Versatile Robust Clustering of 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 every member of a group of cognitive radio users, consolidate the group into a cluster, and the availability of unlicensed channels decides the robustness of that cluster against the licensed users' influence. This paper analyses the problem that how to form robust clusters in cognitive radio network, so that more cognitive radio users can get benefits from cluster structure even when the primary users' operation are intense. We provide a formal description of robust clustering problem, prove it to be NP-hard and propose a centralized solution, besides, a distributed solution is proposed to suit the dynamics in the ad hoc cognitive radio network. Congestion game model is adopted to analyse the process of cluster formation, which not only contributes designing the distributed clustering scheme directly, but also provides the guarantee of convergence into Nash Equilibrium and convergence speed. Our proposed clustering solution is versatile to fulfill some other requirements such as faster convergence and cluster size control. The proposed distributed clustering scheme outperforms the related work in terms of cluster robustness, convergence speed and overhead. The extensive simulation 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 [1]. 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 [2] plays an important role in this process. In this hierarchical spectrum access model [3], 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 [4]. This 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 [5]. 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 [6], [7]. Clustering is a process of grouping certain users in a proximity into a collective. Clustering is also efficient to coordinate the channel switch operation when primary users are detected by at least one CR node residing in the cluster. The cluster head (CH) can enable all the CR devices within the same cluster to stop payload transmission swiftly on the operating channel and to vacate the channel [8]. In addition to the collaborate sensing advantage, the use of clusters is beneficial as it reduces the interference between cognitive clusters [9]. Clustering algorithm has also been proposed to support routing in cognitive radio networks [10].

Forming clusters is conducted in the very beginning or periodically according to the dynamics of the CRN. To form a cluster which is composed by the CR users, there should be channels available for communication within the cluster. Usually there are multiple unlicensed channels available for all the CR nodes in the cluster, which are referred to in the following of this paper as *common control channels* (CCC). Out of the available CCCs, there are always one or multiple channels which are used for the payload communication. When one or several cluster members can not use one certain CCC for example because primary user activity has been detected on that channel, it will be excluded from the set of CCCs. If the channel is being used for payload communication, all the cluster members will switch to another channel in the set of available CCCs to achieve seamless transmission. As long as there is a CCC available, the corresponding cluster can continue to exchange payload data. In the context of CRN, as the activity of primary users is unknown to the secondary users, the availability of CCCs of the formed clusters is totally dependent on primary users' activity. Thus, the clusters with more CCCs are in general better as this anticipates a longer time span that payload can be exchanged. In this paper the robustness of a cluster against primary users is synonymous with the affluence of CCCs which are possessed by that cluster.

In the formation process, there is a trade-off between CCCs and the collaborative sensing performance, as less members in a cluster yield in general a higher number of CCCs while this reduces the sensing accuracy [11]. In addition, there are other reasons that motivate clusters to have more members. For example, instead of less members, the proper cluster size results in

D. Li was with RWTH Aachen university, Germany.
Erwin. Fang is with ETH, Switzerland.
J. 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.

smaller power consumption [12], [13]. Thus the cluster robustness discussed in terms of number of CCCs carries little meaning when the sizes of formed clusters are not given consideration.

There has been a lot of research done for clustering in wireless networks. In ad-hoc and mesh networks, the major goal of clustering is to maximize connectivity or to improve the performance of routes [14], [15]. The emphasis of clustering in sensor networks is on network lifetime and coverage [10]. Various clustering schemes are proposed to target different aspects in cognitive radio networks. The clustering schemes proposed in [7], [16], [17] form clusters which meet the basic condition, having CCC in every cluster. Clustering scheme [11] improves spectrum sensing accuracy. [12], [18] target on the QoS provisioning and energy efficiency. [19] forms clusters to coordinate the control channel usage within one cluster. A event-driven clustering scheme is proposed for cognitive radio sensor network in [20]. No one among the above mentioned schemes provides the robustness to the formed clusters against primary users.

A clustering scheme (denoted by SOC) which is designed to generate robust clusters against primary users is proposed in [21]. 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 around it, or join one of them. The metric adopted by every secondary user in the three phases is the product of the the number of CCCs and cluster size (the number of secondary users in the cluster). But this robust clustering scheme has some drawbacks. Although the adopted metric covers both cluster size and the number of CCCs, the formation of certain clusters may be easily dominated by only one factor, e.g. a node which is able to use many channels will 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 [12], [19]. [22] presents a heuristic method to form clusters, although the authors claim robustness is one goal to achieve, the minimum number of clusters is pursued. [23] proposes a distributed clustering scheme under the game theoretic framework. Compared with the clustering schemes introduced so far, the clusters are formed in shorter time and posses more CCCs within and among clusters. But this scheme doesn't consider the the factor of size in the algorithm design.

In this paper we will give a comprehensive analysis of the robust clustering problem and propose solutions. We stick to the motivation of forming robust clusters in CRN i.e., let more CR users benefited from the cooperative decision making which is due to the clusters. We propose both centralized and distributed clustering schemes, which result in more CR users in the clusters composed with multiple CR users, not only when the clusters are just formed, but also afterwards when the primary users' activity changes. Besides, both centralized and distributed schemes take the cluster size into consideration, and both can control the sizes of the formed clusters to certain extent. In particular, the decentralized clustering approach ROSS (RObust Spectrum Sharing) is able to form clusters with desired sizes. When compared with other distributed works, ROSS involves smaller signaling overhead and the generated clusters are significantly more robust against the primary users which appear after the clusters are formed. We also propose the light weighted versions of ROSS, which involve less overheads and thus are more suitable for the scenario where fast deployment is desired. Throughout this paper, we refer the clustering schemes on the basis of ROSS as *variants of ROSS*, i.e., the fast versions, or that with size control feature.

The rest of paper is organized as follows. We present the system model in Section 2. Then the formal description of robust clustering problem in CRN and the properties of the problem are given in Section 3. The centralized and distributed solutions are introduced in Section 4 and 5 respectively. Extensive performance evaluation is presented in Section 6. Finally, we conclude our work and point out the direction for future research in Section 7.

2 System Model

We consider a set of cognitive radio users $\mathcal{N}$ and a set of primary users distributed over a given area. The CR users use a set of licensed channels opportunistically, which is denoted as $\mathcal{K}$. 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. We assume that every node can detect the presence of a primary user on each channel with certain accuracy. [2] We adopt the unit disk model [24] for the transmission of both primary and CR users. If a CR node locates within the transmission range of an active primary user, that CR node is not allowed to use the channel which is being used by that primary user. As the result of spectrum sensing, $K_i \subseteq \mathcal{K}$ denotes the set of licensed channels which can be used by i.

We assume that in addition to the licensed channels, there is one dedicated control channel. 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 to all its one hop neighbors $\mathrm{Nb}(i)$, where $\mathrm{Nb}(i)$ is the neighbors of i, and is simply defined as the set of CR nodes located within the transmission range of i. In the following, we refer to the word *channel* in general as licensed channel, unless we mention the control channel explicitly.

If a secondary user i is not in the transmission range of an active primary user p, i can certainly not detect the presence of p. As the transmission range of primary users is limited and secondary users have different locations, different secondary users may have different views of the spectrum availability, i.e., for any $i, j \in \mathcal{N}$, $K_i = K_j$ does not necessarily hold. Then, a cognitive radio network can be represented as a graph $G = (\mathcal{N}, E)$, where $E \subseteq \mathcal{N} \times \mathcal{N}$ such that $\{i, j\} \in E$ if, and only if, there exists a channel $k \in \mathcal{K}$ with $k \in K_i \cap K_{i'}$.

3 Robust Clustering Problem in CRN

In this section, we present the definition of cluster in the context of CRNs, define the clustering problem and discuss its complexity.

3.1 Cluster in CRN

A cluster $C \subseteq \mathcal{N}$ is a set of secondary nodes consisting of a cluster head h_C and a number of cluster members. The cluster head is able to communicate with any cluster member directly. In other terms, for any cluster member $i \in C$, $i \in \mathrm{Nb}(h_C)$ holds.

2. 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 [25], [26], i.e., two neighbouring nodes establish communication on the same channel.

Cluster is denoted as $C(i)$ when its cluster head is i. Cluster size of $C(i)$ is written as $|C(i)|$. $K(C) = \cap_{i\in C}K_i$, $K(C)$ denotes the set of common control channels in cluster C. Clustering is performed periodicity, as secondary users are mobile and the channel availability on secondary users are changing as primary users change their operation state.

TABLE 1. Notations

Symbol	Description
$\mathcal{N}$	collection 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
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 CCCs of cluster $C(i)$
h_C	the cluster head of a cluster C
CH	cluster head
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
C_i	the ith potential cluster (only used in Sec. 4)

3.2 Robust Clustering Problem

The clustering scheme we propose in the paper aims to form clusters with preferred sizes, meanwhile maximize the average number of CCCs per cluster. when perusing the cluster robustness without specifying the requirement on cluster sizes, a clustering method will be inclined to form small clusters. In that case, although more CCCs are obtained per cluster, the benefits brought in by the collective of the cluster members are compromised. Large clusters are not preferred in some scenarios neither, e.g., for the CRN composed with resource limited users, managing the cluster members is a substantial burden. Hence, the cluster size should fall in a desired range [27], [28]. In the following, we present the formal definition of robust clustering problem.

