B_j) $\leq r_i + r_j$.

B. Equivalent Maximal Clique Enumeration Problem

Before designing algorithms for clustering, we first provide the theoretical analysis to link our problem to the famous maximal clique enumeration problem in graph theory given by the following proposition.

Proposition 1. *Let $\{\mathcal{C}_l^\star, l\in\mathcal{L}^\star\}$ be the set of all maximal cliques for the interference-conflict graph $G(\mathcal{B}, E)$. Then the interference constraint* (1c) *is equivalent to the following linear constraints*

$$\sum_{i\in\mathcal{C}_l^\star}\sum_{m\in\mathcal{M}} n_m a_{mi} \leq N_b, \qquad \forall l\in\mathcal{L}^\star. \tag{3}$$

Furthermore, given any feasible association decision $\boldsymbol{a}\in\mathcal{A}$ satisfying (3)*, there exists a feasible resource allocation strategy for each UE-m.*

Proof. To prove (3) as both necessary and sufficient in guaranteeing interference-free channel assignment, denote the set of all feasible assignments $\boldsymbol{a} \in \mathcal{A}$ specified by constraints (1c) as $\mathcal{P}^\star$ and that by the constraints formed by replacing (1c) with (3) as $\mathcal{P}_0$. Our equivalence proof must show two parts: 1) $\mathcal{P}^\star \subseteq \mathcal{P}_0$ and 2) $\mathcal{P}_0 \subseteq \mathcal{P}^\star$.

The first part follows readily from the definition of a clique. Since two end nodes of an edge cannot share any RBs and all nodes are connected with other nodes in a clique, all UEs allocated to BS-i with $i \in \mathcal{C}_l^\star$ need to be assigned non-overlapping RBs. Therefore, any point in $\mathcal{P}^\star$ needs to at least satisfy (3), which implies that $\mathcal{P}^\star \subseteq \mathcal{P}_0$.

For the second part, we can show that for $\forall \boldsymbol{a} \in \mathcal{P}_0$, we can construct a feasible resource allocation for all admitted UEs by sequentially assigning each admitted UE-m n_m channels from $|\mathcal{L}|$ containers, each storing available resources for one cluster of BSs. For each $a_{mi} = 1$, we fetch n_m resources of the same indices from all containers l with $i \in \mathcal{C}_l$. The algorithm is summarized in Alg. 1. Due to (3), any two users allocated to adjacent BSs should be allocated orthogonal resources and we can never have starving UE with $a_{mi} = 1$ in $\mathcal{P}_0$.

Note that we only utilized the clique property for each cluster, the reason is that any non-maximal clique is always a subset of some maximal clique and the corresponding constraints would be redundant. $\square$

Algorithm 1 Channel Assignment

Initialize a set $\mathcal{N}_l = \mathcal{N}_b$ for each $l \in \mathcal{L}$.
for each $i \in \mathcal{B}$ **do**
  Find all cluster indices l with $i \in \mathcal{C}_l$, denoted as $\mathcal{I}_i$
end for
for each $i \in \mathcal{B}$ **do**
  for each m with $a_{mi} = 0$ **do**
    Pick arbitrary n_m channels from the set $\cap_{l \in \mathcal{I}_i} \mathcal{N}_l$ for assignment to UE-m.
    Remove the channels assigned to UE-m from each cluster $l \in \mathcal{I}_i$.
  end for
end for

III. Polynomial-time Approximations

Proposition 1 effectively transforms the interference constraint (1c) by (3) based on maximal clique enumeration. The maximal-clique-enumeration problem, however, is NP-hard except for some special graphs. In fact, for general graphs, there may be an exponential number of maximal cliques and the best worst-time complexity of any optimal algorithm is shown to be $O(3^{N/3})$ [12], which is high even for a moderate-size network. Moreover, the goal of clustering in our particular application is to form linear constraints. Finding all maximal cliques could be an unnecessary over-kill as it may complicate our subsequent optimization procedure by including exceedingly large number of constraints. Thus, for practical reasons of transforming interference constraints for auction design, we would like to develop practical sub-optimal algorithms that at least satisfies the following properties:

1) It generates clusters with dense[1] induced subgraphs.
2) Any user association satisfying constraints formed by the sub-optimal clusters $\{\mathcal{C}_l, \quad l \in \mathcal{L}\}$ has feasible interference-free resource allocation schemes.
3) The number of constraints, or equivalently, the number of clusters, is polynomial in the number of nodes.

