Received: 7 May 2017 | Revised: 31 December 2017 | Accepted: 2 January 2018
DOI: 10.1002/ett.3285

RESEARCH ARTICLE

WILEY

Robust clustering for ad hoc cognitive radio network

Di Li[1] | Erwin Fang[2] | James Gross[3]

[1]RWTH Aachen University, Aachen, Germany
[2]Swisscom (Schweiz) AG, Zurich, Switzerland
[3]KTH Royal Institute of Technology, Stockholm, Sweden

Correspondence
Di Li, RWTH Aachen University, 52074 Aachen, Germany.
Email: li@umic.rwth-aachen.de

Abstract

Cluster structure in cognitive radio (CR) networks facilitates cooperative spectrum sensing, routing, and other functionalities. Unlicensed channels, which are temporally available for a group of CR users in one area, consolidate the group into a cluster. More available unlicensed channels in a cluster make the cluster more likely to uphold against the licensed users' influence, making clusters more robust. This paper analyzes the problem of how to form robust clusters in a CR network such that CR systems benefit from collaboration within clusters despite intense primary user activity. 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 analyze the process of cluster formation, which not only contributes to the design of the distributed clustering scheme but also provides a guarantee on the convergence to a Nash equilibrium and the convergence speed. The proposed distributed clustering scheme outperforms state-of-the-art in terms of cluster robustness, convergence speed, and overhead. Extensive simulations are presented supporting the theoretical claims.

1 | INTRODUCTION

Cognitive radio (CR) is a promising approach to mitigate the increasing scarcity of radio spectrum[1] arising from the common practice to license radio frequencies in a de facto exclusive manner. In CR, licensed users can access the spectrum allocated to them at any point in time, whereas unlicensed users may access the spectrum when it is not utilized. This can be realized by the so-called opportunistic spectrum access, ie, unlicensed users access the spectrum only after validating that the channel is currently unoccupied. In the context of CR, licensed users are also called primary users (PUs), whereas unlicensed users are often referred to as secondary users and constitute a CR network (CRN).* For CRN, accurate spectrum sensing is critical, and the rate of false negatives, ie, the likelihood of misdetecting active PUs, needs to be minimized.[2] It has been shown that cooperative spectrum sensing, which relies on the consensus of CR users within a certain area, can significantly decrease the rate of false negatives despite the presence of receiver noise and wireless channel fading.[3,4] Thus, clustering of secondary nodes is regarded as a necessary condition to realize cooperative spectrum sensing[5] for opportunistic spectrum access.

Clustering is the process of logically grouping certain users in geographic proximity. As to wireless networking in general and in particular with respect to wireless ad hoc, mesh, or sensor networks, clustering is known to decrease the power consumption,[6] improve routing performance,[7] and improve the network lifetime and coverage.[8] For CRNs, apart from

*The terms user and node appear interchangeably in this paper. In particular, user is adopted when its networking or cognitive ability is discussed or stressed, whereas we refer to nodes typically in the context of the network topology.

improving the sensing accuracy, clustering also improves spectrum utilization among several CRNs by allowing for coordination particularly when CRNs have to vacate channels,[9] while it has also been known for reducing the interference between cognitive clusters[10] and for improving routing.[8]

In CRNs, formed clusters maintain a set of unlicensed channels that are validated by every CR node in that cluster, meaning that the channel is perceived as not being occupied by a PU. In the following, we refer to these maintained unlicensed channels as *common channels* (CCs). The availability of CCs within a cluster is elementary for the cluster, ie, if no CCs are available, then the corresponding cluster cannot operate any longer as CCs ascertain both control and payload data transmission within the cluster. However, due to PU activity, over time, the list of maintained CCs of a cluster varies randomly as it is generally unknown to secondary nodes when PUs appear on different licensed channels. Being able to maintain a sufficiently large list of CCs ensures the *robustness* of the cluster despite PU activity, ie, it provides a longer uninterrupted operation of the cluster.

On the other hand, the larger is the cluster size, the lower is, in general, the set of CCs that all nodes of a cluster observe as unoccupied by PUs. This is due to the fact that, in general, secondary nodes at different spatial locations will be able to sense the activity of different PUs due to different channel characteristics. Thus, a tradeoff arises for the formation of robust CR clusters: On the one hand, a low number of nodes in a cluster are desirable, as it generally provides more nodes with a common observation of PU activity on different channels and, thus, leads to a larger set of CCs, ultimately increasing the robustness. On the other hand, a too low number of nodes in a cluster compromises the sensing accuracy, particularly if only one or two nodes are members of a cluster.[11] Therefore, one needs to strike a balance between the *size* of a cluster and the *number* of CCs per cluster to balance robustness and sensing accuracy. Furthermore, cluster size plays a role in transmit power consumption, ie, the cluster size affects the transmit power consumption under certain routing schemes.[12,13]

In this paper, we analytically study the aforementioned tradeoff, which we term in the following the *CRN robust clustering problem*. We show it to be an NP-hard problem under certain assumptions and, furthermore, study centralized and distributed algorithms. We propose an alternate metric to measure cluster robustness in contrast to previous works.[14,15] We claim that cluster robustness cannot be indicated merely by the average number of CCs of a cluster but by the ability of the cluster to uphold over time despite random PU activity. Our proposed distributed scheme extends our previous work Robust Spectrum Sharing (ROSS)[14] by additionally incorporating control over the size of a cluster. Throughout this paper, we call these newly proposed distributed schemes *variants of ROSS*.

The rest of this paper is organized as follows. In Section 2, we review related work particularly with respect to clustering techniques in CRN. We also discuss in more detail the relation between the contribution in this paper and our previous work.[14] Our system model and the problem statement with respect to the robust clustering problem are presented in Section 3. The main contributions, ie, the centralized and distributed solutions, are introduced in Sections 4 and 5, respectively. Extensive performance evaluation is given in Section 6 before we conclude our work in Section 7.

2 | RELATED WORK

In the following, we first review the state-of-the-art regarding clustering in CRN in general and, then, focus on robust clustering in particular. With regard to forming clusters in CRN, deciding on the CC within each cluster is the foremost question to answer. Zhao et al,[16] Chen et al,[17] and Baddour et al[18] proposed different clustering schemes and enforced that every cluster possesses at least one CC. The clustering scheme in the work of Wu et al[11] looks for a network partition that improves the accuracy of spectrum sensing without considering robustness.

In the work of Wu et al,[19] clusters are formed by deciding on the cluster heads, where the transmit power for the long-haul transmission between the cluster heads is minimized. Zhang et al[12] proposed a cluster structure that improves energy efficiency. Furthermore, Asterjadhi et al[20] proposed a strategy on how to decide on the CCs and access multiple CCs within clusters. An event-driven clustering scheme was proposed for CR sensor networks in the work of Ozger and Akan.[21] However, none of the aforementioned schemes provide robustness of the clusters against random PU activity.

Mansoor et al[22] proposed a clustering algorithm that aims at speeding up the process of reclustering in case that PU activity eliminates all CCs. However, this work does not consider cluster robustness in the first place but rather focuses on reactive measures. In addition, Mansoor et al[22] presented a heuristic method to form clusters. Although the authors claim that robustness is one goal to achieve, only the minimization of the number of formed clusters is studied. A distributed clustering scheme referred to as Spectrum Opportunity-based Clustering (SOC) was proposed in the work of Liu et al,[15] targeting at cluster generation with multiple CCs per cluster. In the first phase of SOC, every secondary user forms clusters with some one-hop neighbor. In the second and final phase, each secondary user seeks to either merge other clusters or join one of them. The product of the number of CCs and cluster size is adopted as the metric by each secondary user in every phase. The authors compare SOC with other schemes in terms of the average number of CCs of the formed clusters, where SOC outperforms other schemes by 50% to 100%. Nevertheless, the drawbacks of this scheme are as follows. Although the adopted metric considers both the cluster size and the number of CCs, cluster formation can be easily dominated by only one factor. For example, a node that accesses abundant channels may form a cluster solely by itself, ie, a so-called singleton cluster. In addition, this scheme leads to a high variance of the cluster sizes, which is not desirable in certain applications as discussed in the works of Zhang et al[12] and Asterjadhi and et al.[20] In our previous work,[14] we proposed a distributed clustering scheme ROSS under a game theoretic framework. Compared with the clustering schemes introduced above, the clusters are formed faster and the clusters possess more CCs than in the case of being formed by SOC. However, as all the other clustering schemes, this scheme does not have control over formation of very small or very large clusters, being not desirable as discussed above. To summarize, our own previous work and SOC deem cluster robustness just to be the number of CCs per cluster. However, this potentially can lead to a significant number of singleton clusters being formed, which leads to lower sensing accuracy and has also other downsides as for example an increased routing overhead. In the following, we focus on striking the balance between cluster size and cluster robustness.

3 | SYSTEM MODEL AND PROBLEM FORMULATION

We consider a set of CR users $\mathcal{N}$ and a set of PUs distributed over a given area. A set of licensed channels $\mathcal{K}$ is available for the PUs. The CR users are allowed to transmit on channel $k \in \mathcal{K}$ only if no PU is detected to be occupying channel k. The CR users conduct spectrum sensing independently and sequentially on all licensed channels.[†] We adopt the unit disk model[23] for both primary and CR users' transmission. Thus, if a CR node i locates within the transmission range of an active PU p, i is not allowed to use the channel that is being used by p. We assume the PUs to change their operation channels slowly, and thus, we omit the time index when denoting 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 PUs is limited and CR users have different locations, different CR users have different views of the spectrum availability, ie, for any $i, j \in \mathcal{N}$, $K_i = K_j$ typically does not hold. The resulting network of CR nodes is represented 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 spatial distance between nodes i and j, and r is the radius of CR user's transmission range. Among the CR users, we denote by $\mathrm{Nb}(i)$ the neighborhood of i, which consists of the CR nodes located within i's transmission range.