Definition 1: *Robust clustering problem in CRN.*

Given a cognitive radio network $\mathcal{N}$ where $|\mathcal{N}| = N$, secondary users are indexed from 1 to N. A cluster C can be composed with one or multiple secondary users when $|K(C)| > 0$, and any two cluster members can communicate either directly, or through other members belonging to the same cluster. We use $f(C)$ to denote the number of CCCs of a cluster C, and give it a new definition as follows, the number of common control channels is $f(C) = |K(C)|$ if $|C| > 1$, and $f(C) = 0$ when $|C| = 1$. The size of cluster C is not bigger than a positive integer k. The collection of such clusters $\mathcal{S} = \{C_1, C_2, \ldots, C_{|\mathcal{S}|}\}$, satisfies the following condition: $\bigcup_{1\le i\le|\mathcal{S}|} C_i = N$ and $1 \le i \le |\mathcal{S}|$.

The robust clustering problem is to find a subcollection $\mathcal{S}' \subseteq \mathcal{S}$, so that $\sum_{C\in\mathcal{S}'} f(C)$ is maximized, and $\mathcal{S}'$ satisfies the following conditions, $\bigcup_{C_j\in\mathcal{S}'} C_j = \mathcal{N}$ and $C_{j'} \cap C_j = \emptyset$ where $C_{j'}, C_j \in \mathcal{S}'$ and $j' \neq j$.

3.3 Complexity of the Robust Clustering Problem

Based on Definition 1, the decision version of this problem is to ask whether there exists a no-empty $\mathcal{S}' \subseteq \mathcal{S}$, so that $\sum_{C\in\mathcal{S}'} f(C) \geqslant \lambda$ where λ is a real number. We have the following theorem on complexity.

Theorem 3.1: *Robust clustering problem in CRN is NP-hard, when the maximum size of clusters $k \geqslant 3$.*

The proof is in Appendix 19. As it is not likely to generate clusters with size of 2, this theorem actually indicates that the robust clustering problem in CRN is NP-hard. In the following sections, both centralized and distributed solutions will be introduced.

4 Centralized Solution for Robust Clustering

As the robust clustering problem in CRN is NP hard, there is no polynomial solution available. In this paper we formulate the problem into a binary linear programming problem. To make the formulation possible, we adopt heuristics in both the objective function and the constraints, besides, the objective is not totally aligned with the object in the Definition 1.

$$
\begin{aligned}
\min_{x_{ij}} \quad & \Sigma_{j=1}^{N}\Sigma_{i=1}^{G}(-x_{ij}q_{ij} + (1-w_i)*p) \\
\text{subject to} \quad & \Sigma_{i=1}^{G} x_{ij} = 1, for \forall j = 1, \ldots, N \\
& \Sigma_{j=1}^{N} x_{ij} = |C_i| * (1-w_i), for \forall i = 1, \ldots, G \\
& x_{ij} \text{ and } w_i \text{ are binary variables.} \\
& i \in \{1, 2, \cdots G\}, \quad j \in \{1, 2, \cdots N\}
\end{aligned}
$$

The objective is to maximize the sum of the products of cluster size and the corresponding number of CCCs. The second component is designed to be the *punishment* for forming clusters whose sizes are not δ, which is the preferred cluster size. We minimize the opposite of the real objective to make this problem solvable. The constraints guarantee to obtain the clusters which don't overlap. Now we introduce the formulation in details.

$N = |\mathcal{N}|, G = |\mathcal{G}|$. $\mathcal{G}$ is the collective of all the *potential* clusters in a CRN $\mathcal{N}$, and the sizes of these clusters are $1, 2, \ldots, \delta$, where δ is the preferred size for clusters. Potential clusters are the clusters which satisfy the conditions in Section 3.1. Note that the potential clusters include the *singleton* ones, i.e., the cluster which has only one CR node. By permitting the existence of the singleton clusters, we can ensure that the $\mathcal{S}'$ in Definition 1 is always be feasible.

Constant q_{ij} is the element of the constant $G \times N$ matrix $Q_{G\times N}$ which is shown in Figure 1. The subscript i is the index of a potential cluster, j is the node ID of a CR node, $q_{ij} = |K(C_i)|$ when there is $j \in C_i$, and $q_{ij} = 0$ when there is $j \notin C_i$. In other words, each non-zero element q_{ij} denotes the number of CCCs of the cluster i where node j resides. Knowing the meanings of subscripts i and j, it is easy to notice there are $i \in \{1, 2, \cdots, G-1, G\}$ and $j \in \{1, 2, \cdots, N-1, N\}$. The binary variable x_{ij}'s subscripts have the same meanings with that of q_{ij}, thus they have the same scope. x_{ij} indicates whether the CR node j resides in the potential cluster i, i.e., $x_{ij} = 1$ means node j resides in the cluster C_i.

Now let us have a look at the objective function again. Although the related work SOC [21] adopts the product of cluster size and number of CCCs as its sole metric, SOC still generates a large number of singleton clusters and a few large clusters. To refrain the clusters which don't have preferred clusters sizes, the second item is designed, where

$$
\begin{array}{c}
\\ 1 \\ 2 \\ \vdots \\ i \\ \vdots \\ \vdots \\ G
\end{array}
\begin{array}{c}
\begin{array}{cccccccc} 1 & 2 & 3 & \cdots & j & \cdots & N-1 & N \end{array} \\
\begin{pmatrix}
|K(C_1)| & |K(C_1)| & 0 & \cdots & \cdots & \cdots & 0 & 0 \\
|K(C_2)| & 0 & |K(C_2)| & \cdots & \cdots & \cdots & 0 & 0 \\
\vdots & & & \vdots & & \vdots & & \\
0 & |K(C_i)| & 0 & \cdots & \cdots & \cdots & |K(C_i)| & 0 \\
\vdots & & & \vdots & \cdots & \vdots & & \vdots \\
0 & 0 & 0 & \cdots & \cdots & \cdots & |K(C_i')| & 0 \\
|K(C_G)| & \cdots & & \vdots & \cdots & \vdots & & \vdots
\end{pmatrix}
\end{array}
$$

Fig. 1: An example of Matrix Q, its rows correspond to all potential clusters, and columns correspond to the CR nodes in the CRN.

$$
w_i = \begin{cases} 0 & \text{if } i\text{th potential cluster } C_i \text{ is chosen} \\ 1 & \text{if } i\text{th potential cluster } C_i \text{ is not chosen} \end{cases}
$$

w_i is an auxiliary binary variable which denotes whether cluster C_i is chosen by the clustering solution or not. When w_i is zero, it means C_i is chosen by the clustering solution, in this case, the second item in the objective function is p, which is defined as follows,

$$
p = \begin{cases} 0 & \text{if } |C_i| = \delta \\ \alpha_1 & \text{if } |C_i| = \delta - 1 \\ \alpha_2 & \text{if } |C_i| = \delta - 2 \\ \ldots & \end{cases}
$$

where α_x is a positive value, which increases with the divergence between $|C_i|$ and δ, thus these is $\alpha_2 > \alpha_1 > 0$. Because of w_i, any cluster which appears in the clustering solution ($w = 0$) will bring certain *punishment*. Only when that cluster's size is exactly the preferred size δ, the punishment is zero. In contrary, when the chosen cluster's size diverges from δ, the objective function suffers *loss*. In particular, when $w_i = 0$ and $|C_i| = 1$, the punishment is the most severe. With such design, the solution is encouraged not to generate singleton clusters.

As to the constraints, the first constraint restricts each node j to reside in exactly one cluster. The second constraint regulates that when the ith potential cluster C_i is chosen, there will be exactly $|C_i|$ CR nodes residing in cluster C_i.

5 Distributed Clustering Algorithm: ROSS

The centralized robust clustering scheme is inherently not suitable for the CRN which involves dynamics in multiple aspects. Whenever clustering or re-clustering is necessary, which could be triggered by either the mobility of CR users or the variance of the available spectrum, the centralized entity needs to collect the information from all the CR nodes, and then distributes the clustering solution after computation.

In this section we introduce the distributed clustering scheme ROSS. ROSS consists of two cascaded phases: *cluster formation* and *membership clarification*. With ROSS, CR nodes form clusters based on the proximity of the available spectrum in their neighbourhood. Afterwards, CR nodes belong to one certain cluster.