A. Several Known Suboptimal Methods

Only a few approximation algorithms available in the literature can meet all three properties. One method [13] proposes a polynomial time algorithm to approximate an exponential number of maximal cliques with a polynomial-number of 'super-cliques', which is possibly the union of several nearby cliques. The algorithm itself, however, does not apply to our scenario due to a more restricted interference graph. The algorithms proposed in [14] can also be viewed as two possible polynomial-time approximation algorithms, yielding the *Node-ALL* interference constraints and *Node-L* interference constraints.

- *Node-ALL* constraints are formed by clustering each node with all its neighbors. Mathematically, this is equivalent to forming constraints
$$\sum_{j\in\{\text{Neighbors of BS}-i\}} \sum_{m\in\mathcal{M}} n_m a_{mj} \le N_b, \quad \forall i \in \mathcal{B}, \tag{4}$$
where {neighbors of BS-i} includes BS-i itself and all BSs that are connected with BS-i through some edge.
- *Node-L* strategy, on the other hand, only clusters each node with neighbors on the left of the node. Equivalently, we have
$$\sum_{j\in\{\text{Left neighbors of BS}-i\}} \sum_{m\in\mathcal{M}} n_m a_{mj} \le N_b, \quad \forall i \in \mathcal{B}, \tag{5}$$
where {left neighbors of BS-i} contains itself and all BSs that are geometrically *not* to the right of BS-i but are connected with BS-j through some edge. Thus, for both strategies, we will form exactly N clusters including either the neighbors of each BS-i or all left neighbors of each BS-i.

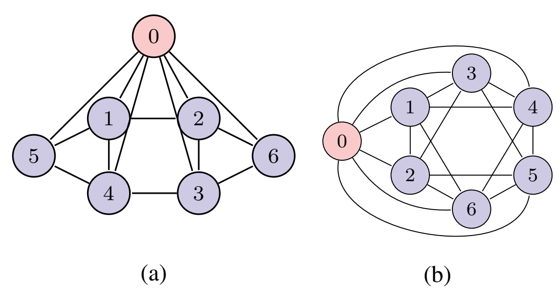


Fig. 1: Examples for clustering illustration.

Because of heterogeneous capabilities among different nodes, neither algorithm can yield a good approximation of

[1]For undirected simple graphs, the density is defined as the ratio of the number of edges over the maximum number of edges for a complete graph. A dense graph has number of edges close to the maximum number of edges.

the original problem. In HetNet, a BS with a higher transmit power is called a macro-cell base-station (MBS) and the one with a lower power is called a femtocell base-station (FBS). Let us consider two simple networks, each consisting of one MBS (node-0) and 6 FBSs within the coverage of the macrocell. As shown in Fig. 1, there is an edge between node 0 and each of the other nodes. In this case, the *Node-All* strategy clusters node 0 with all its neighbors and forms a super cluster as $\{0,1,2,3,4,5,6\}$ and all other clusters will be dominated and deleted, which implies that no BS can share any resources in this case. *Node-L* algorithm can perform better by clustering nodes with its "left neighbors". For example, by removing dominated sets, the clustering results for Fig. 1a and Fig. 1b are $\{\{0,1,4,5\},\{0,1,2,3\},\{0,2,3,4\},\{0,2,3,6\}\}$ and $\{\{0,1,2,3\}, \{0,1,2,6\},\{0,1,3,4,6\},\{0,2,3,4,5,6\}\}$, respectively. Although neither result is optimal, they allow some limited sharing of resources. The performance of the *Node-L* algorithm, however, may vary for equivalent interference graphs. For example, a horizontal flip of Fig. 1b puts nodes 1-6 to the left of node 0, then the *Node-L* algorithm trivially clusters all nodes.