We assume that there is one dedicated control channel that is used to exchange signaling messages during the clustering process. This control channel could be one of the industrial, scientific, and medical frequency channels or other reserved spectrum, which is exclusively used for transmitting control messages.[‡] Over the control channel, a secondary user i can exchange its spectrum sensing result K_i with all its one-hop neighbors $\mathrm{Nb}(i)$.

We next focus on a single CR cluster. A cluster C is a set of secondary nodes in an area, and there is a set of CCs that is available to each node belonging to the cluster. One of the nodes belonging to the cluster is, furthermore, the cluster head $h(C)$. The cluster head is able to communicate with any cluster member directly. The number of nodes belonging to 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 of all nodes in cluster C, ie, $K(C) = \cap_{i \in C} K_i$. Table 1 summarizes all parameters and their assumed relevance in our system model. The locations of the first appearances are provided for some notations.

3.1 | Robust clustering problem in CRN

Robustness of a cluster is its ability to uphold communication among the cluster members despite the influence of the active PUs. Thus, to achieve better robustness, a clear component of an optimization metric needs to be the amount of CCs among each formed cluster. However, this can lead in an extreme situation to a large amount of singleton clusters, if the size of the clusters is not controlled simultaneously. As discussed, a large amount of singleton clusters reduces spectrum

[†]We assume that every node can detect the presence of an active PU on each channel with certain accuracy. The spectrum availability can be validated with a certain probability of detection. While we do argue that too small cluster sizes lead, in general, to a loss of sensing accuracy, a study of the detailed spectrum sensing/validation accuracy is out of the scope of this paper.

[‡]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,[24,25] ie, 2 neighboring nodes establish communication on the same channel.

TABLE 1 Notations

Symbol	Description
$\mathcal{N}$	Set of CR users in a CRN
N	Number of CR users in a CRN, $N = \lvert\mathcal{N}\rvert$
$\mathcal{K}$	Set of licensed channels
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 desired cluster size
S_i	A set of claiming clusters, each of which includes debatable node i after phase I
$f(C)$	The number of CCs of a cluster C in the problem description (Section 3.1)
$\mathcal{S}$	The collection of all the possible clusters in $\mathcal{N}$ (Section 4)
M	The cardinality of $\mathcal{S}$ (Section 4)
C_i	The ith cluster in $\mathcal{S}$ (Section 4)
p	The weight with respect to cluster's size (Section 4)
d_i	Individual connectivity degree of CR node i
g_i	Neighborhood connectivity degree of CR node i
n	The number of debatable nodes
m	The number of claiming cluster heads
J	The new value of d_i for the CR node after it becomes a cluster member (Section 5)

Abbreviations: CCs, common channels; CR, cognitive radio; CRN, CR network.

sensing accuracy through cooperative sensing, as well as being not desirable from different other perspectives (routing, coordination with respect to channel vacation). Thus, we essentially propose to include this tradeoff in the optimization process of building clusters, captured in the following definition.

Definition 1. For a set of CR nodes $\mathcal{N}$, the **CRN robust clustering problem** is to determine a set of clusters $\mathcal{T}$, where the following conditions are satisfied.

1. The intersection of any 2 clusters in $\mathcal{T}$ results in the empty set.
2. The union of all clusters in $\mathcal{T}$ results in $\mathcal{N}$.
3. For all clusters with size within the range $[\delta_1, \delta_2]$, with $\delta_1, \delta_2 \in \mathbb{Z}^+$ and $\delta_1 \leq \delta_2$, the number of CCs per cluster is $f(C) = K(C)$, ie, it is given as the number of jointly available CCs for all nodes of that cluster. The desired size δ is within $[\delta_1, \delta_2]$, which is predecided based on the capability of the CR users and the tasks to be conveyed.
4. When the cluster size is larger or smaller than the range $[\delta_1, \delta_2]$, $f(C)$ is defined as 0, ie, singleton clusters may be formed but does not contribute to the objective function.
5. The sum over $f(C)$ for all clusters $C \in \mathcal{T}$ is maximal.

Note that, in the aforementioned definition, we distinguish between the real number of CCs per cluster, which is given as $K(C)$, and the contribution of each formed cluster toward the objective function, which is given by function $f(C)$. If the cluster size is within range, the 2 parameters are the same, otherwise the distinction enforces a cost for building clusters that are out of range.

The decision version of this problem is to determine whether there exists a set of clusters, say $\mathcal{T}$, so that $\cup_{C\in\mathcal{T}} C = \mathcal{N}$, and $\sum_{C\in\mathcal{T}} f(C) \geqslant \lambda$, where λ is a positive integer number. We have the following theorem on the problem's complexity.

Theorem 1. *The robust clustering problem in CRN is NP-hard, when $\delta_1 = 2$ and $\delta_2 > 3$.*

The proof is given in Appendix C.

4 | CENTRALIZED SOLUTION FOR ROBUST CLUSTERING

We now turn to algorithms that can solve the CRN robust clustering problem despite its complexity. We initially consider a centralized solution for this problem. Assuming some global knowledge of the CRN to be given at some point in the network, ie, the locations of PUs and their working channels and the locations of secondary users and available channels on them, we propose a centralized scheme. We obtain the set of $\mathcal{S}$, which contains all the clusters that satisfy the definition of cluster in Section 3. $\mathcal{S}$ is basically a powerset of $\mathcal{N}$, which nevertheless is restricted by connectivity and spectrum availability in the network. With $|\mathcal{S}| = M$, there is $\mathcal{S} = \{C_1, C_2, \dots, C_M\}$.[§] Then, the problem as defined in Definition 1 can be formulated as a binary linear programming problem

$$\begin{aligned} \max_{x_i} \quad & \Sigma_{i=1}^{M} \left(x_i \cdot |K(C_i)| - x_i \cdot p(||C_i| - \delta|)\right) \\ \text{subject to} \quad & \Sigma_{i=1}^{M} (x_i \cdot e_{ij}) = 1 \quad \text{for } \forall j \in \{1, \dots, N\}, \end{aligned} \tag{1}$$

where $x_i, i \in \{1, 2, \dots, M\}$ is a set of binary optimization variables. Being either 1 or 0, x_i denotes whether the ith cluster C_i in $\mathcal{S}$ is chosen or not. $e_{ij}, i \in \{1, 2, \dots, M\}$ and $j = \{1, \dots, N\}$ are sets of constants, which indicate whether the CR node j resides in the cluster C_i, ie, $e_{ij} = 1$ means node j resides in the cluster C_i and $e_{ij} = 0$ indicates j does not belong to that cluster.

The constraint regulates that for any node j, the sum of $x_i \cdot e_{ij}$ over all the clusters in $\mathcal{S}$ is 1. This constraint has 2 implications. First, the sum being larger than zero indicates that every node should be involved. Second, the sum that is equal to 1 means a node can only appear in one cluster, which prevents the chosen clusters from overlapping.

As to the objective function, the sum of the first term in the bracket over all clusters in $\mathcal{S}$ is the sum of CCs of the clusters that constitute the CRN. For the second term in the bracket, p is a positive increasing function with respect to the difference between C_i's size and the desired size δ. When x_i is 1 (C_i is chosen) but $|C_i|$ does not equal to desired cluster size δ, the second item is negative, which contradicts the direction of the optimization. Thus, the second item discourages the appearance of clusters whose sizes deviate from the desired cluster size δ.

Note that, here, we adopt the desired cluster size instead of the upper and lower bounds for cluster sizes, which is described in Definition 1.

With δ and the deviation from it, we actually have a large range of cluster sizes, which guarantees the feasibility of the problem as we allow the singleton cluster to exist, meanwhile we have a better control of the resulting sizes by setting weights for the formed clusters based on their deviations from the desired size.

The difficulty of implementing this method is obtaining the set $\mathcal{S}$. In the worst case, ie, if every CR node can communicate directly with any other node, then the CRN forms a fully connected graph, and therefore, the size of the powerset $\mathcal{S}$ is $\Sigma_{r=1}^{N} \binom{N}{r} = 2^N - 1$.

5 | DISTRIBUTED CLUSTERING ALGORITHM: VARIANTS OF ROSS

In this section, we introduce our distributed clustering schemes. With the variants of ROSS, CR nodes form clusters based on their own available channels, as well as the available channels of the nodes in their neighborhood. This is clarified and conducted through a series of interactions on the control channel. All variants of ROSS consist of 2 cascaded phases: *cluster formation* and *membership clarification*, as shown in Figure 1. In the first phase, clusters are initially formed such that every CR user becomes either cluster head or cluster member. During this phase, size control is already realized; however, memberships might not be efficient with respect to robustness while also not being necessarily unique. This is addressed in the second phase, where nonoverlapping clusters are formed in a way that the CCs of the involved clusters are predominantly increased.