5.1 Phase I - Cluster Formation

After conducting spectrum sensing and communicating with neighbors, every CR node is aware of the channels which are available on themselves and also those on their one hop neighbors. Two metrics are proposed to characterize the proximity in terms of available spectrum between CR node i and its neighborhood:

- *Individual connectivity degree* d_i: $d_i = \sum_{j \in \text{Nb}(i)} |K_i \cap K_j|$. d_i is the total number of the CCCs between node i and every its neighbor. It is an indicator of node i's adhesion to the CRN.
- *neighborhood connectivity degree* g_i: $g_i = |\bigcap_{j \in \text{Nb}(i) \cup i} K_j|$. It is the number of CCCs which are available for i and all its neighbors. g_i 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 2 illustrates an example CRN where every node's connectivity vector is shown.

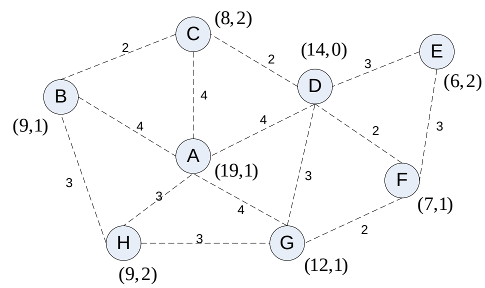


Fig. 2: Connectivity graph of the example CRN and the connectivity vector (d_i, g_i) for each node. The sets of the indices of the available channels sensed by each node are: $K_A = \{1,2,3,4,5,6,10\}, K_B = \{1,2,3,5,7\}, K_C = \{1,3,4,10\}, K_D = \{1,2,3,5\}, K_E = \{2,3,5,7\}, K_F = \{2,4,5,6,7\},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

After introducing the connectivity vector, we proceed to introduce the first phase of algorithm ROSS. In this phase, cluster heads are determined through a series of autonomous negotiations among the CR users, then the neighborhoods of the cluster heads become clusters.

5.1.1 Determining Cluster Heads and Form Clusters

In this phase, 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 $d_j < d_k, \forall k \in \text{Nb}(j) \backslash \{\text{CHs} \cup i\}$, then the node between i and j, which has higher neighborhood connectivity degree will become the cluster head, and the other node 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 a 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 [4]. The pseudo code for the cluster head decision and the initial cluster formation is shown in Algorithm 1 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

4. The reasons for the occurrence of the cluster heads in the neighborhood of a new cluster head will be explained in Section 5.1.2 and 5.1.3)

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 of cluster head eligibility from different CR nodes, d_i is still set to be M. The manipulation of the individual connectivity degree of the cluster members fastens the speed of completing choosing the cluster heads. We have the following theorem to show that as long as a secondary user's individual connectivity degree is greater than zero, every secondary user will eventually be either integrated into a certain cluster, or becomes a cluster head.

THEOREM 5.1: *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 5.1 for one time. The Proof is in Appendix 19.

The procedure of the proof also illustrates the time needed to conduct Algorithm 5.1. Consider an extreme scenario, where all the secondary nodes sequentially execute Algorithm 1, i.e., they constitute a list as discussed in the example in the proof. If one step can be finished within certain time T, then the total time needed for the network to conduct Algorithm 5.1 is $N * T$. In other scenarios, as Algorithm 1 can be executed concurrently by secondary users which locate in different places, the needed time can be considerably reduced. Let us apply Algorithm 1 to the example shown in Figure 2. 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 $\{H, B, A, G\}$.

5.1.2 Guarantee the Availability of Common Control Channel

After executing Algorithm 1, certain formed clusters may don't possess any CCCs. As decreasing cluster size increases the CCCs within a cluster, for those clusters having no CCCs, certain nodes need to be eliminated to obtain at least one CCC. The sequence 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 and resulting in a singleton cluster which is composed by itself. The pseudo code for cluster head to obtain at least one common channel is shown in Algorithm 2. As to the nodes which are eliminated from the previous clusters, they restore their original individual connectivity degrees, executes Algorithm 1, and become either cluster heads or get included into other clusters afterwards according to Theorem 5.1.

During Phase I, when ever 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.

5.1.3 Cluster Size Control in Dense CRN

In this subsection, we illustrate the pressing necessity to control the cluster size when CRN becomes denser.

We consider a cluster $C(i)$ where i is CH in a dense CRN. To make the analysis easier, we assume there is no CHs which are generated within i's neighborhood during the procedure of guaranteeing CCCs. Assuming the CR users and PUs are evenly distributed and PUs occupy the licensed channels randomly, then both CR nodes density and channel availability in the CRN can be seen to be spatially homogeneous. In this case the formed clusters are decided by the transmission range and network density. According to Algorithm 1, the nearest cluster heads could locate just outside node i's transmission range. An instance of this situation is shown in Figure 3. In the figure, black dots represent cluster heads, the circles denotes the transmission ranges of cluster heads. Cluster members are not shown in the figure. Let

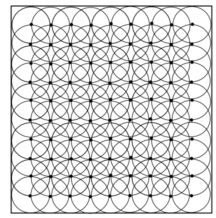


Fig. 3: Clusters formation in extremely dense CRN. Black dots are cluster heads, cluster members are not drawn.

l be the length of side of simulation plan square, and r be CR's transmission radius. Based on the aforementioned analysis and geometry illustration as shown in Figure 3, we give an estimate on the maximum number of generated clusters, which is the product of the number of cluster heads in one row and that number in one line, $l/r * l/r = l^2/r^2$. Given r=10 and l=50, the maximum number of clusters is 25. The number of clusters in the simulation is shown in Figure 4. Simulation is run for 50 times and the confidence interval is 95%. With the increase of CR users, network density (the average number of neighbours) increases linearly, and the number of clusters approaches to 25 which complies with the estimation.

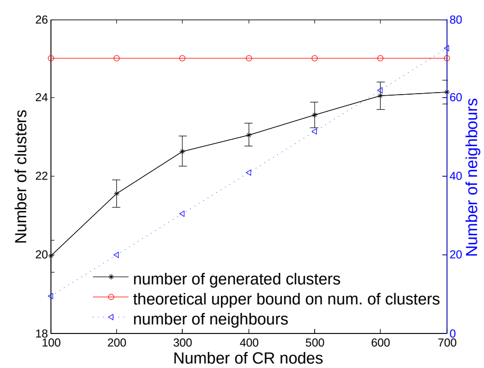


Fig. 4: The correlation between the number of formed clusters and network density.

Both analysis and simulation show that when applying ROSS, after the number of clusters saturates 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 to the cluster heads. To control the cluster size, cluster heads prune their cluster members to achieve the desired cluster size. The desired size δ is decided based on the capability of the CR users and the tasks to be conveyed. Given the desired size δ, a cluster head excludes members sequentially according to the following principle, the absence of one cluster member leads to the maximum increase of common channels within the cluster. This process ends when the size of resultant cluster is δ and at least one CCC is available. This procedure is similar with that to guarantee CCCs in cluster, thus the algorithm can reuse Algorithm 2.

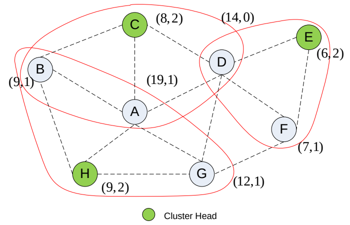


Fig. 5: Clusters formation after the phase I of ROSS. CR nodes A, B, D are debatable nodes as they belong to more than one cluster.

5.2 Phase II - Membership Clarification

After applying phase I of ROSS to the example in Figure 2, the resulted clusters are shown in Figure 5. We notice that nodes A, B, D are included in more than one cluster. We refer 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*.

5.2.1 Distributed Greedy Algorithm (DGA)

Assume a debatable node i needs to decide one cluster $C \in S_i$ to stay, and thereafter leaves the rest others in S_i. In this process, the principle for i is that its move should result in the greatest increase of CCCs in all its claiming clusters. Note that node i is aware of the spectrum availability on all the cluster members of each claiming cluster, thus node i is able to calculate how many more CCCs can be produced in one claiming cluster if i leaves that cluster. If there exists one cluster $C \in S_i$, when i leaves this cluster brings the least increased CCCs than leaving any other claiming clusters, then i chooses to stay in cluster C. When there comes a tie, among the claiming clusters, i chooses to stay in the cluster whose cluster head shares the most CCCs with i. In case there are multiple claiming clusters demonstrating the same on the aforementioned metric, 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 as Algorithm 3. 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.

5.2.2 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 set of 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 [29]. In the past few years, game theory has been extensively applied to problems in communication and networking [30], [31]. Congestion game is an attractive game model which describes the problem where participants compete for limited resources in a non-cooperative manner, it has good property that Nash equilibrium can be achieved after finite steps of best response dynamic, i.e., each player choose strategy to maximizes/minimizes its utility/cost with respect to the other players' strategies. 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 [32], or users downloading files from cloud, etc.

To formulate the debatable nodes' membership clarification into the desired congestion game, we observe this process from a different (or opposite) perspective. From the new perspective, the debatable nodes are regarded to be isolated and don't belong to any cluster, in other words, their claiming clusters become clusters which are beside them. Now for the debatable nodes, the previous problem of deciding which clusters to leave becomes a new problem that which cluster to join. In the new problem, debatable node i chooses one cluster C out of S_i to join if the decrease of CCCs in cluster C is the smallest in S_i, and the decrease of CCCs 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 6.

