Robust Clustering for Ad Hoc Cognitive Radio Network

Di Li, *Member, IEEE,* Erwin Fang, and James Gross, *Member, IEEE*

Abstract—Cluster structure in cognitive radio networks facilitates cooperative spectrum sensing, routing and other functionalities. The unlicensed channels which are available for a group of cognitive radio users, consolidate the group into a cluster and the number of the available unlicensed channels decides that cluster's robustness against the licensed users' influence. This paper analyses the problem of how to form robust clusters in cognitive radio network, so that more cognitive radio users can get the benefits from cluster structure when the primary users' operation becomes more intense. We give a formal description of the robust clustering problem, prove it to be NP-hard and propose both centralized and distributed solutions. The congestion game model is adopted to analyse the process of cluster formation, which not only contributes the design of the distributed clustering scheme, but also provides the guarantee on the convergence into Nash Equilibrium and the convergence speed. The proposed distributed clustering scheme outperforms the related works in terms of cluster robustness, convergence speed and overhead. The extensive simulation is conducted, which clearly supports our claims.

Index Terms—cognitive radio, robust cluster, game theory, congestion game, distributed, centralized, cluster size control.

1 Introduction

Cognitive radio (CR) is a promising technology to solve the spectrum scarcity problem.[?] Licensed users access the spectrum allocated to them whenever there is information to be transmitted. In contrast, as one way, unlicensed users can access the spectrum via opportunistic spectrum access, i.e., they access the licensed spectrum only after validating the channel is unoccupied by licensed users, where spectrum sensing[?] plays an important role in this process. In this hierarchical spectrum access model,[?] the licensed users are also called primary users (PU), while the unlicensed users are referred to as secondary users and constitute a so called cognitive radio network (CRN). Regarding the operation of CRN, efficient spectrum sensing is identified to be critical for a smooth operation of a cognitive radio network.[?] Efficient spectrum sensing can be achieved by cooperative spectrum sensing of multiple secondary users, which has been shown to cope effectively with noise uncertainty and channel fading, thus remarkably improving the sensing accuracy.[?] Collaborative sensing relies on the consensus of CR users[1] within a certain area, in this regard, clustering is regarded as an effective method to realize cooperative spectrum sensing.[?], [? Clustering is a process of grouping certain users in geographic proximity into a collective. In the context of cognitive radio netowrks, a formed cluster helps the participants not only improve the spectrum sensing accuracy, but also coordinate the channel switch operation when the primary users are detected by at least one secondary user in that cluster, then all the cluster menmbers can stop payload transmission swiftly and vacate the operating channel .[?] Clustering also benefits the operation of a CRN by reducing the interference between cognitive clusters,[?] and supporting routing.[?] Except for benefiting the dynamic spectrum sharing in CRN, clustering contributes to the other wireless networks in many aspects, which has already been discussed in many works.

In CRN, clusters are formed in the very beginning of the network operation, and re-formed periodically according to the dynamics of the CRN. Each formed cluster has one or multiple unlicensed channels which are available for every CR node in the cluster. The available unlicensed channels are referred to in the following of this paper as *common licensed channels* (or common channels for short, which is abbreviated as CC). Both payload and control overheads can be transmitted on the CCs. When one or several cluster members can not access one certain CC on which primary user activity is detected, the channel will be excluded from the set of CCs. In particular, if that channel is being used for payload communication, the communication pair will stop and resume the transmission on another available CC. The availability of CCs within a cluster defines the existence of that cluster, i.e., no CCs are available means the corresponding cluster doesn't exist. In the context of CRN, the activity of primary users is usually unknown to the secondary users, then the primary users' activity is deemed as random. In this case, the cluster which secures more CCs will anticipate a longer life expectancy, in other words, is more robust. It is obvious that fewer secondary users in one cluster yield more CCs, but this contradicts to one of the motivations of clustering, i.e., the cooperative decision making, where more users result in more accurate sensing result.[?] Thus the number of small clusters, especially the *singleton clusters*, i.e., the cluster which has only one CR node, should be minimized. As to the cluster which consists of many nodes, although the spectrum sensing is benefited, it usually leads to less common channels, which undermines the robustness of the cluster. Thus the cluster robustness discussed in terms of number of CCs carries little meaning when the sizes of formed clusters are not given consideration. Cluster size also plays a part in transmission power consumption, i.e., cluster size affects the transmission power consumption when

Di Li was with RWTH Aachen university, Germany.
Erwin Fang is with ETH, Switzerland.
James Gross is with KTH Royal Institute of Technology, Sweden.
Manuscript received xxxx xx, 20xx; revised xxxx xx, 20xx.

1. The terms user and node appear interchangeably in this paper. In particular, user is adopted when its networking or cognitive ability are discussed or stressed, while we refer node typicallly in the context of the topology.

routing is conducted.[?], [? In this paper we focus on the efficient formation of the clusters and the evaluation of the robustness of the formed clusters, but our scheme is also able to generate the clusters with the desired size, which gives the network designers more freedom when the power consumption is involved in the design of network applications.