5.1 | Phase I: cluster formation

Before conducting clustering, we assume that spectrum sensing and neighborhood discovery have been completed. Furthermore, neighboring nodes have exchanged already their channel availabilities via the dedicated control channel. As a

[§] The subscript i means the ith cluster in $\mathcal{S}$.

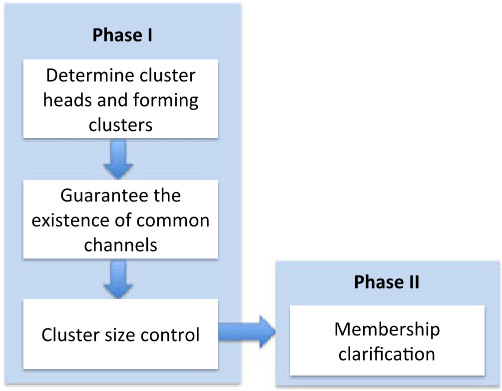

FIGURE 1 Processing steps of Robust Spectrum Sharing

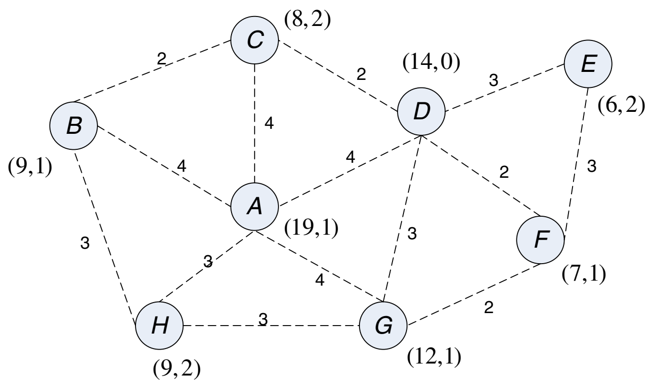

FIGURE 2 Illustration of the resulting connectivity vector (d_i, g_i) for each node of an example cognitive radio network

result, every CR node is aware of the available channels of themselves and their one-hop neighbors. Next, cluster heads are determined after a comparison series among neighbors. Two metrics are proposed to characterize the channel availability in the proximity of each terminal, which subsequently are used to decide 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 *pairwise* CCs between node i and each of its neighbors. Be aware that it does not reflect the amount of *jointly* CCs among all neighbors of i.
- *Neighborhood connectivity degree* g_i: In contrast, g_i is the number of CCs that is *jointly* available for i and all of its neighbors, thus $g_i = |\cap_{j \in \mathrm{Nb}(i) \cup i} K_j|$. It therefore 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* (d_i, g_i). The connectivity vector is determined by every secondary user and is then broadcasted. Figure 2 illustrates the computation of the connectivity vectors for a CRN, where a dashed edge indicates that one end node is within the other's transmission range, whereas the number along the dashed line is the number of CCs between the 2 end nodes. In this example, the sets of available channels on the nodes are $K_A = \{1, 2, 3, 4, 5, 6, 10\}, K_B = \{1, 2, 3, 5, 7\}, K_C = \{1, 3, 4, 10\}, K_D = \{1, 2, 3, 5\}, K_E = \{2, 3, 5, 7\}, K_F = \{2, 4, 5, 6, 7\}, K_G = \{1, 2, 3, 4, 8\}$, and $K_H = \{1, 2, 5, 8\}$. Figure 2 shows, in particular, the resulting connectivity vector per node.

5.1.1 | Determining cluster heads and forming clusters

Given the connectivity vector per node, the procedure of determining cluster heads is as follows. Each CR node decides whether it is a cluster head by comparing its connectivity vector with all neighboring connectivity vectors. When CR node i has lower individual connectivity degree than any of its neighbors except for those that have already been identified as cluster heads, node i becomes a cluster head. If there is a CR node j in i's neighborhood, which has the same individual connectivity degree as i, ie, $d_j = d_i$ while the connectivity degree of j is lower than for all other nodes in its neighborhood (except for nodes that already declared themselves as heads), then out of i and j, the node with higher neighborhood connectivity degree will become cluster head. If $g_i = g_j$ as well, the node ID is used to break the tie, ie, the one with smaller node ID becomes the cluster head. The node that is identified as cluster head broadcasts a message to notify its neighbors of this status update. As a consequence, all neighbors, which have not become cluster head themselves, become cluster members of this cluster head. In this step, nodes can become member of multiple clusters, depending on how many neighbors declare themselves as cluster heads.¶ During the whole phase I, whenever a CR node becomes cluster head or the cluster composition changes, the cluster head broadcasts new/updated information about the cluster structure, particularly the new/updated sets of available channels regarding itself and all its cluster members. Pseudocode regarding this process, ie the cluster head decision and the initial cluster formation, is in Algorithm 1 in Appendix A.

After a CR node, say i, receives notification that there is a new cluster head in its neighborhood, i sets its individual connectivity degree to a positive number $J > |\mathcal{K}| \cdot N$ and broadcasts the new individual connectivity degree. When node i is associated with multiple clusters, ie, i has received multiple notifications from different cluster heads, d_i is still set to be J. The manipulation of the individual connectivity degree of the cluster members accelerates the decision on the cluster heads.

5.1.2 | The existence of CCs

After executing Algorithm 1, several formed clusters may not possess any CCs. As decreasing the cluster size usually increases CCs within a cluster, the next step is to decrease the cluster size accordingly. This is done by the following sequence of removing nodes according to an ascending list of nodes regarding their number of CCs between them and the cluster head. In other words, the cluster member that has the least CCs with the cluster head will be removed first. When there are multiple nodes having the same amount of CCs with the cluster head, the node whose elimination results in more CCs will be removed. In case of a tie, it can be broken by removing the node with smaller node ID. It is possible that cluster heads remove all their neighbors to obtain CCs, which results in a singleton cluster. The pseudocode for this procedure is given as in Algorithm 2. As for the nodes that are removed from a cluster, they restore their original individual connectivity degrees, then execute Algorithm 1 and become either cluster heads or get included into other clusters (see also Theorem 2).

5.1.3 | Cluster size control in dense CRN

Both analysis and simulation[26] show that with ROSS, when network density increases to a certain level, the number of formed clusters becomes constant. This means if the network density keeps on increasing, the cluster size increases linearly with the network density. Thus, it is necessary to control the cluster size when CRN becomes denser, and this task falls upon the cluster heads.

To control the cluster size, cluster heads remove their cluster members when cluster sizes are larger than a threshold. The threshold should be larger than the desired size δ because there are overlaps between neighboring clusters. We set the threshold as $t \cdot \delta$, where the constant parameter t is dependent on network density and CR nodes' transmission range. We adopt t to be between 1 and the ratio of the average neighborhood size and the desired size. When t is smaller, eg, $t = 1$, the formed cluster in phase I is δ. For a cluster which has members included by other clusters, the size of that cluster will be smaller than δ after the membership clarification phase. If t is chosen large, eg, $t \cdot \delta$ is equal to the size of the neighborhood, cluster size control will not work any more.

The cluster head removes the cluster members sequentially according to the above explained principle. The removed nodes restore their original individual connectivity degrees. This process ends when each cluster's size is smaller or equal to $t \cdot \delta$. As this procedure is similar with that in Section 5.1.2, Algorithm 2 can also be applied.

We have the following lemmas to show every secondary user will eventually be either integrated into a cluster or become a cluster head.

Lemma 1. *Given a CRN where any secondary user is able to communicate with any other secondary user through the other nodes, then after the phase of cluster head selection and initial cluster formation, every secondary user either becomes cluster head or gets included into at least one cluster.*

The Proof is given in Appendix B.

¶The issues arising out of cluster heads in the neighborhood of a newly formed cluster head are addressed in Sections 5.1.2 and 5.1.3

Lemma 2. *When a secondary user becomes cluster head, it will not become cluster member again.*

Proof. A secondary node, say i, becomes cluster head when its *individual connectivity degree* is smaller than any of its neighbors. Afterwards, the *individual connectivity degrees* of its neighbors becomes J. If certain nodes are removed from the cluster due to guaranteeing CC or size control, these nodes may become either cluster members of another cluster head or cluster heads themselves. In both cases, i's *individual connectivity degree* is still smaller than the one of the respective other nodes. Note that, when the removed node becomes cluster head, it will not include its former cluster head i, so that i does not become cluster member and so its *individual connectivity degree* does not change. □

Lemma 3. *In the process of cluster head selection and initial cluster formation, the maximum number of times that a secondary node becomes cluster head is N.*

This lemma follows from Lemma 2 considering that N is the number of all the secondary users in the CRN. Based on the aforementioned lemmas, we have the following.

Theorem 2. *Assuming the time for a secondary user to update the information about cluster heads in its neighborhood is T, then it takes at most $N * T$ to finish the process of cluster head selection and initial cluster formation.*

Phase I ends when no more secondary users become cluster heads. Based on Lemmas 1 and 3, Theorem 2 follows directly. Note that as Algorithm 1 is executed concurrently by different secondary users, the required time is typically considerably lower.

If we apply Algorithm 1 to the example CRN in Figure 2, the outcome results to the representation in Figure 3. Nodes B and H have the same individual connectivity degree, ie, $d_B = d_H$. As $g_H = 2 > g_B = 1$, node H becomes the cluster head and cluster $C(H)$ is $\{H, B, A, G\}$.