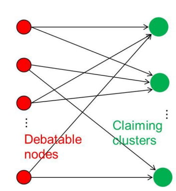


Fig. 6: Debatable nodes and claiming clusters

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 $\Gamma = (\mathcal{P}, \mathcal{R}, \sum_{i, i \in \mathcal{P}}, f)$, and the elements in Γ are explained below,

- $\mathcal{P}$, 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 $\sum_i, i \in \mathcal{P}$ is the set of claiming clusters S_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 decrease of CCCs in cluster C when debatable node i joins C. As to cluster $C \in S_i$, the decrease of CCCs caused by the enrolment of 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 [33]. 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 CCCs $\sum_{i:C \in S_i} \Delta|K^i(C)|$, and a player's strategy in the game is exactly the same with the behaviour of a debatable node in the membership clarification phas.

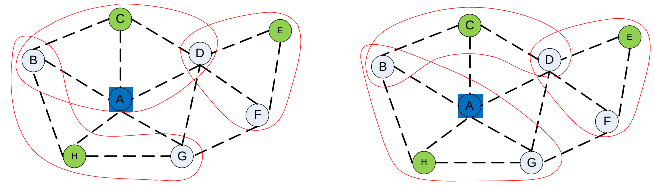
As to singleton congestion game, there exists a pure equilibria which can be reached with the best response update, and the upper bound for the number of steps before convergence is $n^2 * m$ [33], where n is the number of players, and m is the number of resources. In our problem, the players are the debatable nodes, and the resources are the claiming clusters. Thus the upper bound of the number of steps can be expressed as $O(N^3)$.

In fact, the number of steps which are actually involved in this process is much smaller than N^3, as both n and m are considerably smaller than N. The percentage of debatable nodes in $\mathcal{N}$ is illustrated in Figure 14, which is between 10% to 60% of the total number of CR nodes in the network. The number of clusters heads, as discussed in Section 5.1, is dependent on the network density and the CR node's transmission range. As shown in Figure 4, the cluster heads take up only 3.4% to 20% of the total number of CR nodes.

5.2.3 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 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 decision 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 node executes only one loop in Algorithm 3.

Now we apply both ROSS-DGA and ROSS-DFA to the toy network in Figure 5 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 7. In Figure 7(a), node A staying in $C(C)$ and leaving $C(H)$ brings 2 more CCCs to S_A, which is more than that brought by another strategy showed in 7(b). After the decisions made similarly by the other debatable nodes B and D, the final clusters are formed as shown in Figure 8.



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

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

Fig. 7: Membership clarification: possible cluster formations caused by node A's different choices

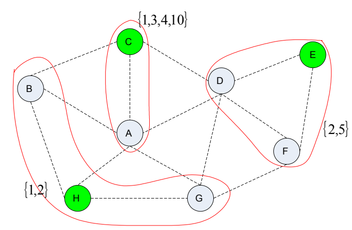


Fig. 8: Final formation of clusters. $K(C(C)), K(C(E)), K(C(H))$ are shown beside corresponding clusters.

5.3 Apply the Centralized and Comparison Distributed Schemes in the Example CRN

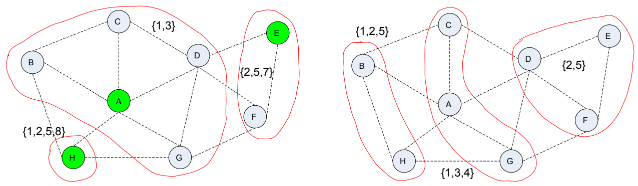
After introducing the resulted clusters from ROSS in Fig. 8, we apply the other robust clustering schemes in the same example CRN in Figure 2. As to the centralized robust clustering scheme, we let the desired cluster size δ be 3. A collection of clusters $\mathcal{G}$ is obtained, which contains all the clusters satisfying the conditions of cluster in Section 3.1 and the sizes of clusters are 1, 2 or 3. $\mathcal{G} = \{\{A\}, \{B\}, \ldots, \{B, C\}, \{B, A\}, \{B, H\}, \cdots, \{B, A, C\}, \{B, H, C\}, \{A, D, C\}, \cdots\}$, and $G = |\mathcal{G}| = 38$. When α_1 and α_1 are set as 0.2 and 0.8, the formed clusters are shown in Fig. 9(b). The resulted clustering solutions from SOC is shown in Fig. 9(a).

As to the average number of CCCs, the results of ROSS (including both ROSS-DGA and ROSS-DFA), centralized and SOC are 2.66, 2.66, and 3 respectively. Note there is one singleton cluster $C(H)$ generated by SOC, which is not preferred. When we take no account of the singleton clusters, then the average number of common channels of SOC drops to 2.5.

6 Performance Evaluation

The schemes involved in the simulation are listed as follows,

- ROSS without size control, i.e., ROSS-DGA and ROSS-DFA.
- ROSS-δ-DGA and ROSS-δ-DFA, which are the variants of ROSS with the size control feature. δ is the preferred cluster size.
- SOC [21], a distributed clustering scheme pursuing cluster robustness.
- Centralized robust clustering scheme. The formulated optimization is an integer linear optimization problem, which is solved by the function *bintprog* provided in MATLAB.



(a) Resulted from SOC

(b) Resulted from the centralized clustering scheme

Fig. 9: Final clusters formed in the example network when being applied with SOC and the centralized clustering scheme.

The authors of [21] compared SOC with other schemes in terms of the average number of CCCs 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 [34], or for CRN but just considering connection among CR nodes [7]. Thus SOC is chosen to be the only distributed scheme as comparison, besides, we also compare ROSS with the centralized scheme. We investigate the schemes with the following metrics.

- **Average number of CCCs per non-singleton cluster.** Non-singleton cluster refers the cluster whose cluster size is larger than 1. Comparing with the metric adopted by SOC [21], which is the average number of CCCs per cluster without excluding the singleton clusters, this metric provides an more accurate description of the robustness of the non-singleton cluster. The larger number of CCCs per non-singleton clusters means these clusters have higher probability to survive when the primary users' operation becomes more intense. Although this metric doesn't disclose the information about the CR nodes which are not included in any non-singleton clusters, we still examine this metric as the number of CCCs is involved in the utility which is adopted by all robust clustering schemes.
- **Cluster sizes.** The distribution of CRs residing in the formed clusters with different sizes is presented.
- **Robustness of the clusters against newly added PUs.** We increase the number of PUs to challenge the clusters, and count the number of *unclustered* CR nodes which are the synonyms of the singleton clusters. This metric 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, in the first part, we investigate the performance of centralized scheme, and the gap between the distributed schemes and the centralized scheme. This part is conducted 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.

We give a brief introduction to the settings which are common 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%. 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.

6.1 Centralized Schemes vs. Decentralized Schemes

There are 10 primary users and 20 CR users are 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, the parameters used in the *punishment* for choosing the clusters with undesired sizes are set as follows, $\alpha_1 = 0.4$, $\alpha_2 = 0.6$.

6.1.1 Number of CCCs in Non-singleton Clusters

From Figure 10, we can see the centralized schemes outperform the distributed schemes. As to the distributed schemes, SOC achieves the best than all the variants of ROSS. The reason is, SOC is liable to group the neighbouring CRs which share the most abundant spectrum together, no matter how many of them are, thus the number of CCC of the formed clusters is higher. We have discussed the flaw of this metric as it doesn't convey the number of unclustered CR nodes, in fact, SOC generates the most unclustered CRs, which can be seen when we discuss the performance on the number of unclustered CR nodes. As to the variants of ROSS, we notice that the greedy mechanism increases CCCs in non-singleton clusters significantly.

6.1.2 Cluster Size Control

Figure 11 depicts the empirical cumulative distribution of the CRs residing in certain sized clusters, from which we have two conclusions. The first, SOC generates more unclustered CR nodes than other schemes. The centralized schemes produce no unclustered CR nodes, ROSS-DGA/DFA generate 3% unclustered 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 5.1.2 and section 5.1.3). Second, the centralized schemes and cluster size control mechanism of RPSS generate clusters which satisfy the requirement on cluster size strictly. 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.

6.1.3 Robustness of the clusters against newly added PUs

We add PUs in the CRN to decrease the available spectrum, and observe the number of unclustered CR nodes. Clusters are formed with the presence of 10 PUs in the beginning, then extra 20 batches of PUs are added sequentially, where each batch includes 5 PUs.

Figure 12 shows with the increase of PUs, certain clusters disappear and the number of unclustered CR nodes increases. Among the robust clustering schemes, the centralized scheme with desired size of two generates the most robust clusters, meanwhile, SOC results in the most vulnerable clusters. The centralized scheme with desired size of three doesn't outperform the variants of ROSS, because the centralized scheme peruses clusters with size of 3 at the expenses of sanctifying CCCs. In contrary, the variants of ROSS generate some smaller clusters which are more likely to survive when PUs' activities become intense.