B. A New Practical and Suboptimal Clustering

Motivated by the illustrated shorcomings of existing methods, we present a new sub-optimal algorithm to generate a higher quality polynomial number of super-cliques. To get started with our algorithm, we introduce another definition.

Definition 3 (Chordal Graph)**.** An undirected *chordal graph* is an undirected graph in which all cycles of four or more vertices have a chord, which is an edge that is not part of the cycle but connects two vertices of the cycle.

Since many NP-hard problems on general graphs are polynomial-time solvable for chordal graphs, we will use this to design our sub-optimal algorithms. Our suboptimal clustering procedure is designed as follows:

Algorithm 2 Sorting and Cluster Splitting (SACS) Algorithm

Step-1: Sort the nodes according to their degrees $d(\cdot)$ in the conflict graph. Let $v_1, v_2, \ldots$ denote the sorted BSs with $d(v_1) \geq d(v_2) \geq \cdots \geq d(v_N)$.

Step-2: Let $\mathcal{C}$ denote the set of clusters initialized to be empty. For each node, form the cluster

$$\mathcal{C}_i = \{v_i\} \cup \{v_j \in \mathcal{B} | \exists \text{edge } (v_i, v_j), j < i\}.$$

Step-3: Check each cluster in $\mathcal{C}$ and try to split the cluster into smaller clusters if the subgraph induced by the cluster is not a clique, but is a chordal graph. Remove any dominated clusters if there is any split.

Intuitively, our SACS algorithm first forms a series of clusters based on an ordering of nodes in Step-1. The last step tries to split sub-optimal super-cliques into maximal cliques.

Let us revisit the clustering using our algorithm for the two examples in Fig. 1. For Fig.1a, one possible ordering of the nodes by Step-1 is $\{0,1,2,3,4,5,6\}$. The initial clustering from Step-2 is then given by $\{\{0,1,2\},\{0,2,3\},\{0,1,3,4\},\{0,1,4,5\},\{0,2,3,6\}\}$.

Step-3 finds that all clusters are cliques except for $\{0,1,3,4\}$. Since the induced subgraph is a chordal graph, it can be split into two maximal cliques $\{0,1,4\}$ and $\{0,3,4\}$. In fact, for this case, we have already found the optimal clustering. This follows from the properties of the clustering algorithm introduced later, and can also be verified by exhaustively listing all subgraphs to find all maximal cliques associated with it.

Alg. 2 does not always guarantee optimality. For example, in Fig. 1b, after Step-2, we will get clusters $\{\{0,1,2,3\},\{0,1,3,4\},\{0,2,3,4,5\}, \{0,1,2,4,5,6\}\}$, in which both $\{0,2,3,4,5\}$ and $\{0,1,2,4,5,6\}$ are not cliques. However, in this example, only the first cluster is a chordal graph and is splittable in Step-3 into $\{0,2,3\}$, $\{0,3,4\}$, $\{0,5\}$, each dominated by other clusters. Since cluster $\{0,1,2,4,5,6\}$ is flagged to be non-splittable in Step-3, we will have a sub-optimal solution. In fact, this topology is a special worst case scenario where we have an exponential number of cliques in the graph [13]. Therefore, any polynomial-time algorithm is expected to be sub-optimal in this case.

The algorithm is polynomial time in Step-1 and Step-2. Step-3 is a best effort approach to only split a non-clique cluster when the induced subgraph is a chordal graph. Since the detection and the split of a chordal graph both take polynomial time, Alg. 2 is a polynomial-time algorithm.

The procedure to perform a cluster split is given in Alg. 3. Due to limited space, we omit the pseudo-code for lexicographic Breadth First Search (Lex-BFS) and the detailed procedure of how the maximal cliques can be generated. Please refer to [15] [16] and book [17] for the pseudo-code or related discussions. By running the Lex-BFS, we can either decide the graph is non-chordal along the process or return all its maximal cliques after it is finished. This is because Lex-BFS returns a perfect elimination order for a chordal graph. To list all maximal cliques of a chordal graph, on can form a clique for each node v together with the neighbors of v that are later than v in the perfect elimination ordering, and test whether each of the resulting cliques is maximal.