Different clustering strategies have been proposed for wireless networks of different type. As there is no restriction on the nodes to access the spectrum in wireless ad-hoc, mesh networks and sensor networks, the objective of clustering is to decrease the transmission power consumption,[?] to improve the routing performance,[?] or to imrove the network lifetime and coverage.[?] With regard to forming clusters in CRN, deciding on the common channel within each cluster is the foremost question to answer.[?], [?], [? propose the clustering schemes and make sure every cluster possesses at least one CC. Clustering scheme[?] looks for the network partition which improves the accuracy of spectrum sensing with the cluster structure.[?] forms the clusters by deciding on the cluster heads, where the transmit power for the long-haul transmission between the cluster heads is minimized.[?] proposes a cluster structure which promises energy efficiency.[?] proposes strategy on how to decide on the CCs and access the multiple CCs within clusters. An event-driven clustering scheme is proposed for cognitive radio sensor network in.[?] No one among the above mentioned schemes provides the robustness to the formed clusters against primary users.

Robustness of clusters is discussed in,[?] where a distributed clustering scheme (denoted as SOC) which is designed to generate robust clusters against primary users is proposed. The robustness of the clusters comes from the fact that there are multiple CCs available for the clusters. SOC involves three phases of distributed executions. In the first phase, every secondary user forms clusters with some one-hop neighbors, in the second and third phase, each secondary user seeks to either merge other clusters or join one of them. The metric adopted by every secondary user in all the phases is the product of the number of CCs and cluster size. The drawbacks of this scheme are as follows, although the adopted metric considers both cluster size and the number of CCs, cluster formation can be easily dominated by only one factor, e.g. a node which can access many channels may exclude its neighbor and form a cluster by itself. In addition, this scheme leads to the high variance of the cluster sizes, which is not desired in certain applications as discussed in.[?], [?? presents a heuristic method to form clusters, although the authors claim robustness is one goal to achieve, the minimum number of clusters is finally pursued. A distributed clustering scheme ROSS is proposed in[?] under the game theoretic framework. Compared with the clustering schemes introduced above, the clusters are formed faster and the clusters possess more CCs than SOC. But as all the other clustering schemes, this scheme doesn't have control over the formation of the very small or very large clusters which are not desirable. Furthermore, this work doesn't consider the robustness of clusters against the increasing activity of primary users, which leaves their claim of robustness unverified. This paper is on the basis of the work in,[?] but extends in two dimensions. First, this paper renews the definition of robustness of clusters in a CRN, which is no longer the average number of CCs of clusters, but the ability of the clusters to sustain when the primary users' activity becomes more dynamic. Second, this paper proposes size control mechanism, which solves the problem of devergence of clusters sizes in [?] and.[?] Besides, this paper provides a comprehensive analysis of the robust clustering problem and proposes a centralized solution. The new extensions are made on basis of ROSS and its light weight version, the latter involves less overheads thus is more suitable for the scenario where fast deployment is desired. Throughout this paper, we refer to the clustering schemes on the basis of ROSS as the *variants of ROSS*.

The rest of paper is organized as follows. We present the system model and the robust clustering problem in Section **??**. The centralized and distributed solutions are introduced in Section **??** and **??** respectively. Extensive performance evaluation is presented in Section **??**. Finally, we conclude our work and point out the direction for future research in Section **??**.

2 System Model and Problem Formulation

We consider a set of CR users $\mathcal{N}$ and a set of primary users distributed over a given area. A set of licensed channels $\mathcal{K}$ is available for the primary users. The CR users are allowed to transmit on channel $k \in \mathcal{K}$ only if no primary user is detected on channel k. CR users conduct spectrum sensing independently and sequentially on all licensed channels.[2] We adopt the unit disk model[?] for both primary and CR users' transmission. If a CR node i locates within the transmission range of an active primary user p, i is not allowed to use the channel which is being used by p. We assume the primary users change their operation channels slowly, thus we consider the clustering problem at some point in time given some degree of information of the availability of the control channels. Due to the same reason, we omit the time index for the spectrum availability. As the result of spectrum sensing, $K_i \subseteq \mathcal{K}$ denotes the set of available licensed channels for CR user i. As the transmission range of primary users is limited and secondary users have different locations, different secondary users have different views of the spectrum availability, i.e., for any $i, j \in \mathcal{N}$, $K_i = K_j$ does not necessarily hold. We therefore represent the network of CR nodes by a graph $G = (\mathcal{N}, E)$, where $E \subseteq \mathcal{N} \times \mathcal{N}$ such that $\{i, j\} \in E$ if and only if $K_i \cap K_j \neq \emptyset$ and $d_{i,j} < r$, where $d_{i,j}$ is the distance between i, j and r is the radius of secondary user's transmission range. Among the secondary users, we denote by $\text{Nb}(i)$ user i's neighborhood, which consists of the CR nodes located within the transmission range of i.

We assume there is one dedicated control channel which is used to exchange signaling messages during the clustering process. This control channel could be one of the ISM bands or other reserved spectrum which is exclusively used for transmitting control messages.[3] Over the control channel, a secondary user i can exchange its spectrum sensing result K_i with all its one hop neighbors $\text{Nb}(i)$. In the following, we refer to the licensed channels as channels in general, and will explicitly mention the dedicated control channel if necessary.

We give the definition of cluster in CRN as follows. A cluster C is a set of secondary nodes which possess the same set of common channels. In particular, a cluster consists of a cluster head $h(C)$ and a number of cluster members, and the cluster head is able to communicate with any cluster member directly. A cluster can

2. We assume that every node can detect the presence of an active primary user on each channel with certain accuracy. The spectrum availability can be validated with a certain probability of detection. Spectrum sensing/validation is out of the scope of this paper.

3. Actually, the control messages involved in the clustering process can also be transmitted on the available licensed channels through a rendezvous process by channel hopping,[?], [? i.e., two neighboring nodes establish communication on the same channel.

be formed only by the cluster head. The size of C is denoted by $|C|$. When the cluster head of a cluster is i, we denote that cluster by $C(i)$. $K(C)$ denotes the set of CCs in cluster C, $K(C) = \Cap_{i \in C} K_i$. The notations used in the system model are listed in Table **??**.