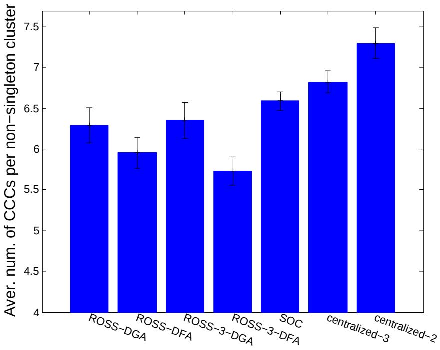


Fig. 10: Number of common channels of non-singleton clusters

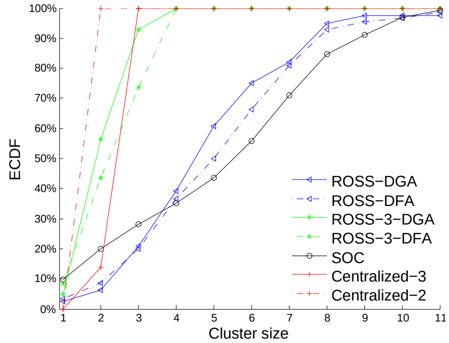


Fig. 11: Cumulative distribution of CRs residing in clusters with different sizes

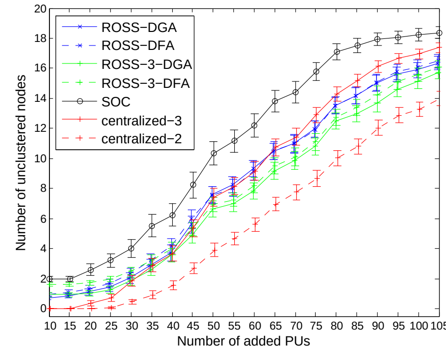


Fig. 12: Number of unclustered CRs with decreasing spectrum availability

Fig. 13: Comparison between the distributed and centralized clustering schemes ($N = 20$)

6.1.4 Control Signalling Overhead

In this section we compare the overhead of signalling involved in different clustering schemes. We omit the control messaged involved in neighbourhood discovery, which is the premise for any clustering scheme. According to [35], the message complexity is defined as the number of messages used by all nodes. To have the same metric to compare, we count *the number of transmissions of control messages*, without distinguishing they are sent with broadcast or unicast. This metric is synonymous with *the number of updates* discussed in Section 5.

As to ROSS, the control messages are generated in both phases. In the first phase, when a CR node decides itself to be the cluster head, it broadcasts a message containing its ID, cluster members and the set of CCCs in its cluster. In the second phase, a debatable node broadcasts its affiliation to inform its claiming clusters, then the CHs of the claiming clusters broadcast message about the new cluster members if they are changed due to the debatable node's decision. The total number of the decisions involved in cluster formation has been analysed in Theorem 5.1 and Section 5.2.2 respectively.

Comparison scheme SOC involves three rounds of execution. In the first two rounds, every CR node maintains its own cluster and seek to integrate neighbouring clusters, or joins one of them. The final clusters are obtained in the third round. In each round, every CR node is involved in comparisons and cluster mergers.

The centralized scheme is conducted at the centralized control device, but it involves two phases of control message transmission. The first phase is information aggregation, in which every CR node's channel availability and neighborhood is transmitted to the centralized controller. The second phase is broadcasting, where the clustering solution is disseminated to every CR node. We adopt the algorithm proposed in [36] to broadcast and gather information as the algorithm is simple and self-stabilizing. This scheme needs building a backbone structure to support the communication, and we apply ROSS to generate cluster heads which serve as the backbone, besides, the debatable nodes as used as the gateway nodes between the backbone nodes. As the backbone is built once and support transmission for multiple times, the messages involved in the clustering process are not counted. As to the process of information gathering, we assume that every cluster member sends the spectrum availability and its ID to its cluster head, which further forwards or the message to the controller, thus the number of transmission is N. As to the process of dissemination, in an extreme situation where all the gateway and the backbone nodes broadcast, the number of transmission is $h + m$, where h is the number of cluster heads and m is number of debatable nodes.

The number of control messages which are involved in both ROSS and the centralized scheme is related with the number of debatable nodes. Figure 14 shows the percentage of debatable nodes when the CRN network becomes denser, from which we can obtain the value of m. The message complexity, quantitative

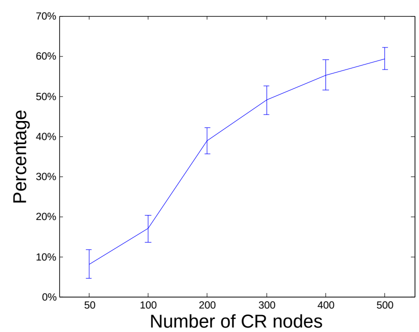


Fig. 14: The percentage of debatable nodes after phase I of ROSS.

analysis of the number of control messages involved in clustering, and the size of control messages are shown in Table 2. Figure 15 shows the number of transmissions of SOC, the upper bound of the number of transmissions for ROSS, and the analytical number of transmissions of the centralised scheme.

6.2 Comparison among the Distributed Schemes

In this section we investigate the performances of distributed clustering schemes in CRN with different network scales and densities. The transmission range of CR is $A/5$, PR's transmission range is $2A/5$. The initial number of PU is 30. The desired sizes adopted are listed in the Table 3, which is about 60% of the average number of neighbours.

6.2.1 Number of CCCs per Non-singleton Clusters

Figure 16 shows the average number of CCCs of the non-singleton clusters. We notice that SOC achieves the most CCCs per non-

TABLE 2. Signalling overhead

Scheme	Message Complexity	Quantitative number of messages	Content of message (size of message)
ROSS-DGA, ROSS-δ-DGA	$O(N^3)$ (worst case)	$h + 2 * m^2 d$ (upper bound)	Cluster head broadcasts cluster composition ($\|C(i)\| * \|\mathcal{K}\|$ bytes). Cluster members broadcast the manipulated new individual connectivity degree (1 byte)
ROSS-DFA, ROSS-δ-DFA	$O(N)$ (worst case)	$h + 2m$ (upper bound)	
SOC	$O(N)$	$3 * N$	$C(i)$ ($\|C(i)\|$ bytes) and $K(C(i))$ ($\|\mathcal{P}\|$ bytes), $i \in \mathcal{N}$
Centralized	$O(N)$	$h + m + N$ (upper bound) [36]	$\{C\}$ ($\|C_i\| * N$ bytes)

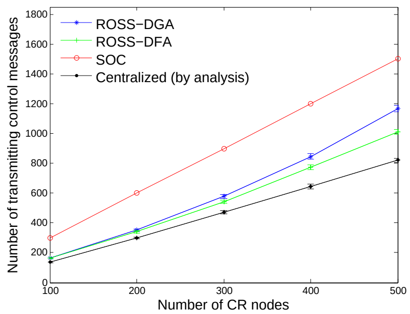


Fig. 15: Number of control messages. Note the curves of ROSS-DGA and ROSS-DFA are the upper bounds

TABLE 3

Number of CRs	100	200	300
Average num. of neighbours	9.5	20	31
Desired size δ	6	12	20

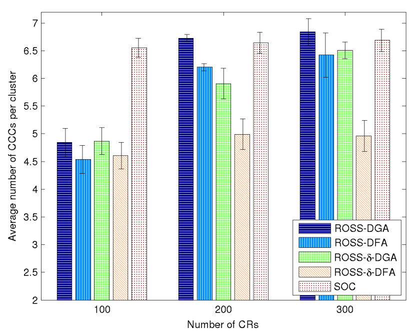


Fig. 16: Number of common channels of non-singleton clusters.

singleton cluster, although the lead over the variants of ROSS shrinks significantly when N increases.

6.2.2 Robustness of the clusters against newly added PUs

We add extra 20 batches of PUs sequentially in the CRN, where each batch includes 10 PUs. Figure 17 and 18 show that when $N = 100$ and 200, more unclustered CR nodes appear in the CRN where SOC is applied. When the network becomes denser, ROSS-DGA/DFA generate slightly more unclustered CR nodes than SOC when new PUs are not many, but SOC's performance deteriorates quickly when the number of PUs becomes larger. We only show the average values of the variants of ROSS as their confidence intervals overlap. When applying ROSS with size control mechanism, significantly less unclustered CR nodes are generated. Besides, the greedy mechanism moderately strengthens the robustness of the clusters.

6.2.3 Cluster Size Control

Figure 25 shows when the network density scales up, the number of formed clusters by ROSS increases by smaller margin, and that generated by SOC increases linearly. This result coincides with the analysis in Section 5.1.3. To better understand the distribution of the sizes of formed clusters, we depict the empirical cumulative distribution of CR nodes which reside in clusters with certain sizes in CRNs in Figures 21 22 23.