5.2 | Phase II: membership clarification

After running phase I of ROSS, we notice that nodes A, B, and D are included in more than one cluster, as shown in Figure 3. We refer to these nodes as *debatable nodes* as their cluster affiliations are not uniquely decided. All clusters that include debatable node i are called *claiming clusters* of node i, and the set of these clusters is denoted as S_i. Nevertheless, debatable nodes need to be exclusively associated with only one cluster and be removed from the other claiming clusters. We refer to this procedure as *cluster membership clarification*.

5.2.1 | Distributed greedy algorithm (DGA)

When a debatable node i decides to join the cluster $C \in S_i$, the guiding idea is that its decision should result in the greatest increase of CCs in all its claiming clusters. As the node i has been notified of the spectrum availability on all the nodes in each claiming cluster, node i can calculate how many more CCs will be generated in S_i if it chooses a claiming cluster and leaves the other claiming clusters. In case of a tie between 2 claiming clusters, i chooses to stay in the cluster whose cluster head shares the most CCs with i. When a tie still exists, node i chooses to stay in the claiming cluster that has the smallest size. Node IDs of cluster heads will be used to break tie in the end if necessary. The pseudocode of this algorithm is given by Algorithm 3. After deciding its membership, debatable node i notifies all its claiming clusters.

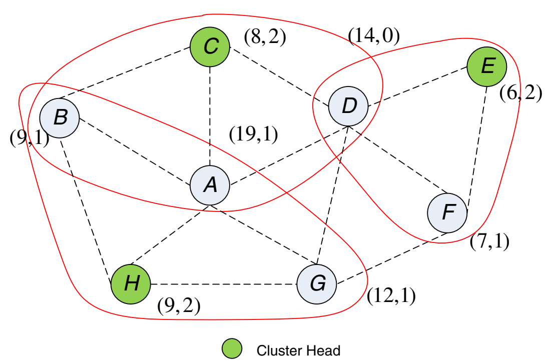

FIGURE 3 Cluster formation after phase I of Robust Spectrum Sharing. Nodes A, B, and D are debatable nodes as they belong to multiple clusters

The autonomous decisions made by the debatable CR nodes raises the possibility of an endless chain effect during the membership clarification phase. A debatable node's choice is dependent on the composition of its claiming clusters, and the members of these claiming clusters can be changed by other debatable nodes' moves. There is the possibility that this process may go on forever. However, by formulating the process of membership clarification into a game, we can show that equilibrium is reached after a finite number of best response updates made by the debatable nodes. Thus, the membership clarification phase is guaranteed to terminate.

5.2.2 | Formulation of ROSS-DGA to congestion game

Game theory is a powerful mathematical tool for studying, modeling, and analyzing the interactions among individuals. A game consists of 3 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 modeled as rational and intelligent decision makers, which are related through one explicit formalized incentive expression (the utility or cost). Game theory provides standard procedures to study potential equilibria.[27] Over the last decade, game theory has been extensively applied to problems in communication and networking.[28,29] Congestion game is an interesting game model, which describes the problem where participants compete for limited resources in a noncooperative manner. It has the good property that a Nash equilibrium can be achieved after finite steps of best response dynamic, ie, 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 server selection in distributed computing platforms[30] or users downloading files from cloud, etc.

To formulate the debatable nodes' membership clarification into a congestion game, we see this process from a different perspective. In particular, for a debatable node, instead being in all its claiming clusters, now it is not included in any claiming cluster and it needs to decide on one cluster to join. When a debatable node i joins one cluster C, the decrease of CCs in cluster C is $\sum_{C\in S_i}\Delta|K(C)| = \sum_{C\in S_i}(|K(C)| - |K(C\cup i)|)$. Then, node i chooses the cluster C, where the decrease of CCs in cluster C is smaller than the decrease if i would have joined any other claiming cluster in S_i. The relation between the debatable nodes and the claiming clusters is shown in Figure 4.

In the following, we show that the decision of debatable nodes to clarify their membership can be mapped to the behavior 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 $\Gamma = (\mathcal{P}, \mathcal{R}, \sum_{i,i\in\mathcal{P}}, \rho)$ with the following elements.

- $\mathcal{P}$ is the set of players in the game, which are the debatable nodes in our problem.
- $\mathcal{R} = \cup S_i, i \in \mathcal{P}$, is set of the resources for players to choose. In our problem, S_i is the set of the claiming clusters of 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 the claiming clusters S_i. As debatable node i is supposed to choose only one claiming cluster, only a single resource will be allocated to i.
- The cost function $\rho(C)$ regarding resource C. $\rho(C) = \Delta|K^i(C)|, C \in S_i$, 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 accepting 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 nondecreasing with respect to the number of nodes joining cluster C.

When the utility function is decided purely by the amount of players accessing the resource, the game is a canonical congestion game.[31] In our game, as the channel availability on debatable nodes (players) is different, the loss of CCs (cost) caused by a debatable node could also be different. Hence, this congestion game is player specific.[31] In this game, every player greedily updates its strategy (choosing one claiming cluster to join) if joining a different claiming cluster minimizes

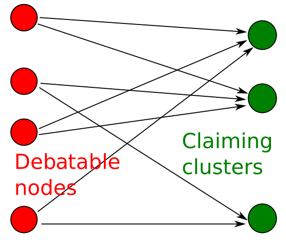

FIGURE 4 Illustration of debatable nodes and claiming clusters

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 behavior of a debatable node in the membership clarification phase.

As to singleton congestion game, there exists a pure equilibrium that can be reached with the best response update, while 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.[31] 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 $\mathcal{O}(N^3)$. In fact, the upper bound for the number of steps that are involved in this process is much smaller than N^3. The percentage of debatable nodes in the network is shown in Figure 11, which is between 10% and 60%. On the other hand, the number of cluster heads is dependent on the network density and the CR node's transmission range, as mentioned in Section 5.1. The simulation in our other work[32] shows that the cluster heads account from 3.4% to 20% of the total CR nodes with increasing network density. Furthermore, as the game is played locally and in parallel, ie, a debatable node can only interact with a few claiming clusters, the execution speed is significantly reduced.

5.2.3 | Distributed fast algorithm (DFA)

Based on 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, it is more suitable for CRN where the channel availability changes frequently. To run ROSS-DFA, debatable nodes execute only one loop of Algorithm 3.

Now, we apply both ROSS-DGA and ROSS-DFA to the network in Figure 3 after phase I of ROSS is complete. In the network, node A's claiming clusters are cluster $C(C), C(H) \in S_A$, while the respective members are $\{A, B, C, D\}$ and $\{A, B, H, G\}$. The 2 possible strategies of node A are illustrated in Figure 5. In Figure 5A, node A stays in $C(C)$ and leaves $C(H)$, which brings 2 more CCs to S_A, which is more than that brought by another strategy, as shown in Figure 5B. After similar decisions are made by the other debatable nodes B and D, the final clusters are formed as shown in Figure 6.

6 | PERFORMANCE EVALUATION

Taking the final clustering result of ROSS into account for our toy example shown in Figure 6, we can compare the outcome with our centralized scheme proposed in Equation 1 as well as the state-of-the-art algorithm SOC.[15] Those corresponding results of the latter 2 schemes are shown in Figure 7. We observe for this example case that ROSS and the centralized scheme achieve cluster sizes that are more balanced, whereas SOC leads to a larger variance in terms of the cluster size. Regarding the amount of CCs, the same observation holds.

In the following, we are interested in a more general performance comparison regarding clustering in CRNs. Therefore, we present an extensive evaluation study. We base our evaluations on simulations and consider the following comparison schemes.

- ROSS without size control: ROSS-DGA, ROSS-DFA;
- ROSS with size control: ROSS-δ-DGA and ROSS-δ-DFA, where δ is the desired cluster size;

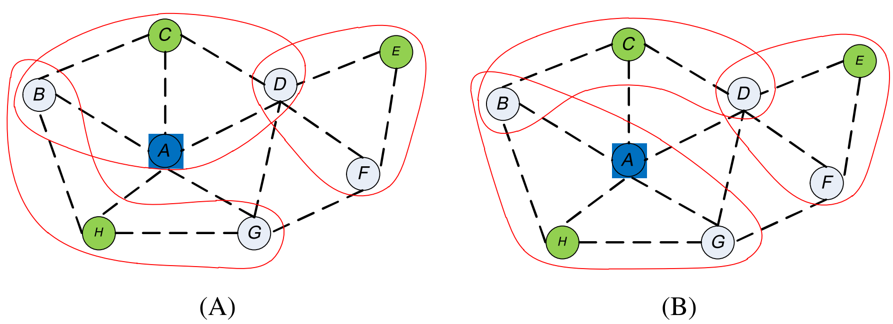

FIGURE 5 Membership clarification: possible cluster formations caused by node A's different choices. A, Node A stays in cluster $C(C)$ and quits $C(H)$, $\Delta|K(C(C))| + \Delta|K(C(H))| = 2$; B, Node A stays in cluster $C(H)$ and quits $C(C)$, $\Delta|K(C(C))| + \Delta|K(C(H))| = 1$

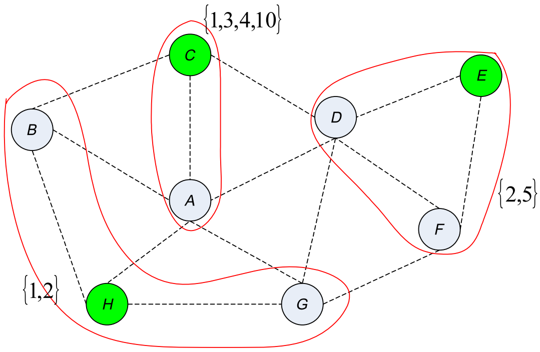

FIGURE 6 Final formation of clusters. Common channels are shown as well as the corresponding clusters