TABLE 1. Notations

Symbol	Description
$\mathcal{N}$	set of CR users in a CRN
N	number of CR users in a CRN, $N = \|\mathcal{N}\|$
$\mathcal{K}$	set of licensed channels
$k(i)$	the working channel of user i
$\mathrm{Nb}(i)$	the neighborhood of CR node i
$C(i)$	a cluster whose cluster head is i
K_i	the set of available channels at CR node i
$K(C(i))$	the set of available CCs of cluster $C(i)$
$h(C)$	the cluster head of a cluster C
δ	the cluster size which is preferred
S_i	a set of claiming clusters, each of which includes debatable node i after phase I
d_i	individual connectivity degree of CR node i
g_i	neighborhood connectivity degree of CR node i
$f(C)$	the number of CCs of a cluster C, which is used in the problem description
$\mathcal{S}$	the collection of all the possible clusters in $\mathcal{N}$
C_i	the i-th cluster in $\mathcal{S}$
$\|C_i\|$	size of the cluster C_i
$\|K(C_i)\|$	the number of CCs of cluster C_i
n	the number of debatable nodes
m	the number of claiming cluster heads

2.1 Robust Clustering Problem in CRN

As introduced in Section **??**, in order to be robust against primary users' activity, the formed clusters should have more CCs. On the other hand, the sizes of the formed clusters should be regulated, i.e., they don't diverge from a given value greatly.

Definition 1: *Robust clustering problem in CRN.*

As to a cognitive radio network, where the set of CR nodes is $\mathcal{N}$. Each cluster complies with the definition in **??**, *whose number of CCs in a cluster C is $f(C)$. The robust clustering problem is to decide the set of clusters $\mathcal{T}$, where*

1) *the sum of the $f(C) is the maximal$, where $C \in \mathcal{T}$*
2) *the intersection of any two clusters in $\mathcal{T}$ is an empty set*
3) *the union of clusters in $\mathcal{T}$ is $\mathcal{N}$*
4) *the cluster sizes should fall in the scope $\langle \delta_1, \delta_2 \rangle$, where $\delta, \delta_1, \delta_2 \in \mathbb{Z}^+$, $\delta_1 \leq \delta \leq \delta_2$, and δ is decided according to the application to be implemented.*

Assume we have obtained the set of $\mathcal{S}$ which contains all the possible clusters in $\mathcal{N}$, i.e., $\mathcal{S} = \{C_1, C_2, \ldots, C_i, \ldots, C_{|\mathcal{S}|}\}$ [4] and there is $\bigcup_{1 \leq i \leq |\mathcal{S}|} C_i = \mathcal{N}$. The decision version of this problem is to determine whether there is a non-empty set $\mathcal{S}' \subseteq \mathcal{S}$, so that $\sum_{C_i \in \mathcal{S}'} f(C_i) \geqslant \lambda$ where λ is a real number. We have the following theorem on the complexity of this problem.

Theorem 2.1: *The robust clustering problem in CRN is NP-hard, when the maximum size of clusters is larger than 3, $\delta_1 = 1$ and $\delta_2 = N$.*

4. The subscript i means the i-th cluster in $\mathcal{S}$.

The proof is in Appendix **??**.

3 Centralized Solution for Robust Clustering

When the global knowledge of the CRN is available to us, we can propose a centralized scheme as comparison. The proposed centralized solution formulates the problem in Definition **??** as an optimization problem which is solved with standard software packages. The optimization searches the set $\mathcal{S}$ and decides on the clusters according to the following optimization formulation,

$$\begin{aligned} \max_{y_i, x_{ij}} \quad & \Sigma_{j=1}^{N} \Sigma_{i=1}^{M} (y_i \cdot t_{ij}) \\ \text{subject to} \quad & \Sigma_{i=1}^{M} x_{ij} = 1, for\ \forall j = 1, \ldots, N \\ & \Sigma_{j=1}^{N} x_{ij} = |C_i| \cdot y_i,\ for\ \forall i = 1, \ldots, M \\ & i \in \{1, 2, \cdots M\}, \quad j \in \{1, 2, \cdots N\} \end{aligned} \tag{1}$$

This problem is a binary linear programming problem, which can be solved by many available solvers. y_i and x_{ij} are two binary variables. Being either 1 or 0, y_i denotes whether the i-th cluster C_i in $\mathcal{S}$ is chosen or not. x_{ij} indicates whether the CR node j resides in the cluster C_i, i.e., $x_{ij} = 1$ means node j resides in the cluster C_i. N is the total number of CR users in network $\mathcal{N}$, M is the number of clusters in $\mathcal{S}$.

The constraints guarantee to obtain the clusters which together include all the CR users and don't overlap. The first constraint regulates that a CR node should reside in exactly one cluster. The second constraint regulates that when the i-th cluster C_i is chosen, there will be exactly $|C_i|$ CR nodes residing in C_i.