The sizes of clusters generated by ROSS-DGA and ROSS-DFA span a wider range than ROSS with size control feature. Most of the generated clusters are smaller than the average number of neighbours, which is roughly equal with the 95% percentile of the ROSS-DGA curve. The 50% percentile of the ROSS-DGA curve is roughly the desired size δ. When the variants of ROSS with size control feature are applied, the sizes of the most generated clusters are smaller than δ. As to the curves of SOC, the 95% percentiles are 36, 30, and 40 in respective networks.

Overviewing Figure 24, we conclude that the sizes of the clusters generated by ROSS are limited by the network density, the sizes of the clusters formed by ROSS with size control feature are restricted by the desired size. In contrary, the clusters generated from SOC demonstrate strong divergence on cluster sizes.

6.3 Insights from the Simulation

The centralized clustering scheme can form the clusters which satisfy the requirement on cluster size strictly, and the clusters are robust against the newly added PUs, besides, it generates the smallest control overhead in the process of cluster formation.

As distributed schemes, ROSS outperform the comparison scheme SOC considerately on three metrics. ROSS generates much less singleton clusters than SOC, and the resulted clusters are robuster than SOC in front of newly added PUs. The signalling overhead involved in ROSS is about half of that needed for SOC, and the signalling messages are much shorter that the latter. The sizes of the clusters which are generated by ROSS are restricted by the network density, which demonstrate smaller discrepancy than that of SOC. Besides, ROSS achieves similar performance to the centralized scheme, i.e., the cluster sizes obtained by ROSS with size control feature are limited by the desired size, and the cluster robustness is similar when applying the variants of ROSS and the centralized scheme respectively. As to the variants of ROSS, the greedy mechanism in ROSS-DGA helps to improve the performance on cluster size and cluster robustness at the cost of mildly increased signalling overhead.

We also notice that the metric that the number of CCCs per non-singleton cluster doesn't indicate the robustness of clusters as

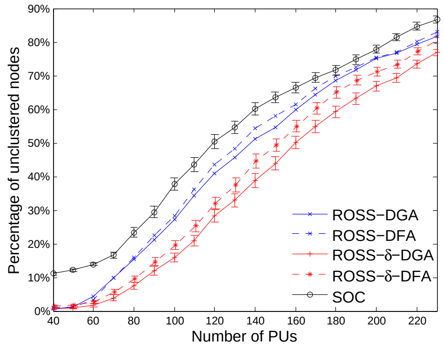


Fig. 17: 100 CRs

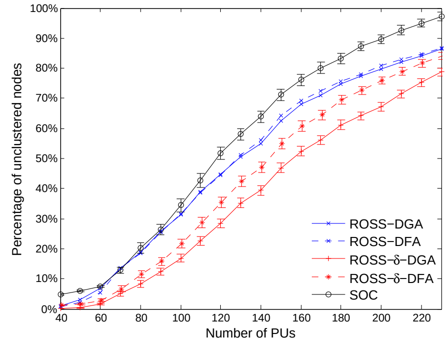


Fig. 18: 200 CRs

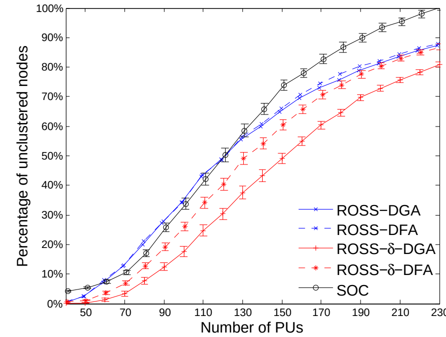


Fig. 19: 300 CRs

Fig. 20: Percentage of CR nodes which are not included in any non-singleton clusters

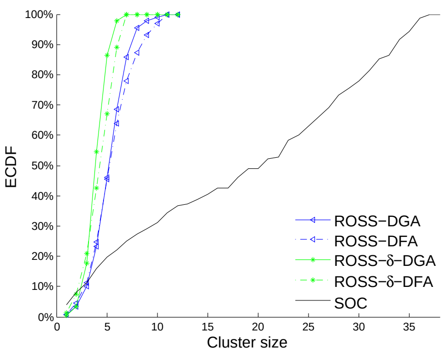


Fig. 21: 100 CRs, 30 PUs in network

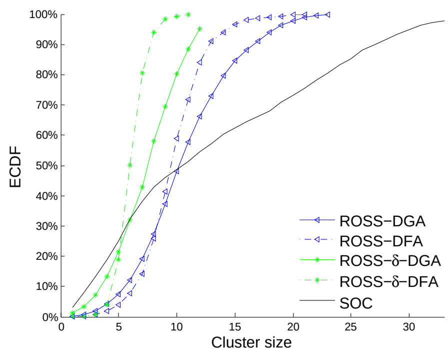


Fig. 22: 200 CRs, 30 PUs in network

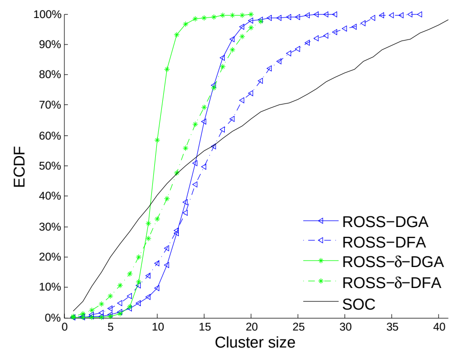


Fig. 23: 300 CRs, 30 PUs in network

Fig. 24: Cumulative distribution of CRs residing in clusters with different sizes

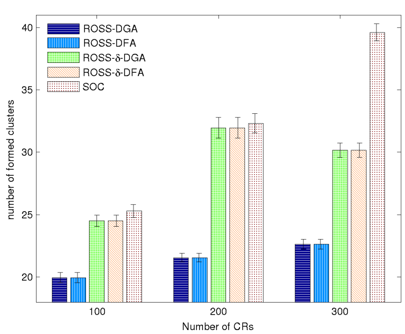


Fig. 25: The number of formed clusters.

shown in Figure 12 and 20, although it is adopted as the metric in the formation of clusters.

7 Conclusion

In this paper we investigate the robust clustering problem in CRN extensively. The proposed scheme results in a cluster structure in CRN, which has longer time expectancy against the primary users' activity. Besides, the clusters generated have similar sizes compared with other distributed schemes. We provide the mathematical description of the problem and prove the NP hardness of it. We propose both centralized and distributed schemes, where the congestion game model in game theory is used to design the distributed schemes. Through simulation and theoretical analysis, we find that distributed schemes achieve similar performance with centralized optimization in terms of cluster robustness, signaling overhead and cluster sizes, and outperform the comparison distributed scheme on the above mentioned metrics.

The shortage of ROSS scheme is it doesn't generate big clusters whose sizes exceed the cluster head's neighborhood. This problem is attributed to fact that ROSS forms clusters on the basis of cluster head's neighborhood, and doesn't involve interaction with the nodes outside its neighborhood. In the other way around, forming big cluster which extends a cluster head's neighborhood has limited applications, as multiple hop communication and coordination are required within such clusters.

Proof of Theorem 5.1

Proof. We consider a CRN which can be represented as a connected graph. To simplify the discussion, we assume the secondary users have unique individual connectivity degrees. Each user has an identical ID and a neighborhood connectivity degree. This assumption is fair as the neighborhood connectivity degrees and node ID are used to break ties in Algorithm 1, when the individual

Algorithm 1: ROSS phase I: cluster head determination and initial cluster formation for CR node i

Input: $d_j, g_j, j \in \mathrm{Nb}_i \setminus \mathrm{CHs}$. Empty sets τ_1, τ_2
Result: Returning 1 means i is cluster head, then d_j is set to 0, $j \in \mathrm{Nb}_i \setminus \mathrm{CHs}$. returning 0 means i is not CH.

if $\nexists j \in Nb_i \setminus CHs$, *such that* $d_i \geq d_j$ **then**
  return 1;
end
if $\exists j \in Nb_i \setminus CHs$, *such that* $d_i > d_j$ **then**
  return 0;
else
  if $\nexists j \in Nb_i \setminus CHs$, *such that* $d_j == d_i$ **then**
    $\tau_1 \leftarrow j$
  end
end
if $\nexists j \in \tau_1$, *such that* $g_i \leq g_j$ **then**
  return 1;
end
if $\exists j \in \tau_1$, *such that* $g_i < g_j$ **then**
  return 0;
else
  if $\nexists j \in \tau_1$, *such that* $g_j == g_i$ **then**
    $\tau_2 \leftarrow j$
  end
end
if $\mathtt{ID}_i$ *is smaller than any* $\mathtt{ID}_j$, $j \in \tau_2 \setminus i$ **then**
  return 1;
end
return 0;

Algorithm 2: ROSS phase I: cluster head guarantees the availability of CCC (start from line 1) / cluster size control (start from line 2)

Input: Cluster C, empty sets τ_1, τ_2
Output: Cluster C has at least one CCC, or satisfies the requirement on cluster size