Algorithm 3 Cluster Split

for all $\mathcal{C}_i \in \mathcal{C}$ **do**
  if $\mathcal{C}_i$ is not a maximal clique **then**
    Perform Lex-BFS on the induced subgraph of $\mathcal{C}_i$. (Lex-BFS can be used to check whether a graph is a chordal graph and to list all maximal cliques from a chordal graph [16]). If the subgraph is a chordal graph, by letting I be the number of maximal cliques, the set of maximal cliques can be denoted as $\{C'_{i1}, C'_{i2}, \ldots, C'_{iI}\}$.
    Replace $\mathcal{C}_i$ by $\{C'_{i1}, C'_{i2}, \ldots, C'_{iI}\}$ in $\mathcal{C}$ and remove any redundant set if the split is successful.
  end if
end for

IV. Properties of the SACS Algorithm

In this section, we will establish that the proposed SACS clustering algorithm generates a set of linear constraints that are stricter with the feasible region no larger than the original feasible region defined in (3). The property guarantees resource allocation feasibility in terms of interference avoidance as shown in the following proposition.

Proposition 2. *Let $\mathcal{C}$ be the clusters generated by Alg. 2. Then any maximal clique $\mathcal{C}_l^\star \in \mathcal{C}^\star$ of the same interference conflict graph is a subset of some cluster $\mathcal{C}_l \in \mathcal{C}$.*

Proof. To show this, it suffices to show that any maximal clique is a subset of some cluster in the cluster set obtained in Step-2. The reason is that any additional clusters obtained in Step-3 come from splitting existing clusters from Step-2 by successfully enumerating all possible maximal cliques in the induced chordal subgraph. With a small abuse of notation, we also denote the cluster set as $\mathcal{C}$ at the end of Step-2.

Step-1 returns an ordering of nodes and let v_{max} be the node of the highest label in $\mathcal{C}_l^\star$. Then we know that when node v_{max} enters, all nodes in $\mathcal{C}_l^\star$ will be included to form a cluster by the definition of a clique. Therefore, we know there has to exist some $\mathcal{C}_l$ with $\mathcal{C}_l^\star \subseteq \mathcal{C}_l$. □

Proposition 2 implies that the constraints

$$\sum_{i\in\mathcal{C}_l}\sum_{m\in\mathcal{M}} n_m a_{mi} \le N_b, \qquad \forall l \in \mathcal{L} \tag{6}$$

generated from the sub-optimal cluster set $\mathcal{C}$ are more restrictive than the optimal ones, including no infeasible association.

V. Numerical Results

A. Complexity Analysis

Step-1 has a complexity of $O(|E| + N\log(N))$ to get the degrees and sorting. Step-2 is $O(|E|)$ in complexity and the potential split of clusters takes at most $O(|E|+N)$. Removing dominated sets takes at most $O(N^3)$. Note that, even when dominated sets are not deleted from the cluster set, the feasible region stays the same. However, it is still preferable to remove redundant constraints for later optimization design.

Note that the complexity of both *Node-All* and *Node-L* algorithms is also dominated by the complexity of removing unnecessary clusters of complexity $O(N^3)$. Without this extra step, our proposed algorithm is only slightly more complex due to the sorting procedure, but the performance improvement is quite impressive in terms of both the induced graph densities[2] of clusters and optimality probabilities.

Since in *Node-ALL* or *Node-L* strategies, a node with a higher degree[3] is more likely to cluster with neighbors that belong to more than one maximal cliques, the intuition for the sorting in the first step is to reduce the possibility of making the cluster much bigger than a maximal clique in Step-2. The split in Step-3 is a best-effort strategy. The probability of getting chordal subgraphs is high when the subgraph size is small; therefore, the split is most effective if Step-1 and Step-2 can produce small clusters.

B. Performance and Comparison

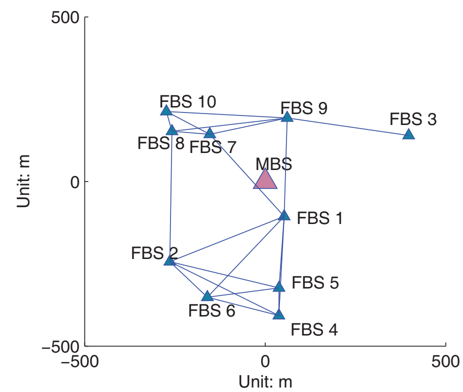


Fig. 2: Interference conflict graph of an example network with 10 FBS and 1 MBS (edges between MBS and any other FBS are omitted for clearer view.)