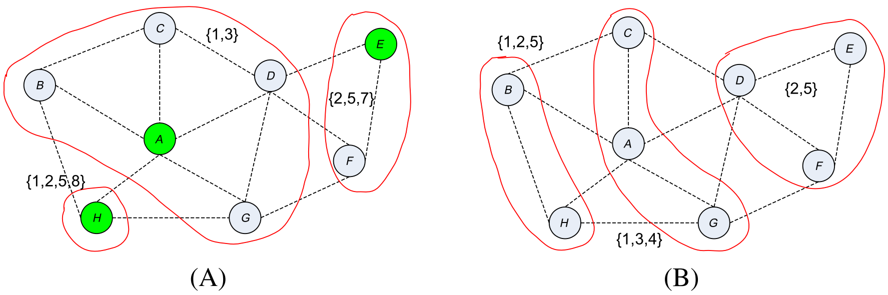

FIGURE 7 Final clusters formed by system on a chip (SOC) and the centralized clustering scheme. A, Generated by SOC; B, Generated by the centralized clustering scheme

- SOC[15]: a distributed clustering scheme pursuing cluster robustness;
- Centralized robust clustering scheme: in our evaluations, we use the built-in function *bintprog* of MATLAB to solve the corresponding integer optimization problem given in Equation 1.

Given these comparison schemes, we are interested in the following performance metrics regarding clustering.

- **The average number of CCs per nonsingleton cluster.** Previous works[14,15] claimed that the larger average number of CCs over all the clusters indicates robustness. As mentioned, this interpretation has several shortcomings. First,

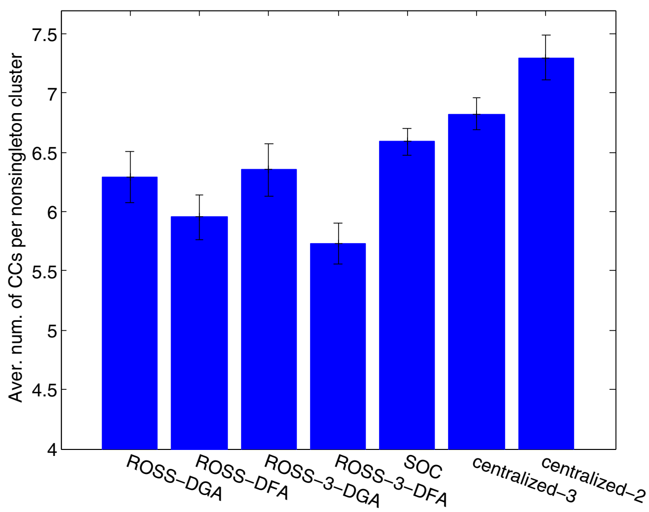

FIGURE 8 Average number of common channels (CCs) of nonsingleton clusters. DFA, distributed fast algorithm; DGA, distributed greedy algorithm; ROSS, Robust Spectrum Sharing; SOC, system on a chip

singleton clusters should not be considered when calculating the average number of CCs, as singleton clusters do not contribute to the collaborative computing or sensing. Second, the average number of CCs does not necessarily indicate the robustness of a cluster because the ability of a cluster to sustain PU activity also depends on the size and the location of the cluster members. This information, however, is not reflected by the average number of CCs. Thus, in the following, we will consider the average number of CCs per cluster, excluding singleton clusters from this averaging, as our first performance metric.

- **Robustness of the clusters against newly added PUs.** If clusters are less robust, this leads to an increasing number of unclustered CR nodes if clusters are exposed to random PU activity. Thus, we are interested in this effect as a second measure for robustness. In particular, we are interested in the number of CR nodes that is still part of a cluster after exposing the clusters to PU activity.
- **Cluster sizes.** We investigate the distribution of the sizes of the formed clusters. This metric reflects the aforementioned size constraints, ie, clusters are supposed to be neither too big nor too small.
- **Control message overhead.** We investigate the number of control messages involved until the final clustering result is established.
- **Influence from inaccurate spectrum sensing.** While most of our evaluations are conducted under the assumption of perfect channel sensing by the individual CR nodes, an important question relates to the fact how the clustering

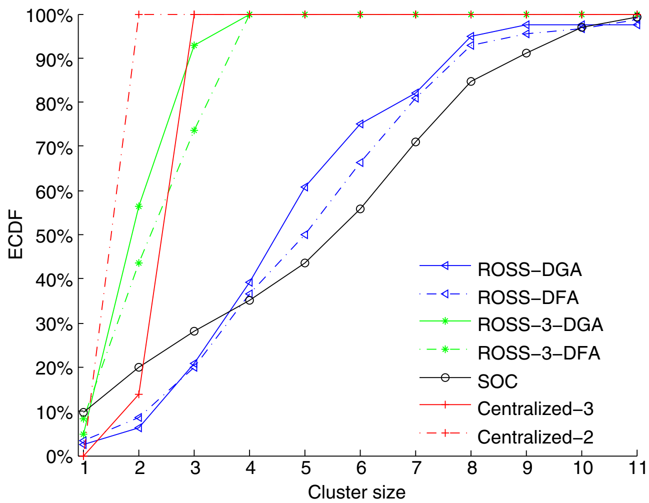

FIGURE 9 Empirical cumulative distribution function (ECDF) of cognitive radios residing in clusters with different sizes. DFA, distributed fast algorithm; DGA, distributed greedy algorithm; ROSS, Robust Spectrum Sharing; SOC, system on a chip

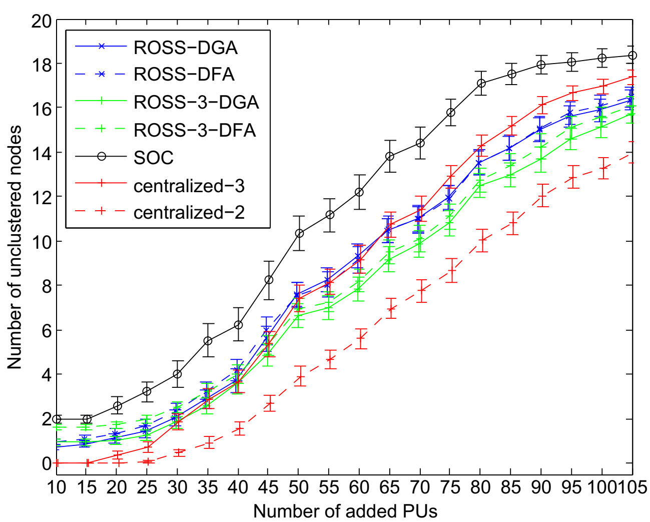

FIGURE 10 Number of unclustered cognitive radios with decreasing spectrum availability. DFA, distributed fast algorithm; DGA, distributed greedy algorithm; PUs, primary users; ROSS, Robust Spectrum Sharing; SOC, system on a chip

performs under imperfect sensing accuracy. In case of erroneous channel sensing, false negatives harm PUs, whereas false positives harm the CR nodes. Both effects obviously impact the clustering process. However, in the following, we only consider the impact from false negatives. In particular, we assume that PU activity is only correctly detected by a CR node in transmission range with a certain probability, ie, there is a certain probability for misdetections. Given this erroneous sensing result, the secondary users nevertheless make their clustering decisions. As we are interested in the distorting impact of the erroneous channel sensing results on the clustering process, after the clustering is complete, we provide ground truth and reevaluate the discrepancy between the assumed channel utilization and its effect on clustering, and the de facto channel utilization and how this affects the CCs of formed clusters.

Our performance evaluation is split into 2 parts. First, we investigate the performance of the centralized scheme and the distributed schemes for a small network, as the runtime for the centralized solution quickly grows out of hand as a function of the network size. In the second part, we investigate the performance only of the distributed schemes for larger settings. The following simulation settings are identical for both evaluations: CRs and PUs are deployed on a 2-dimensional Euclidean plane. The number of licensed channels is 10, each PU is operating on each channel with probability of 50%. The constant t that is used to control cluster size for ROSS (discussed in Section 5.1.3) is 1.3. The CR users are assumed to sense the existence of PUs and identify available channels perfectly, unless we investigate the impact from erroneous channel sensing. All primary and CR users are assumed to be static during the process of clustering. All other parameters ie, the number of CR and PU, and their transmission ranges are given at the beginning of the respective subsections. The simulation is written in C++, and the performance results are averaged over 50 randomly generated topologies. We provide confidence intervals corresponding to confidence level of 95%.