1 **while** $K_C = \emptyset$ **do**
2   **while** $|C| > \delta$ **do**
3   **if** $\exists$ *only one* $i \in C \setminus H_C$, $i = \arg\min(|K_{H_C} \cap K_i|)$ **then**
4       $C = C \setminus i$;
5   **else**
6       $\exists$ multiple i which satisfies $i = \arg\min(|K_{H_C} \cap K_i|)$;
7       $\tau_1 \leftarrow i$;
8   **end**
9   **if** $\exists$ *only one* $i \in \tau_1$, $i = \arg\max(|\cap_{j \in C \setminus i} K_j| - |\cap_{j \in C} K_j|)$ **then**
10       $C = C \setminus i$;
11   **else**
12       $C = C \setminus i$, where $i = \arg\min_{i \in \tau_1} \mathtt{ID}_i$
13   **end**
14   **end**
15 **end**

Algorithm 3: Debatable node i decides its affiliation in phase II of ROSS

Input: all claiming clusters $C \in S_i$
Output: one cluster $C \in S_i$, node i notifies all its claiming clusters in S_i about its affiliation decision.

while *i has not chosen the cluster, or i has joined cluster $\tilde{C}$, but $\exists C' \in S_i, C' \neq \tilde{C}$, which has* $|K(C' \setminus i)| - |K(C')| < |K(C \setminus i)| - |K(C)|$ **do**
  if $\exists$ *only one* $C \in S_i$, $C = \arg\min(|K(C \setminus i)| - |K(C)|)$ **then**
    return C;
  else
    $\exists$ multiple $C \in S_i$ which satisfies $C = \arg\min(|K(C \setminus i)| - |K(C)|)$;
    $\tau_1 \leftarrow C$;
  end
  if $\exists$ *only one* $C \in \tau_1$, $C = \arg\max(K_{h_C} \cap K_i)$ **then**
    return C;
  else
    $\exists$ multiple $C \in S_i$ which satisfies $C = \arg\max(K_{h_C} \cap K_i)$;
    $\tau_2 \leftarrow C$;
  end
  if $\exists$ *only one* $C \in \tau_2$, $C = \arg\min |C|$) **then**
    return C;
  else
    return $\arg\min_{C \in \tau_2} h_C$;
  end
end

connectivity degrees are unique, it is not necessary to use the former two metrics.

For the sake of contradiction, let us assume there exist some secondary user α which is not included into any cluster. Then there is at least one node $\beta \in \mathrm{Nb}_\alpha$ such that $d_\alpha > d_\beta$. According to Algorithm 1, δ is not included in any clusters, because otherwise $d_\beta = M$, a large positive integer. Now, we distinguish between two cases: If β becomes cluster head, node α is included, the assumption is not true. If β is not a cluster head, then β is not in any cluster, we can repeat the previous analysis made on node α, and deduce that node β has at least one neighbouring node γ with $d_\gamma < d_\beta$. Till now, when there is no cluster head identified, the unclustered nodes, i.e., α, β form a linked list, where their connectivity degrees monotonically decrease. But this list will not continue to grow, because the minimum individual connectivity degree is zero, and the length of this list is upper bounded by the total number of nodes in the CRN. An example of the formed node series is shown as Figure 26.

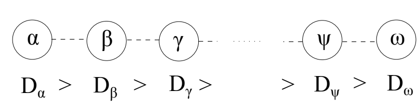


Fig. 26: The node series discussed in the proof of Theorem 5.1, the deduction begins from node α

In this example, node ω is at the tail of the list. As ω does not have neighboring nodes with lower individual connectivity degree, ω becomes a cluster head. Then ω incorporates all its one-hop neighbours (here we assume that every newly formed cluster has common channels), including the nodes which precede ω in the list. The nodes which join a cluster set their individual connection degrees to M, which enables the node immediately precede in the list to become a cluster head. In this way, cluster heads are generated from the tail of list to the head of the list, and all the nodes in the list are in at least one cluster, which contradicts the assumption that α is not included in any cluster.

If we see a secondary user *becoming a cluster head*, or *becoming a cluster member* as one step, as the length of the list of secondary users is not larger than N, there are N steps for this scenario to form the initial clusters.

□

Proof of Theorem 3.1

Proof. We put the definition of weighted k-set packing problem here as it to be used in the analysis on the complexity of the centralised clustering problem.

Definition 2: *Weighted k-set packing.*

Given a finite set $\mathcal{G} = \{g_1, \ldots, g_N\} \subsetneq \mathbb{N}$ of non-negative integers and a collection of sets $\mathcal{Q} = \{S_1, S_2, \cdots, S_m\}$ such that $S_i \subseteq \mathcal{G}$ for every $1 \le i \le N$. Each set $S \in \mathcal{Q}$ is associated with a weight $\omega(S) \in \mathbb{R}$. Further, we are given a threshold value $\lambda \in \mathbb{N}$.
The question is whether there exists a collection $\mathcal{S} \subseteq \mathcal{Q}$ such that $\mathcal{S}$ contains only pairwise disjoint sets, i.e., for all $S, S' \in \mathcal{S}$ with $S \neq S'$ it holds that $S \cap S' = \emptyset$, and the total weight of the sets in $\mathcal{S}$ is greater than λ, i.e., $\sum_{S \in \mathcal{S}} \omega(S) > \lambda$.

Weighted k-set packing is NP-hard when $k \geqslant 3$. [37]

To prove that the centralized clustering problem is NP-hard, we reduce the NP-hard problem *weighted k-set packing* to it to prove the former is at least as hard as the latter. We show the existence of a polynomial-time algorithm σ that transforms any instance $\mathcal{I}$ of a weighted k-set packing into an instance $\sigma(\mathcal{I}))$ of centralized clustering problem.

W.l.o.g. let set $\mathcal{G} = \{1, \ldots, N\}$. The polynomial algorithm σ consists of three steps.

- In the first step, we transform the instance $\mathcal{I}$ to $\mathcal{I}'$ by adding dummy elements into each set in $\mathcal{I}$. Formally, for each set $s_i \in \mathcal{I}$, we create $s_i' = s_i \times \{1, 2\}$. Then we change the dummy elements by adding N on them. The purpose of this transformation is to eliminate the set in $\mathcal{I}$, which has single element. The weight of these sets remain unchanged, i.e., $\omega(s_i') = \omega(s_i)$.
- Mapping the elements in $\mathcal{I}'$ to CR nodes on a two-dimensional Euclidean plane, and mapping the sets in $\mathcal{I}'$ to clusters. There is a bijection between the union of sets in $\mathcal{I}'$ to the CRN which consists of the CR nodes, in particular, each element one-to-one corresponds to the CR node whose node ID equals to the element. We further regulate that, each set in $\mathcal{I}'$ has a corresponding cluster, i.e., there is a bijection between a set in $\mathcal{I}'$ and a set of node IDs in a cluster, besides, the number of CCCs is equal to the weight of the corresponding set.
- In this step, we transform the instance $\mathcal{I}'$ to a clustering proposal for CRN. We adjust the locations of the CR nodes which are mapped from the dummy elements in the instance $\mathcal{I}'$, and let them beside the CR nodes whose ID is exact N smaller. Because of the dummy nodes, the clustering solution which corresponds to $\mathcal{I}'$ doesn't have singleton cluster. This transformation requires $2 \cdot \sum_{s_i \in \mathcal{I}'} |s_i|$ steps. Now we look back to the elements in $\mathcal{Q}$, which don't appear in $\mathcal{I}'$. We one-to-one map these elements to CR nodes, and arbitrarily put these CR nodes in the plane, and these CR nodes become single node clusters. According to the definition of clustering problem in CRN, the number of CCCs in these single node clusters is 0. These singleton clusters and the clusters mapped from $\mathcal{I}'$ constitute a clustering proposal, and finding these singleton clusters requires at most N steps. An example is shown in Table 4.

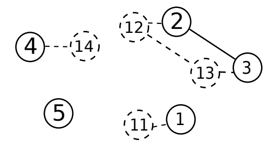
$\mathcal{N}$	$\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$
$\mathcal{Q}$	$\{\{1\}, \{1, 5\}, \{1, 2, 4\}, \{2, 3\}, \{4\}\}$
Instance for Weighted k-set packing	$\{\{1\}, \{2, 3\}, \{4\}\}$
Instance with dummy elements	$\{\{1, 11\}, \{2, 12, 3, 13\}, \{4, 14\}\}$
Instance for clustering solution (dashed circles are dummy nodes)	

TABLE 4

We have crossed the hurdle of finding one polynomial algorithm σ which transforms an instance of weighted k-set packing to an instance of the clustering problem in CRN. When $\mathcal{I}$ is not an instance for weighted k-set packing problem due to the existence of joint sets, the corresponding cluster strategy is not a solution for the centralized clustering problem, as there are overlapped clusters.