The objective is to maximize the sum of the numbers of CCs in the clusters which constitute the CRN. t_{ij} is a constant and there is

$$t_{ij} = \frac{q_{ij}}{|C_i|} - p_i(C_i) \tag{2}$$

where constant $q_{ij} = |K(C_i)|$ when node $j \in C_i$, and $q_{ij} = 0$ when node $j \notin C_i$. $p_i(C_i)$ is the size-related weight, which reflects the deviation of C_i's size from the desired size. Assuming δ is the desired size, then there is

$$p_i(C_i) = \begin{cases} 0 & \text{if } |C_i| = \delta \\ \rho_1 & \text{if } ||C_i| - 1| = \delta \\ \rho_2 & \text{if } ||C_i| - 2| = \delta \\ \vdots & \end{cases}$$

where $\rho_1, \rho_2, \cdots$ are positive values and these is $\rho_2 > \rho_1 > 0$.

When t_{ij} is replaced with $\frac{q_{ij}}{|C_i|} - p_i(C_i)$, the objective function becomes,

$$\max_{y_i, x_{ij}} \quad \Sigma_{j=1}^{N} \Sigma_{i=1}^{M} (y_i \cdot \frac{q_{ij}}{|C_i|} - y_i \cdot p_i(C_i))$$

The sum of the first items is the sum of CCs of all the chosen clusters. As to the second item, when w_i is 1 (C_i is chosen) and $|C_i| \neq \delta$, it will be negative, which contradicts the direction of the optimization. Thus the second item discourages the appearance the clusters whose sizes deviate from δ.

The difficulty of using this method lies in obtaining the set $\mathcal{S}$. In the worst case, i.e., the CRN forms a full connected graph, the size of $\mathcal{S}$ is $\Sigma_{r=1}^{N} \binom{N}{r} = 2^N - 1$. Another obstacle comes from the fact that the centralized controller needs to be reliable at anytime, which is a challenge for CRN as the spectrum on the controller can not be guaranteed.

4 Distributed Clustering Algorithm: ROSS

In this section we introduce the distributed clustering scheme ROSS. With ROSS, CR nodes form clusters based on the proximity of the available spectrum in their neighborhood after a series of interactions with their neighbors. ROSS consists of two cascaded phases: *cluster formation* and *membership clarification*. In the first phase, clusters are formed quickly and every CR user becomes either a cluster head or a cluster member, besides, cluster size control is implemented in this phase. In the second phase, non-overlapping clusters are formed in a way that the CCs of relevant clusters are mostly increased.

4.1 Phase I - Cluster Formation

We assume that before conducting clustering, spectrum sensing, neighbor discovery and exchange of spectrum availability have been completed, so that every CR node is aware of the available channels for themselves and their neighbors. In this phase, cluster heads are determined after a series of comparisons with their neighbors. Two metrics are proposed to characterize the proximity in terms of available spectrum between CR node i and its neighborhood, which are used in the comparisons to decide on the cluster heads.

- *Individual connectivity degree* d_i: $d_i = \sum_{j \in \mathrm{Nb}(i)} |K_i \cap K_j|$. d_i is the total number of the CCs between node i and each of its neighbors.
- *Neighborhood connectivity degree* g_i: the number of CCs which are available for i and all its neighbors. $g_i = |\bigcap_{j \in \mathrm{Nb}(i) \cup i} K_j|$, which represents the ability of i to form a robust cluster with its neighbors.

Individual connectivity degree d_i and neighborhood connectivity degree g_i together form the *connectivity vector*. Figure **??** illustrates an example CRN where every node's connectivity vector is shown.

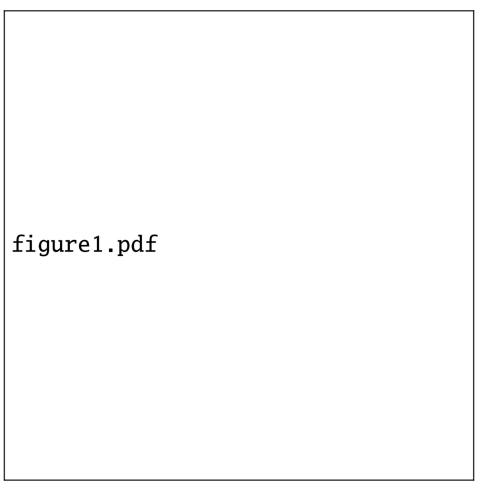

Fig. 1: Connectivity graph of the example CRN and the connectivity vector (d_i, g_i) for each node. The desired cluster size $\delta = 3$. The sets of the indices of the available channels sensed by each node are: $K_A = \{1,2,3,4,5,6,10\}, K_B = \{1,2,3,5,7\}, K_C = \{1,3,4,10\}, K_D = \{1,2,3,5\}, K_E = \{2,3,5,7\}, K_F = \{2,4,5,6,7\}, K_G = \{1,2,3,4,8\}, K_H = \{1,2,5,8\}$. Dashed edge indicates the end nodes are within each other's transmission range.

4.1.1 Determining Cluster Heads and Forming Clusters

The procedure of determining the cluster heads is as follows. Each CR node decides whether it is a cluster head by comparing its connectivity vector with its neighbors. When CR node i has lower individual connectivity degree than all its neighbors except for those which have already identified to be cluster heads, node i becomes a cluster head. If there is another CR node j in its neighborhood which has the same individual connectivity degree as i, i.e., $d_j = d_i$ and $d_j < d_k, \forall k \in \mathrm{Nb}(j) \setminus \{\Lambda \cup i\}$ where Λ denotes the cluster heads, then the node between i and j, which has higher neighborhood connectivity degree will become the cluster head, and the other node will become one member of the newly identified cluster head. If $g_i = g_j$ as well, the node ID is used to break the tie, i.e., the one with smaller node ID becomes the cluster head. The node which is identified as a cluster head broadcasts a message to notify its neighbors of this change, and its neighbors which are not cluster heads become cluster members [5]. The pseudo code for the cluster head decision and the initial cluster formation is shown in Algorithm **??** in the appendix.