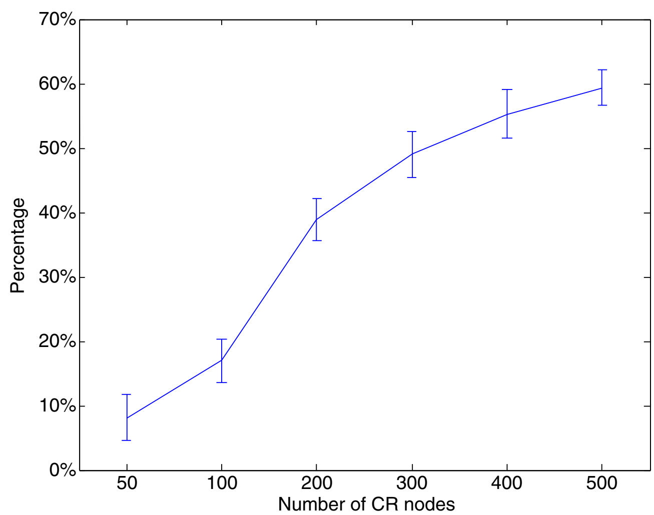

FIGURE 11 Percentage of debatable nodes after Robust Spectrum Sharing phase I. CR, cognitive radio

TABLE 2 Quantity of control messages

Scheme	Message Overhead	Number of Messages	Content and Size of the Message
ROSS-DGA, ROSS-δ-DGA	$\mathcal{O}(N^3)$[a]	$N + n^2m$ (upper bound)	Phase I: ID, d_i, g_i, which are 3 bytes
ROSS-DFA, ROSS-δ-DFA	$\mathcal{O}(N)$[b]	$N + n$ (upper bound)	Phase II: Cluster head i broadcasts channel availability to all members, which are $\vert C(i)\vert\vert\mathcal{K}\vert$ bytes
SOC	$\mathcal{O}(N)$	$3N$	Every cognitive radio node i broadcasts channel availability on all cluster members, which is $\vert C(i)\vert\vert\mathcal{K}\vert$ bytes
Centralized	$\mathcal{O}(N)$	$N + n + m$ (upper bound)	clustering result, which is $2N$ bytes[c]

[a]For the upper bound on the number of messages.

[b]For the upper bound on the number of messages.

[c]Assuming the data structure of the clustering result is in the form of $\{i, C\}, i \in C, i \in \mathcal{N}$.

Abbreviations: DFA, distributed fast algorithm; DGA, distributed greedy algorithm; ROSS, Robust Spectrum Sharing; SOC, system on a chip.

6.1 | Centralized scheme vs decentralized schemes

We start with the comparison of the centralized scheme versus various distributed ones. For this, we consider that 10 PUs and 20 CR users are dropped randomly (with uniform distribution) in a square area where side length is A. Transmission ranges of both primary and CR users are set to $A/3$. By doing this, we abstract from the influence of any given physical layer technology and propagation environment parameters. Due to the parameterization, on average, 7 channels are available per CR node when the clustering process is started. We set the desired cluster size δ as 3. As for the centralized schemes, we set the following parameters numerically: $p(1) = 0.4$, $p(2) = 0.6$.

We start with the consideration of the CCs in all nonsingleton clusters. Figure 8 shows that basically the centralized schemes outperform all distributed schemes in terms of the average number of CCs per cluster. SOC achieves the most CCs among the distributed schemes because SOC groups the neighboring CRs that share the most abundant spectrum together, without considering the size of them. However, as a consequence, SOC also generates the most singleton clusters. As to the variants of ROSS, we notice that the greedy mechanism (ie the ROSS-DGA variants) maximize the CCs in nonsingleton clusters significantly.

Figure 9 provides further insights into the performance comparison. Here, we depict the empirical cumulative distribution function (ECDF) of the size of the clusters.

The centralized schemes do not result in any singleton clusters in the considered evaluation scenarios. In contrast, ROSS-DGA/DFA account for 3% singleton clusters of the total CR nodes, as compared to 10% of nodes being unclustered when applying SOC. ROSS-DGA and ROSS-DFA with size control feature generate 5% to 8% unclustered CR nodes, which is due to the cluster pruning procedure (discussed in Sections 5.1.2 and 5.1.3). In terms of cluster size, the clusters resulting from the centralized schemes and ROSS with cluster size control mechanism have little deviation from the desired cluster size. In contrast, the size of clusters resulting from ROSS-DGA and ROSS-DFA have a higher variance but appear to be better than SOC, ie, the 50th percentiles for ROSS-DGA, ROSS-DFA, and SOC are 4.5, 5, and 5.5 and the 90th percentiles for the 3 schemes are 8, 8, and 9. Thus, the corresponding sizes resulting from ROSS are closer to the desired size.

Next, we consider the robustness of clusters if facing random PU activity. Thus, we extend the simulation by adding more PUs sequentially into the area of the CRN, leading to a decreasing spectrum availability. While 10 PUs are in the network at start, some extra 19 batches of PUs are added sequentially, each batch including 5 PUs that are placed randomly in the area. These added PUs choose then an active channel also at random. Figure 10 shows the corresponding average number

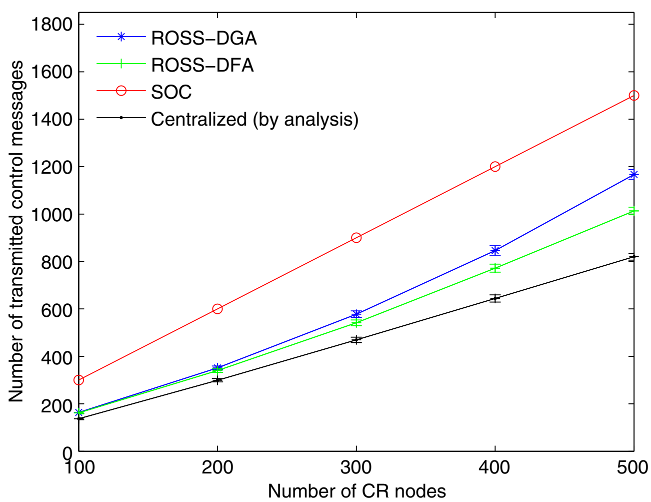

FIGURE 12 The number of transmitted control messages required for clustering (based on the third column in Table 2). CR, cognitive radio; DFA, distributed fast algorithm; DGA, distributed greedy algorithm; ROSS, Robust Spectrum Sharing; SOC, system on a chip

TABLE 3 The average numbers of neighbors and the chosen desired sizes with respect to different network scales

Number of Cognitive Radios	100	200	300
Average number of neighbors	9.5	20	31
Desired size δ	6	12	20

of unclustered CR nodes as a result of this significant increase in PU activity. The figure reveals that the centralized scheme with a desired size of 2 leads to the best robustness, whereas SOC leads to the worst one. Surprisingly, the centralized scheme with desired size of 3 does not outperform the variants of ROSS because pursuing larger cluster sizes generally leads to clusters with a lower amount of CCs. In contrary, the variants of ROSS generate some smaller clusters that are more likely to be maintained despite the increasing PU activity.

Alternatively, we can consider the total share of users (still) residing in a cluster after the addition of the PUs as performance metric for robustness. If we do so, the ROSS-based schemes maintain 5%, 30%, and 230% more secondary users within clusters than SOC, when the numbers of newly added PR are 10, 40, and 80, respectively (no figure is provided for this data). This observation illustrates clearly that the average number of CCs of nonsingleton clusters does not necessarily reflect the robustness of clusters, ie, SOC obtains the most CCs among the distributed schemes, but the resulting clusters are vulnerable to PU activity.

We finally turn to a comparison of the amount of the involved control messages for the different clustering schemes. For this, we count the number of *transmissions of control messages* as metric,[33] without distinguishing broadcast or unicast control messages.

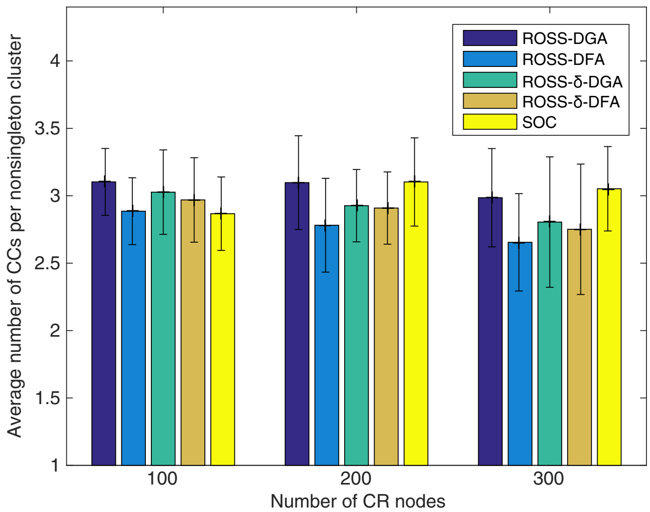

FIGURE 13 Average number of common channels (CCs) of nonsingleton clusters in case of increasing the number of cognitive radio (CR) nodes. DFA, distributed fast algorithm; DGA, distributed greedy algorithm; ROSS, Robust Spectrum Sharing; SOC, system on a chip

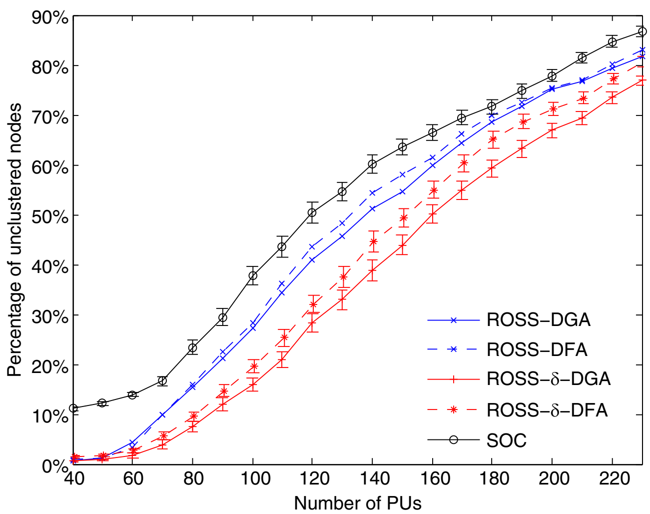

FIGURE 14 Percentage of unclustered cognitive radio nodes with increasing number of primary users (PUs), where $N = 100$. DFA, distributed fast algorithm; DGA, distributed greedy algorithm; ROSS, Robust Spectrum Sharing; SOC, system on a chip