An example of interference conflict graph (omitting the edges from macro-BS to each femto-BS) can be seen in Fig.2. The value of d used in this graph is 200m. Using Alg.2, we can successfully find the optimal clustering given by $\{\{0,1,2,4,5,6\}, \{0,1,7,9\}, \{0,7,8,9,10\}, \{0,3,9\}, \{0,2,8\}\}$.

To compare the proposed algorithm with the existing methods, we consider an example network with a mixture of one high-power BS, called a macro-BS (MBS) and several low-power BSs termed as the femto-BS (FBS) within the coverage of the MBS ($r_0 = 500\,\text{m}$). All of these BSs are controlled by the cloud. Label the MBS as node 0 and assume that it interferes with each FBS as shown in the examples in Fig. 1. Assume that the FBSs have the same coverage radius denoted as $r_i = d$ for any $i \neq 0$. Averaging over 1000 randomly generated topologies[4], the performance comparisons are shown in Fig. 3a and Fig. 3b. It can be seen that our algorithm performs better than existing algorithms by yielding optimal clustering with higher probability and higher average subgraph densities.

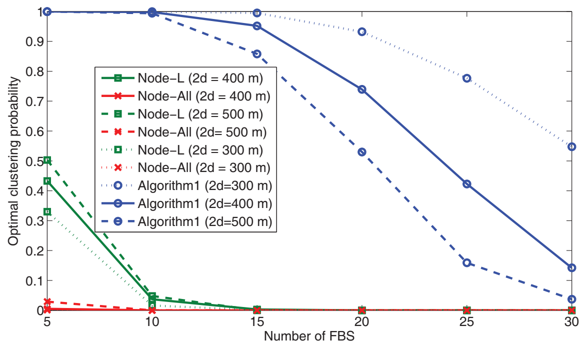
VI. Conclusion

In this paper, we investigate the problem of coordinated enforcement of interference constraint in heterogeneous cloud RAN for auction based service provisioning and resource allocation. For effective resource sharing. One major technical challenge is the integration of interference constraints in auction-based solutions. In this work we transform the interference constraints into a set of linear constraints on each

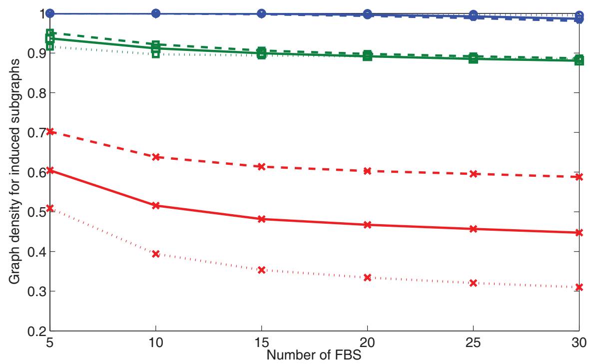
[2]For undirected simple graphs, the density is defined as the ratio of the number of edges over the maximum number of edges for a complete graph. The denser the graph is, the closer it is to a maximal clique.

[3]In an undirected graph, the degree of a node is defined as the number of neighbors of the node.

[4]Random topologies are generated with MBS position fixed at $(0, 0)$, while FBS and UE positions are randomly distributed according to a uniform point distribution over a circular area with radius $500\,m$



(a) Optimality probability comparison.



(b) Graph density comparison.

Fig. 3: Comparison of different clustering algorithms.

cluster. We tackle the generally NP-hard clustering problem by developing a novel practical suboptimal solution that can meet our design requirement. Our polynomial time approximate algorithm demonstrates strong performance in terms of subgraph density and probability of optimal clustering without requiring the high complexity of the optimal solution.