When there are only disjoint sets in an instance $\mathcal{I}$ for weighted k-set packing, the sum weight is identical to the sum number of CCCs in the CRN mapped from $\mathcal{I}'$, even $\mathcal{I}$ contains sets which only have one element. Thus, when a instance $\mathcal{I}$ for k-set packing problem is true, i.e., the sum of weights is greater than λ, then in the CRN which is mapped from $\mathcal{I}'$, the summed number of CCCs of the clusters is greater than λ. In the other way around, when an instance if false for weighted k-set packing problem, the summed number of CCCs of the clusters in the mapped CRN is smaller than λ, and there is no clustering solution satisfying the centralized clustering problem.

Thus, weighted k-set packing can be reduced to clustering problem in CRN, then the latter problem is of NP-hard. □

Acknowledgment

The authors would like to thank xxxx

References

[1] I. Mitola, J. and J. Maguire, G.Q., "Cognitive radio: making software radios more personal," *Personal Communications, IEEE*, vol. 6, no. 4, pp. 13 –18, aug. 1999.
[2] T. Yucek and H. Arslan, "A survey of spectrum sensing algorithms for cognitive radio applications," *IEEE Communications Surveys Tutorials*, vol. 11, no. 1, pp. 116–130, First 2009.
[3] Q. Zhao and B. Sadler, "A survey of dynamic spectrum access," *Signal Processing Magazine, IEEE*, vol. 24, no. 3, pp. 79–89, May 2007.
[4] A. Sahai, R. Tandra, S. M. Mishra, and N. Hoven, "Fundamental design tradeoffs in cognitive radio systems," in *Proc. of the 1st Intl. workshop on technology and policy for accessing spectrum*, ser. ACM TAPAS '06.
[5] I. F. Akyildiz, B. F. Lo, and R. Balakrishnan, "Cooperative spectrum sensing in cognitive radio networks: A survey," *Phys. Commun.*, vol. 4, no. 1, pp. 40–62, Mar. 2011.

[6] C. Sun, W. Zhang, and K. B. Letaief, “Cluster-based cooperative spectrum sensing in cognitive radio systems,” in *proc. of IEEE ICC*, 2007, pp. 2511–2515.

[7] J. Zhao, H. Zheng, and G.-H. Yang, “Spectrum sharing through distributed coordination in dynamic spectrum access networks,” *Wireless Com. and Mobile Computing*, vol. 7, no. 9, pp. 1061–1075, 2007.

[8] D. Willkomm, M. Bohge, D. Hollós, J. Gross, and A. Wolisz, “Double hopping: A new approach for dynamic frequency hopping in cognitive radio networks,” in *Proc. of PIMRC 2008*.

[9] C. Passiatore and P. Camarda, “A centralized inter-network resource sharing (CIRS) scheme in IEEE 802.22 cognitive networks,” in *Proc. of IFIP Annual Mediterranean Ad Hoc Networking Workshop 2011*, 2011.

[10] A. A. Abbasi and M. Younis, “A survey on clustering algorithms for wireless sensor networks,” *Comput. Commun.*, vol. 30, no. 14-15, pp. 2826–2841, 2007.

[11] Q. Wu, G. Ding, J. Wang, X. Li, and Y. Huang, “Consensus-based decentralized clustering for cooperative spectrum sensing in cognitive radio networks,” *Chinese Science Bulletin*, vol. 57, no. 28-29, 2012.

[12] H. D. R. Y. Huazi Zhang, Zhaoyang Zhang1 and X. Chen, “Distributed spectrum-aware clustering in cognitive radio sensor networks,” in *Proc. of GLOBECOM 2011*.

[13] B. E. Ali Jorio, Sanaa El Fkihi and D. Aboutajdine, “An energy-efficient clustering routing algorithm based on geographic position and residual energy for wireless sensor network,” *Journal of Computer Networks and Communications*, vol. 2015, April 2015.

[14] V. Kawadia and P. R. Kumar, “Power control and clustering in ad hoc networks,” in *Proc. of INFOCOM ’03*, 2003, pp. 459–469.

[15] M. Krebs, A. Stein, and M. A. Lora, “Topology stability-based clustering for wireless mesh networks,” in *Global Telecommunications Conference (GLOBECOM 2010), 2010 IEEE*, Dec 2010, pp. 1–5.

[16] T. Chen, H. Zhang, G. Maggio, and I. Chlamtac, “Cogmesh: A cluster-based cognitive radio network,” *Proc. of DySPAN ’07*.

[17] K. Baddour, O. Ureten, and T. Willink, “Efficient clustering of cognitive radio networks using affinity propagation,” in *Proc. of ICCCN 2009*.

[18] D. Wu, Y. Cai, L. Zhou, and J. Wang, “A cooperative communication scheme based on coalition formation game in clustered wireless sensor networks,” *IEEE Transactions on Wireless Communications,*, vol. 11, no. 3, pp. 1190 –1200, march 2012.

[19] A. Asterjadhi, N. Baldo, and M. Zorzi, “A cluster formation protocol for cognitive radio ad hoc networks,” in *Proc. of European Wireless Conference 2010*, pp. 955–961.

[20] M. Ozger and O. B. Akan, “Event-driven spectrum-aware clustering in cognitive radio sensor networks,” in *Proc. of IEEE INFOCOM 2013*.

[21] S. Liu, L. Lazos, and M. Krunz, “Cluster-based control channel allocation in opportunistic cognitive radio networks,” *IEEE Trans. Mob. Comput.*, vol. 11, no. 10, pp. 1436–1449, 2012.

[22] N. Mansoor, A. Islam, M. Zareei, S. Baharun, and S. Komaki, “Construction of a robust clustering algorithm for cognitive radio ad-hoc network,” in *Proc. of CROWNCOM 2015*.

[23] D. Li and J. Gross, “Robust clustering of ad-hoc cognitive radio networks under opportunistic spectrum access,” in *Proc. of IEEE ICC ’11*.

[24] B. Clark, C. Colbourn, and D. Johnson, “Unit disk graphs,” *Annals of Discrete Mathematics*, vol. 48, no. C, pp. 165–177, 1991.

[25] Y. Zhang, G. Yu, Q. Li, H. Wang, X. Zhu, and B. Wang, “Channel-hopping-based communication rendezvous in cognitive radio networks,” *IEEE/ACM Transactions on Networking*, vol. 22, no. 3, pp. 889–902, June 2014.

[26] Z. Gu, Q.-S. Hua, and W. Dai, “Fully distributed algorithms for blind rendezvous in cognitive radio networks,” in *Proceedings of the 2014 ACM MobiHoc*, ser. MobiHoc ’14.

[27] Y. P. Chen, A. L. Liestman, and J. Liu, “Clustering algorithms for ad hoc wireless networks,” in *Ad Hoc and Sensor Networks. Nova Science Publishers*, 2004.

[28] E. Perevalov, R. S. Blum, and D. Safi, “Capacity of clustered ad hoc networks: how large is "large"?” *IEEE Transactions on Communications*, vol. 54, no. 9, pp. 1672–1681, Sept 2006.

[29] A. MacKenzie and S. Wicker, “Game theory in communications: motivation, explanation, and application to power control,” in *Proc. of IEEE GLOBECOM 2001*.

[30] J. O. Neel, “Analysis and design of cognitive radio networks and distributed radio resource management algorithms,” Ph.D. dissertation, Blacksburg, VA, USA, 2006, aAI3249450.

[31] B. Wang, Y. Wu, and K. R. Liu, “Game theory for cognitive radio networks: An overview,” *Comput. Netw.*, vol. 54, no. 14, pp. 2537–2561, Oct. 2010.

[32] B. J. S. Chee and C. Franklin, Jr., *Cloud Computing: Technologies and Strategies of the Ubiquitous Data Center*, 1st ed. CRC Press, Inc., 2010.

[33] H. Ackermann, H. RÃűglin, and B. VÃűcking, “Pure Nash equilibria in player-specific and weighted congestion games,” *Theoretical Computer Science*, vol. Vol. 410, no. 17, pp. 1552 – 1563, 2009.

[34] S. Basagni, “Distributed clustering for ad hoc networks,” *Proc. of I-SPAN ’99*, pp. 310 –315, 1999.

[35] X.-Y. Li, Y. Wang, and Y. Wang, “Complexity of data collection, aggregation, and selection for wireless sensor networks,” *IEEE Transactions on Computers*, vol. 60, no. 3, pp. 386–399, 2011.

[36] M. Onus, A. Richa, K. Kothapalli, and C. Scheideler, “Efficient broadcasting and gathering in wireless ad-hoc networks,” in *Proc. of ISPAN 2005*.

[37] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979.

PLACE PHOTO HERE

Di Li received BE and MS degrees in control engineering from Zhejiang University and Shaanxi University of Science and Technology respectively in China. He worked with James Gross for his PhD in RWTH Aachen University since 2010.

PLACE PHOTO HERE

Erwin Fang Biography text here. Biography text here. Biography text here. Biography text here. Biography text here. Biography text here.

PLACE PHOTO HERE

James Gross Biography text here. Biography text here. Biography text here. Biography text here. Biography text here. Biography text here. Biography text here.