As to ROSS, in the first phase, the maximal number of broadcasts is N according to 2. In the second phase, the upper bound for the number of message exchanges is n^2m and n for ROSS-DGA and ROSS-DFA, respectively, where n is the number of debatable nodes and m is the number of claiming clusters. SOC consists of 3 rounds, and in each round every node needs to perform a broadcast to do comparisons and cluster merging. The centralized scheme is conducted at some control device, which involves information aggregation and subsequent dissemination of clustering decisions. To analyze the centralized scheme's message overhead, we adopt a backbone structure proposed in the work of Onus et al[34] and apply ROSS to generate cluster heads that serve as the backbone. In the stage of information aggregation, all the nodes transmit information to the cluster heads that forward the messages to the controller. In the dissemination stage, all the cluster heads and the debatable nodes broadcast the clustering result, thus the upper limit for the number of broadcast is $N + m + n$.

The number of control messages that are involved in ROSS variants and the centralized scheme is related to the number of debatable nodes. Figure 11 shows the percentage of debatable nodes with different network densities. Table 2 shows the amount of control messages using big O notation, the number (or upper bound) of control messages (illustrated in Figure 12), and the size of control messages for the different schemes under consideration. From Figure 12, we can see that the upper bound on the number of control messages that are involved in the variants of ROSS is still smaller than the

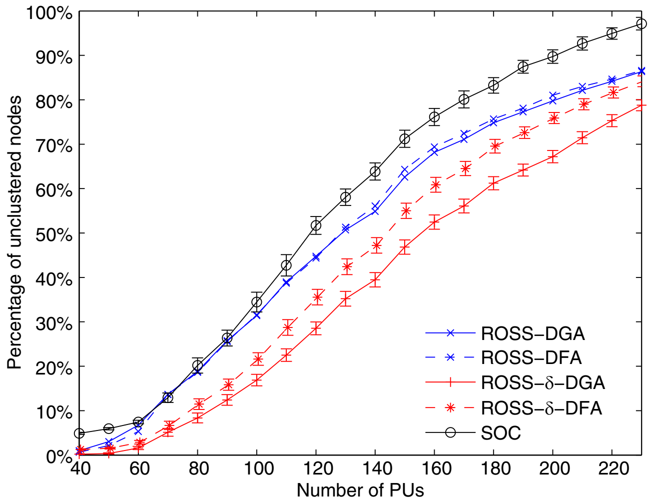

FIGURE 15 Percentage of unclustered cognitive radio nodes with increasing number of primary users (PUs), where $N = 200$. DFA, distributed fast algorithm; DGA, distributed greedy algorithm; ROSS, Robust Spectrum Sharing; SOC, system on a chip

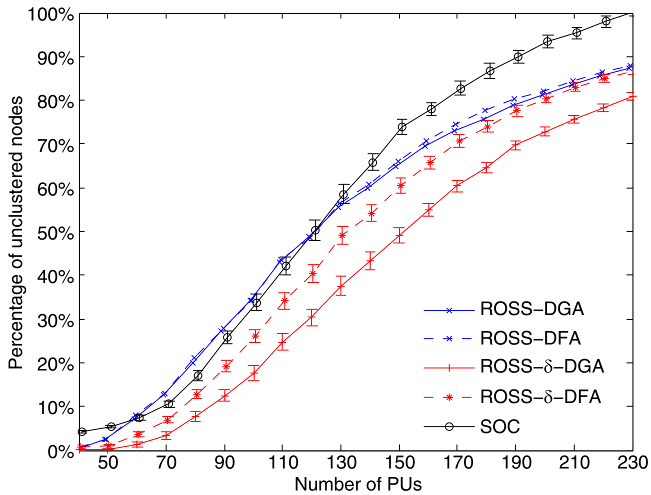

FIGURE 16 Percentage of unclustered cognitive radio nodes with increasing number of primary users (PUs), where $N = 300$. DFA, distributed fast algorithm; DGA, distributed greedy algorithm; ROSS, Robust Spectrum Sharing; SOC, system on a chip

one involved in SOC. Meanwhile, the length of the control messages involved in the variants of ROSS is shorter than that involved in the centralized scheme.

6.2 | Comparison among the distributed schemes

We now switch to a more fine-grained investigation only of the distributed schemes. Here, we are most interested in their properties when the network size and density scales. In particular, we set the desired size based on the density of the network. As shown in Table 3, the desired size is equal to 60% of the average number of neighbors. The transmission range of CR is now set to $A/5$, whereas the PU transmission range is set to $2A/5$. The initial number of PUs is set to 30.

We start again with considering the average number of CCs over all nonsingleton clusters, as shown in Figure 13. Note that, in this case, we increase the number of CR nodes in the scenario. As in the previous section, the result does not reveal a significant performance advantage of either of the distributed schemes.

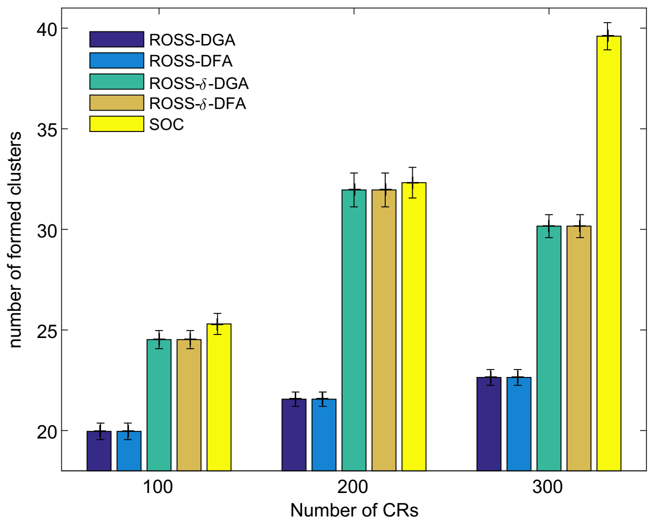

FIGURE 17 The number of formed clusters in cognitive radio (CR) network with different densities. DFA, distributed fast algorithm; DGA, distributed greedy algorithm; ROSS, Robust Spectrum Sharing; SOC, system on a chip

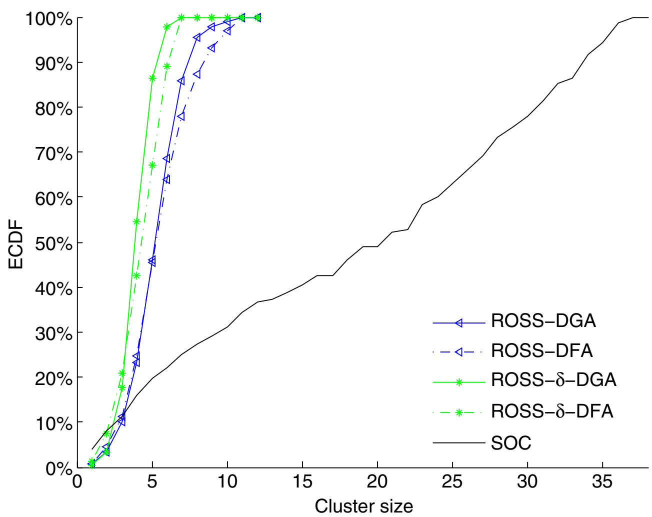

FIGURE 18 Empirical cumulative distribution function (ECDF) associated with cluster sizes when there are 100 cognitive radios and 30 primary users in network. DFA, distributed fast algorithm; DGA, distributed greedy algorithm; ROSS, Robust Spectrum Sharing; SOC, system on a chip

Next, we consider the robustness of the formed clusters in case that more and more PUs are added to the scenario. In this case, we increase PUs' activity by adding 20 batches of PUs sequentially in CRN, each batch including 10 PUs, which are placed randomly, and select a channel at random. Figures 14 and 15 show the corresponding results for $N = 100$ and 200 CR nodes in the scenario. We basically see that as the PU activity increases, more unclustered CR nodes result from SOC than the variants of ROSS. This corroborates a somewhat similar observation of the previous section. When $N = 300$, as shown in Figure 16, and the amount of newly added PUs is moderate, ROSS-DGA/DFA results in slightly more unclustered CR nodes than SOC, while SOC's performance deteriorates quickly when the number of PUs continues to increase. In addition, Figures 14 to 16 reveal that ROSS with size control mechanism results in significantly less singleton clusters.

Next, we turn to the size of the formed clusters under the different distributed schemes. For this, we study in Figure 17 the average number of total clusters formed under the different schemes for the different parameter combinations considered. The Figure shows that the number of clusters resulting from SOC increases linearly, whereas the number of formed clusters increases sublinearly in case of the variants of ROSS. This result coincides with the analysis in Section 5.1.3.