REFERENCES

[1] M. Gerasimenko, D. Moltchanov, R. Florea, S. Andreev, Y. Koucheryavy, N. Himayat, S.-p. Yeh, and S. Talwar, "Cooperative radio resource management in heterogeneous cloud radio access networks," *IEEE Access*, vol. 3, pp. 397 – 406, 2015.

[2] A. Checko, H. L. Christiansen, Y. Yan, L. Scolari, G. Kardaras, M. S. Berger, and L. Dittmann, "Cloud RAN for mobile networks– a technology overview," *IEEE Communications Surveys Tutorials*, vol. 17, no. 1, pp. 405–426, Firstquarter 2015.

[3] R. Wang, H. Hu, and X. Yang, "Potentials and challenges of c-ran supporting multi-rats toward 5G mobile networks," *IEEE Access*, vol. 2, pp. 1187–1195, 2014.

[4] A. Abdelnasser, E. Hossain, and D. I. Kim, "Tier-aware resource allocation in OFDMA macrocell-small cell networks," *IEEE Trans. Commun.*, vol. 63, no. 3, pp. 695–710, March 2015.

[5] Y.-P. Zhang, S. Feng, and P. Zhang, "Adaptive cell association and interference management in lte-a small-cell networks," in *IEEE 78th Vehicular Technology Conference (VTC Fall)*, Sept 2013, pp. 1–6.

[6] Q. Ye, B. Rong, Y. Chen, M. Al-Shalash, C. Caramanis, and J. Andrews, "User association for load balancing in heterogeneous cellular networks," *IEEE Trans. Wireless Commun.*, vol. 12, no. 6, pp. 2706–2716, June 2013.

[7] G. Iosifidis, L. Gao, J. Huang, and L. Tassiulas, "A double-auction mechanism for mobile data-offloading markets," *IEEE/ACM Trans. Netw.*, vol. 23, no. 5, pp. 1634–1647, Oct 2015.

[8] H. Wang and Z. Ding, "Power control and resource allocation for outage balancing in femtocell networks," *IEEE Trans. Wireless Commun.*, vol. 14, no. 4, pp. 2043–2057, April 2015.

[9] L. Wang, X. Feng, X. Wang, X. Gan, and J. Wu, "A user-centric load balance scheme for small cell networks," in *Proc. of IEEE International Conference on Wireless Communications and Signal Processing (WCSP)*, Oct 2014, pp. 1–6.

[10] H. Huang, Y. Sun, X. Li, S. Chen, M. Xiao, and L. Huang, "Truthful auction mechanisms with performance guarantee in secondary spectrum markets," *IEEE Trans. Mobile Comput.*, vol. 14, no. 6, pp. 1315–1329, June 2015.

[11] M. C. Golumbic, *Algorithmic Graph Theory and Perfect Graphs (Annals of Discrete Mathematics, Vol 57)*. Amsterdam, The Netherlands, The Netherlands: North-Holland Publishing Co., 2004.

[12] E. Tomita, A. Tanaka, and H. Takahashi, "The worst-case time complexity for generating all maximal cliques and computational experiments," *Theoretical Computer Science*, vol. 363, no. 1, pp. 28 – 42, 2006, computing and Combinatorics 10th Annual International Conference on Computing and Combinatorics (COCOON 2004). [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0304397506003586

[13] R. Gupta, J. Walrand, and O. Goldschmidt, "Maximal cliques in unit disk graphs: Polynomial approximation," in *Proc. of INOC*, 2005.

[14] S. Gandhi, C. Buragohain, L. Cao, H. Zheng, and S. Suri, "A general framework for wireless spectrum auctions," in *Proc. of 2nd IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks (DySPAN)*, April 2007, pp. 22–33.

[15] D. G. Corneil, *Graph-Theoretic Concepts in Computer Science*. Springer Berlin Heidelberg, 2005, ch. Lexicographic breadth first search – a survey, pp. 1–19.

[16] S. Mneimneh, "Depth / breadth first search with applications." [Online]. Available: www.cs.hunter.cuny.edu/ saad/cources/493.55/note5.pdf

[17] A. Brandstadt, V. Le, and J. Spinrad, *Graph Classes: A Survey*. Society for Industrial and Applied Mathematics, 1999.