After receiving the notification from a cluster head, a CR node i is aware that it becomes a member of a cluster. Consequently, i sets its individual connectivity degree to a positive number $M > |\mathcal{K}| \cdot N$, and broadcasts the new individual connectivity degree to all its neighbors. When a CR node i is associated to multiple clusters, i.e., i has received multiple notifications from different cluster heads, d_i is still set to be M. The manipulation of the individual connectivity degree of the cluster members accelerates the decision on the cluster heads. We have the following theorem to show that when there is no one secondary which shares no CC with any other secondary users, i.e., all secondary users' individual connectivity degrees are greater than zero, then every secondary user will eventually be either integrated into a certain cluster, or becomes a cluster head.

Theorem 4.1: *Given a CRN, it takes at most N steps that every secondary user either becomes cluster head, or gets included into at least one cluster.*

Here, by *step* we mean one secondary user executing Algorithm **??** for one time. The Proof is in Appendix **??**.

The procedure of the proof also illustrates the maximal time needed to conduct Algorithm **??**. Consider an extreme scenario, where all the secondary nodes sequentially execute Algorithm **??**, i.e., they constitute a list as discussed in the example in the proof. If one step can be finished within a certain time span T, then the total time needed for the network to conduct Algorithm **??** is $N * T$. Actually as Algorithm **??** can be executed concurrently by different secondary users, the needed time can be considerably reduced. If we apply Algorithm **??** to the example shown in Figure **??**, then the outcome is shown in Figure **??**. Node B and H have the same individual connectivity degree, i.e., $d_B = d_H$. As $g_H = 2 > g_B = 1$, node H becomes the cluster head and cluster $C(H)$ is $\{H, B, A, G\}$.

4.1.2 The Existence of Common Channels

After executing Algorithm **??**, certain formed clusters may not possess any CCs. As decreasing cluster size increases the CCs within a cluster, for those clusters having no CCs, certain nodes need to be eliminated to obtain at least one CC. The sequence

5. The reason for the occurrence of the cluster heads in the neighborhood of a new cluster head will be explained in Section **??** and **??**)

of elimination is performed according to an ascending list of nodes which are sorted by the number of common channels between the nodes and the cluster head. In other words, the cluster member which has the least common channels with the cluster head is excluded first. If there are multiple nodes having the same number of common channels with the cluster head, the node whose elimination brings in more common channels will be excluded. If this criterion meets a tie, the tie will be broken by deleting the node with smaller node ID. It is possible that the cluster head excludes all its neighbors, resulting in a singleton cluster which is composed by itself. The pseudo code for this procedure is shown in Algorithm **??**. As to the nodes which are eliminated from the previous clusters, they restore their original individual connectivity degrees, then execute Algorithm **??** and become either cluster heads or get included into other clusters afterwards according to Theorem **??**.

During Phase I, whenever a CR node is decided to be a cluster head and accordingly forms a cluster, or its cluster's composition is changed, the cluster head will broadcast the updated information about its cluster, which includes the sets of available channels on all its cluster members.

4.1.3 Cluster Size Control in Dense CRN

It is necessary to control the cluster size when CRN becomes denser. Both analysis and simulation[?] show that when applying ROSS, after the clusters are saturated with the increase of network density, the cluster size increases linearly with the network density, thus certain measures are needed to curb this problem. This task falls upon the cluster heads. To control the cluster size, cluster heads prune their cluster members to reach the desired cluster size. The desired size δ is decided based on the capability of the CR users and the tasks to be conveyed. As there are overlaps between neighboring clusters, the sizes of the clusters formed in this phase are larger than that of the finally formed clusters. Hence, a cluster head excludes some cluster members when the cluster size exceeds $t \cdot \delta$, where constant parameter t is dependent on the network density and CR nodes' transmission range and $t > 1$. In particular, the cluster head removes the cluster members sequentially according to the following principle, the absence of one cluster member leads to the maximum increase of the CCs within the cluster. This process ends when each cluster's size is smaller or equal to $t \cdot \delta$. This procedure is similar with that in Section **??**, thus Algorithm **??** can be reused. The t is set to 1.3.

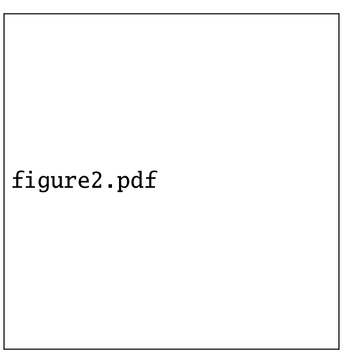

Fig. 2: Clusters formation after the phase I of ROSS. CR nodes A, B, D are debatable nodes as they belong to multiple clusters.

4.2 Phase II - Membership Clarification

As to the example CRN shown in Figure **??**, the resulted clusters are shown in Figure **??** after running phase I of ROSS. We notice that nodes A, B, D are included in more than one cluster. We refer to these nodes as *debatable nodes* as their cluster affiliations are not decided. The clusters which include the debatable node i are called *claiming clusters* of node i, and the set of these clusters is denoted as S_i. The debatable nodes which are generated from the first phase of ROSS should be exclusively associated with only one cluster and be removed from the other claiming clusters, this procedure is called *cluster membership clarification*.

4.2.1 Distributed Greedy Algorithm (DGA)

Assuming a debatable node i which needs to decide one cluster $C \in S_i$ to stay and leaves the other clusters in S_i, then the principle for i is its decision should result in the greatest increase of CCs in all its claiming clusters. As node i has been notified of the spectrum availability on all the nodes in each claiming cluster, node i is able to calculate how many more CCs will be produced in a claiming cluster if i leaves that cluster. Then node i decides on the cluster $C \in S_i$, if i leaving cluster C results in less increased CCs than leaving any other claiming clusters in S_i. When there comes a tie between two claiming clusters, i chooses to stay in the cluster whose cluster head shares the most CCs with i. When the tie still exists, node i chooses to stay in the claiming cluster which has the smallest size. Node IDs of cluster heads will be used to break tie if all the previous metrics could not decide on the unique claiming cluster for i to stay. The pseudo code of this algorithm is given in Algorithm **??**. After deciding its membership, debatable node i notifies all its claiming clusters of its choice, and the claiming clusters from which node i leaves also broadcast their new cluster composition and the spectrum availability on all their cluster members.

The autonomous decisions made by the debatable CR nodes raise the concern on the endless chain effect in the membership clarification phase. A debatable node's choice is dependent on the compositions of its claiming clusters, which can be changed by other debatable nodes' decisions. As a result, the debatable node which makes decision first may change its original choice, and this process may go on forever. To erase this concern, we formulate the process of membership clarification into a game, where a equilibrium is reached after a finite number of best response updates made by the debatable nodes.

4.2.2 Bridging ROSS-DGA with Congestion Game

Game theory is a powerful mathematical tool for studying, modelling and analysing the interactions among individuals. A game consists of three elements: a set of players, a selfish utility for each player, and a feasible strategy space for each player. In a game, the players are rational and intelligent decision makers, which are related with one explicit formalized incentive expression (the utility or cost). Game theory provides standard procedures to study its equilibriums.[?] In the past few years, game theory has been extensively applied to problems in communication and networking.[?], [?] Congestion game is an attractive game model which describes the problem where participants compete for limited resources in a non-cooperative manner, it has the good property that Nash equilibrium can be achieved after finite steps of best response dynamic, i.e., each player chooses the strategy to maximize/minimize its utility/cost with respect to the other players' strategies. The framework of the congestion game has been used to model certain problems in internet-centric applications or cloud computing, where self-interested clients compete for the centralized resources and meanwhile interact with each other. For example, server selection is involved in distributed computing platforms,[?] or users downloading files from cloud, etc.

To formulate the debatable nodes' membership clarification into the desired congestion game, we reexamine this process from a different (or opposite) perspective. From the new perspective, the debatable nodes are not included in any cluster, and they are looking for the proper cluster to join. A debatable node i chooses one cluster C out of S_i to join if the decrease of CCs in cluster C is the smaller than any other cluster in S_i. The decrease of CCs in cluster C is $\sum_{C\in S_i} \Delta|K(C)| = \sum_{C\in S_i}(|K(C)| - |K(C \cup i)|)$. The interaction between the debatable nodes and the claiming clusters is shown in Figure **??**.

singletongame_png.png

Fig. 3: Debatable nodes and claiming clusters

In the following, we show that the decision of debatable nodes to clarify their membership can be mapped to the behaviour of the players in a *player-specific singleton congestion game* when proper cost function is given. The game to be constructed is represented with a 4-tuple $\Gamma = (\mathcal{P}, \mathcal{R}, \sum_{i,i\in\mathcal{P}}, f)$ with the following elements:

- $\mathcal{P}$, the set of players in the game, which are the debatable nodes in our problem.
- $\mathcal{R} = \cup S_i, i \in \mathcal{P}$, denotes the set of resources for players to choose, in our problem, S_i is the set of claiming clusters of node i, and $\mathcal{R}$ is the set of all claiming clusters.
- Strategy space $\sum_i, i \in \mathcal{P}$, $\sum_i$ is the set of claiming clusters S_i for i. As debatable node i is supposed to choose only one claiming cluster, then only one piece of resource will be allocated to i.
- The utility (cost) function $f(C)$ as to a resource C. $f(C) = \Delta|K^i(C)|, C \in S_i$, which represents the decreased number of CCs in cluster C when debatable node i joins C. As to cluster $C \in S_i$, the decrease of CCs caused by including the debatable nodes is $\sum_{i:C\in S_i, i\to C} \Delta|K^i(C)|$. $i \to C$ means i joins cluster C. Obviously this function is non-decreasing with respect to the number of nodes joining cluster C.
  The utility function f is not purely decided by the number of players accessing the resource (debatable nodes join claiming clusters), which happens in a canonical congestion game. The reason is in this game the channel availability on debatable nodes is different. Given two same groups of debatable nodes and their sizes are the same, when the nodes are not completely the same (neither are the channel availabilities on these nodes), the cost happened on one claiming cluster could be different if the two groups of debatable nodes join that cluster respectively. Hence, this congestion game is player specific.[?] In this game, every player greedily updates its strategy (choosing one claiming cluster to join) if joining a different claiming cluster minimizes the decrease of CCs $\sum_{i:C\in S_i} \Delta|K^i(C)|$, and a player's strategy in the game is exactly the same with the behaviour of a debatable node in the membership clarification phase.

As to singleton congestion game, there exists a pure equilibria which can be reached with the best response update, and the upper bound for the number of steps before convergence is $n^2 * m$,[?] where n is the number of players, and m is the number of resources. In our problem, the players are the debatable nodes, and the resources are the claiming clusters. Thus the number of steps can be expressed as $O(N^3)$.

In fact, the upper bound of the number of steps which are actually involved in this process is much smaller than N^3.The percentage of debatable nodes in the network is illustrated in Figure **??**, which is between 10% to 60% of the total number of CR nodes in the network. In the other hand, the number of clusters heads is dependent on the network density and the CR node's transmission range as mentioned in Section **??**. The simulation in[?] shows that the cluster heads are only 3.4% to 20% of the total CR nodes.

4.2.3 Distributed Fast Algorithm (DFA)

On the basis of ROSS-DGA, we propose a faster version ROSS-DFA which differs from ROSS-DGA in the second phase. With ROSS-DFA, debatable nodes decide their respective cluster heads only once. The debatable nodes consider their claiming clusters to include all their debatable nodes, thus the membership of claiming clusters is static and all the debatable nodes can make decisions simultaneously without considering the change of membership of their claiming clusters. As ROSS-DFA is quicker than ROSS-DGA, the former is especially suitable for the CRN where the channel availability changes dynamically and re-clustering is necessary. To run ROSS-DFA, debatable nodes execute only one loop in Algorithm **??**.

Now we apply both ROSS-DGA and ROSS-DFA to the toy network in Figure **??** which has been applied the phase I of ROSS. In the network, node A's claiming clusters are cluster $C(C), C(H) \in S_A$, their members are $\{A, B, C, D\}$ and $\{A, B, H, G\}$ respectively. The two possible strategies of node A is illustrated in Figure **??**. In Figure **??**, node A staying in $C(C)$ and leaving $C(H)$ brings 2 more CCs to S_A, which is more than that brought by another strategy shown in **??**. After the decisions made similarly by the other debatable nodes B and D, the final clusters are formed as shown in Figure **??**.

figure4AinC.pdf

(a) Node A stays in cluster $C(C)$, quits $C(H)$, $\Delta|K(C(C))| + \Delta|K(C(H))| = 2$

figure4AinH.pdf

(b) Node A stays in cluster $C(H)$, quits $C(C)$, $\Delta|K(C(C))| + \Delta|K(C(H))| = 1$

Fig. 4: Membership clarification: possible cluster formations caused by node A's different choices

5 Performance Evaluation

The schemes involved in the simulation are listed as follows, final_clustering_ross.pdf

Fig. 5: Final formation of clusters. Common channels are shown beside corresponding clusters.

- ROSS without size control, i.e., ROSS-DGA and ROSS-DFA.
- ROSS with size control, i.e., ROSS-δ-DGA and ROSS-δ-DFA where δ is the desired cluster size. In the following, we refer to the above mentioned four schemes as the variants of ROSS.
- SOC,[?] a distributed clustering scheme pursuing cluster robustness.
- Centralized robust clustering scheme. As shown in Section **??**, the centralized robust clustering scheme is formulated as an integer linear optimization problem and is solved by MATLAB with the function *bintprog*.

The ROSS without size control mechanism is similar with the schemes proposed in.[?] The authors of[?] compared SOC with other schemes in terms of the average number of CCs of the formed cluster, on which SOC outperforms other schemes by 50%-100%. SOC's comparison schemes are designed either for ad hoc network without consideration of channel availability,[?] or for CRN but just considering connection among CR nodes.[?] Thus SOC is the only distributed scheme as comparison. As to the CRN shown in Figure **??**, the resulting clusters by the centralized scheme and SOC are shown in Figure **??**.

(a) Generated by SOC (b) Generated by the centralized clustering scheme

Fig. 6: Final clusters formed by the centralized clustering scheme and SOC.

We investigate the schemes with respect to four metrics.

- **The average number of CCs per non-singleton cluster.** Non-singleton cluster refers to the cluster whose cluster size is larger than 1. Previous work[?] and[?] claim that the larger average number of CCs over all the clusters indicates longer life expectancy when the primary users' operation becomes more intense. We see two flaws in this claim. First, the unclustered CR nodes (synonym of singleton clusters) should not be considered when obtaining the average number of CCs, as singleton clusters don't contribute to the collaborative computing. Second, the average number of CCs doesn't necessarily indicate the robustness of individual clusters, because the ability for a cluster to sustain also depends on the cluster size and the locations of the cluster members, and this information is not illustrated in the average number of CCs. In the performance evaluation, we will examine the metric of average number of CCs per non-singleton cluster, which excludes the bias brought in by the unclustered CR nodes. More importantly, we will examine whether this metric reflects the robustness of the clusters against increasingly active primary users.
- **Cluster sizes.** We investigate the distribution of CRs residing in the formed clusters with different sizes.
- **Robustness of the clusters against newly added PUs.** We increase the number of PUs to challenge the non-singleton clusters, and count the number of the unclustered CR nodes. This metric directly indicates the robustness of clusters from a more practical point of view, i.e., as to the clusters formed for a given CRN and spectrum availability, how many CR nodes can still make use of the clusters when the spectrum availability decreases.
- **Amount of control messages involved.** We investigate the number of control messages involved in the clustering process.

Simulation consists of two parts, first we investigate the performance of centralized scheme and the distributed schemes in a small network, as there is no polynomial time solution available to solve the centralized problem. In the second part, we investigate the performance of the proposed distributed schemes in the CRN with different scales and densities. The following simulation setting is the same for both simulation parts. CRs and PUs are deployed on a two-dimensional Euclidean plane. The number of licensed channels is 10, each PU is operating on each channel with probability of 50%. The constant t which is used to control cluster size for ROSS (discussed in Section **??**) is 1.3. CR users are assumed to be able to sense the existence of primary users and identify available channels. All primary and CR users are assumed to be static during the process of clustering. The simulation is written in C++, and the performance results are averaged over 50 randomly generated topologies, and the confidence interval corresponds to 95% confidence level.

5.1 Centralized Schemes vs. Decentralized Schemes

There are 10 primary users and 20 CR users dropped randomly (with uniform distribution) within a square area of size A^2, where we set the transmission ranges of primary and CR users to $A/3$. When clustering scheme is executed, around 7 channels are available on each CR node. The desired cluster size δ is 3. As for the centralized scheme, the parameters used in the *punishment* for choosing the clusters with undesired sizes are set as follows, $\rho_1 = 0.4$, $\rho_2 = 0.6$.

5.1.1 Average number of CCs in Non-singleton Clusters

From Figure **??**, we can see the centralized schemes outperform the distributed schemes. Among the distributed schemes, SOC achieves the most CCs. The reason is, SOC is liable to group the neighboring CRs which share the most abundant spectrum together, no matter how many of them are there, thus the number of CC of the formed clusters is higher. On the other hand, SOC generates the most unclustered CRs. As to the variants of ROSS, we notice that the greedy mechanism increases CCs in non-singleton clusters significantly.

ccc_20.pdf

Fig. 7: Number of common channels of non-singleton clusters

cdf_clusterSize_20.pdf

Fig. 8: Cumulative distribution of CRs residing in clusters with different sizes

survival_rate_20.pdf

Fig. 9: Number of unclustered CRs with decreasing spectrum availability

Fig. 10: Comparison between the distributed and centralized clustering schemes ($N = 20$)

5.1.2 Cluster Size

Figure **??** depicts the empirical cumulative distribution of the CRs in clusters of different sizes, from which we have two conclusions. First, given the channel availability in the CRN, SOC generates more unclustered CR nodes than other schemes. The centralized schemes don't produce unclustered CR nodes in the simulation, the unclustered nodes generated by ROSS-DGA/DFA account for 3% of the total CR nodes, as comparison, 10% of nodes are unclustered when applying SOC. ROSS-DGA and ROSS-DFA with size control feature generate 5%-8% unclustered CR nodes, which is due to the cluster pruning procedure (discussed in section **??** and section **??**). Second, the centralized schemes and cluster size control mechanism of ROSS generate clusters with the desired cluster size. As to ROSS-DFG and ROSS-DFA with size control feature, CR nodes reside averagely in clusters whose sizes are 2, 3 and 4. The sizes of clusters resulted from ROSS-DGA and ROSS-DFA are disperse, but appear to be better than SOC, i.e., the 50% percentiles for ROSS-DGA, ROSS-DFA and SOC are 4.5, 5, and 5.5, and the 90% percentiles for the three schemes are 8, 8, and 9, the corresponding sizes of ROSS are closer to the desired size.

5.1.3 Robustness of the clusters against newly added PUs

In this part of simulation, we put PUs sequentially into CRN to decrease the available spectrum. 10 PUs are in the network in the beginning, then extra 19 batches of PUs are added sequentially, where each batch includes 5 PUs. Figure **??** shows certain clusters can not maintain and the number of unclustered CR nodes grows when the number of PUs increases. The centralized scheme with desired size of 2 generates the most robust clusters, meanwhile, SOC results in the most vulnerable clusters. The centralized scheme with desired size of 3 doesn't outperform the variants of ROSS, because pursuing cluster size prevents forming the the clusters with more CCs. In contrary, the variants of ROSS generate some smaller clusters which are more likely to maintain when there are more PUs.

The above observation shows that the average number of CCs of non-singleton clusters doesn't necessarily illustrate the robustness of cluster against increasingly active PUs, i.e., SOC obtains the largest average CCs, but generates the most vulnerable clusters. Besides, with similar distribution of sizes, the clusters generated by ROSS-DGA and ROSS-DFA are more robust than that by SOC.

5.1.4 Control Signaling Overhead

In this section we compare the overhead of signaling involved in different clustering schemes. We count the number of *transmissions of control messages* as message complexity,[?] and without distinguishing broadcast or uni-cast control messages. In Section **??**, this metric is synonymous with the *the number of updates*.

As to ROSS, in the first phase the maximal number of broadcast is N according to **??**. The upper bounds of the transmissions are n^2m and n for ROSS-DGA and ROSS-DFA respectively. Scheme SOC consists of three rounds and in each round every node needs to broadcast in order to do comparisons and cluster mergers. The centralized scheme is conducted at the centralized control device, and it involves information aggregation and clustering decision dissemination. We adopt the backbone structure proposed in[?] to analyze the centralized scheme's message complexity. We apply ROSS to generate cluster heads which serve as the backbone. In the process of information aggregation, all the nodes transmit information to the cluster heads which forward the messages to the controller, then in the process of dissemination, all the cluster heads and the debatable nodes broadcast the clustering result, thus the upper bound of the number of broadcast is $N{+}m{+}n$.

The number of control messages which are involved in ROSS variants and the centralized scheme is related with the number of debatable nodes. Figure **??** shows the percentage of debatable nodes with different network densities. Table **??** shows the message complexity, quantitative amount of the control messages, and the size of control messages. Figure **??** shows the analytical result of the amount of transmissions involved in different schemes.

5.2 Comparison among the Distributed Schemes

In this section we investigate the performances of the proposed distributed clustering schemes in CRN with different network scales and densities. The transmission range of CR is $A/5$, PU's