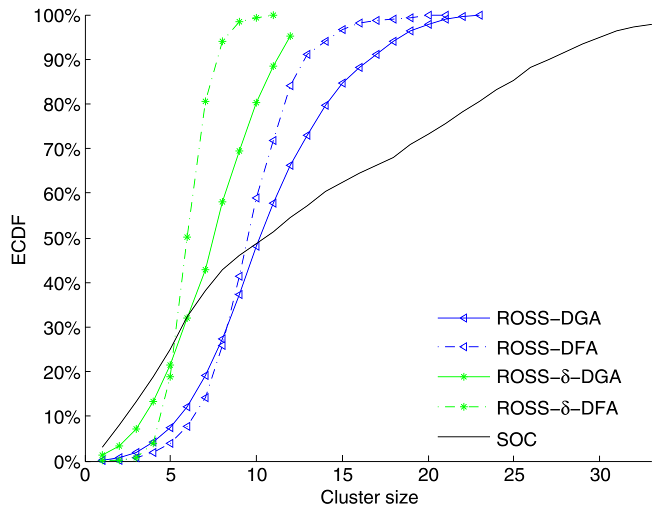

FIGURE 19 Empirical cumulative distribution function (ECDF) associated with cluster sizes when there are 200 cognitive radios and 30 primary users in network. DFA, distributed fast algorithm; DGA, distributed greedy algorithm; ROSS, Robust Spectrum Sharing; SOC, system on a chip

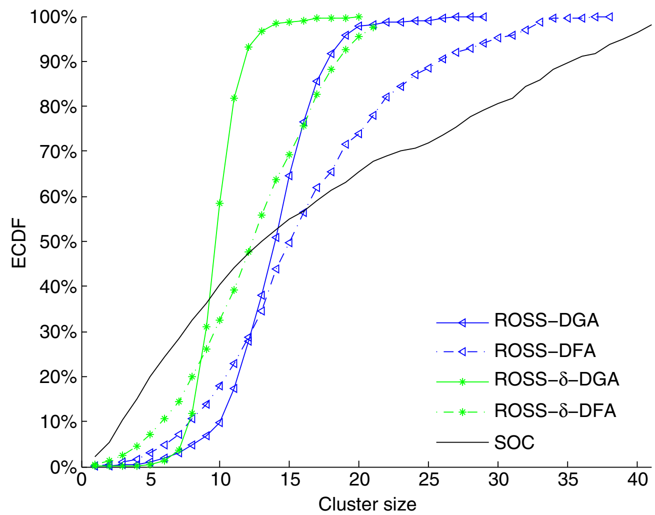

FIGURE 20 Empirical cumulative distribution function (ECDF) associated with cluster sizes when there are 300 cognitive radios and 30 primary users in network. DFA, distributed fast algorithm; DGA, distributed greedy algorithm; ROSS, Robust Spectrum Sharing; SOC, system on a chip

Furthermore, we consider the empirical distribution function of the size of the formed clusters, for each considered network density, in Figures 18, 19, and 20, respectively.

The ECDF associated with the cluster sizes shows that the cluster sizes resulting from the variants of ROSS are clearly influenced by the chosen desired size, ie, as shown in Figure 18, where the number of CR nodes is 100 and the desired cluster size is 6, 90% of CR nodes are in clusters whose sizes are between 3 and 9, whereas for SOC, only 17% of nodes are in the clusters with these sizes. Similarly, when $N = 200$ and the desired size is 12 (as shown in Figure 19), 80% of nodes are in clusters whose sizes are between 6 and 18, whereas only 30% of nodes are in clusters of similar sizes when SOC is executed. The cluster sizes from ROSS-δ-DGA and ROSS-δ-DFA concentrate more around the desired size than that of ROSS-DGA and ROSS-DFA.

We finally turn to the results of clustering under erroneous spectrum sensing. In Figure 21, we first study the impact of erroneous spectrum sensing and subsequent clustering on the number of CCs per cluster. The Figure shows that the average number of CCs decreases slightly when the false negative rate increases. Nevertheless, as with the previous investigated scenarios, the results do not show large differences between the distributed variants. Furthermore, we consider the ECDF of the size of the formed clusters under erroneous spectrum sensing in Figure 22. For all the schemes, when the rate of false negatives increases, the number of singleton clusters and smaller clusters increases accordingly. Clusters

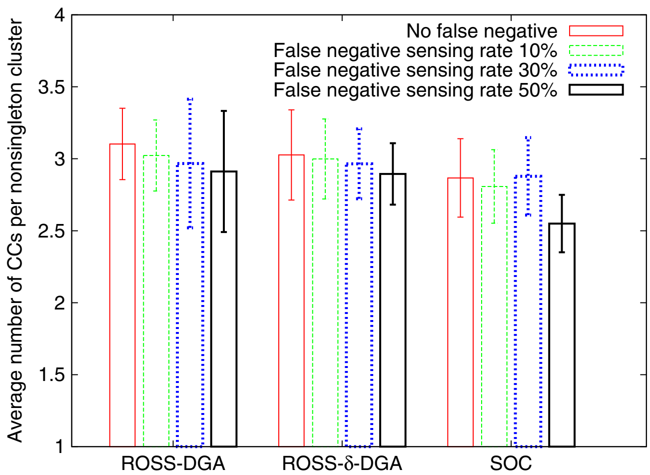

FIGURE 21 The number of common channels (CCs) per nonsingleton cluster with the presence of spectrum sensing false negative. DGA, distributed greedy algorithm; ROSS, Robust Spectrum Sharing; SOC, system on a chip

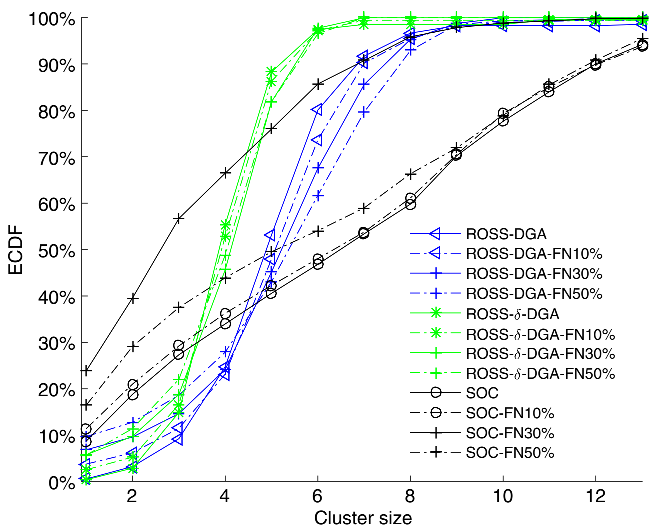

FIGURE 22 Empirical cumulative distribution function (ECDF) associated with cluster sizes, where there are 30 primary users and 100 cognitive radios with false negative in spectrum sensing. DFA, distributed fast algorithm; DGA, distributed greedy algorithm; ROSS, Robust Spectrum Sharing; SOC, system on a chip

formed by SOC are, furthermore, affected by the sensing errors significantly. More unclustered nodes are generated, and a lot of small clusters are formed, eg, when the false negative rate is 30%. In contrary, the ROSS variants are resilient in terms of unclustered nodes and cluster sizes. We can conclude that, due to the negotiation step within neighborhoods, ROSS variants successfully rule out the false negative channels resulting from erroneous spectrum sensing. This is an interesting and remarkable advantage of ROSS in comparison to SOC.

7 | CONCLUSION

In this paper, we investigate the robust clustering problem in CRN, give a mathematical description of the problem, and prove NP-hardness of it. Both centralized and distributed clustering solutions are proposed. With the increasing intensity of the PUs' activity, our proposed schemes generate clusters that make more secondary users to be in the clusters composed with multiple users, so that more secondary users can benefit from cooperative spectrum sensing. Besides, the resulting cluster sizes lie in a smaller range centered around the desired cluster size and involve less control messages than the comparison scheme. In particular, the proposed centralized scheme outperforms the proposed distributed schemes in all aspects, on the other hand it requires a centralized device and the involved control overhead is larger. Our proposed distributed scheme is more robust against erroneous spectrum sensing compared with the state-of-the-art schemes scheme. The simulation confirms that the metric of average number of CCs of clusters alone is not an accurate indicator for the cluster robustness against the PUs' activity.

ORCID

Di Li http://orcid.org/0000-0001-8116-1065
James Gross https://orcid.org/0000-0001-6682-6559

REFERENCES

1. Mitola J, Maguire GQ. Cognitive radio: making software radios more personal. *IEEE Pers Commun*. 1999;6(4):13-18. https://doi.org/10.1109/98.788210
2. Sahai A, Tandra R, Mishra SM, Hoven N. Fundamental design tradeoffs in cognitive radio systems. Paper presented at: Proceedings of the First International Workshop on Technology and Policy for Accessing Spectrum (TAPAS'06); 2006; Boston, MA.
3. Jacob J, Jose BR, Mathew J. Cellular automata approach for spectrum sensing in energy efficient sensor network aided cognitive radio. Paper presented at: Proceedings of the International Conference on Engineering of Complex Computer Systems (ICECCS); August 2012; Kochi, India.
4. Akyildiz IF, Lo BF, Balakrishnan R. Cooperative spectrum sensing in cognitive radio networks: a survey. *Phys Commun*. 2011;4(1):40-62. https://doi.org/10.1016/j.phycom.2010.12.003
5. Sun C, Zhang W, Letaief KB. Cluster-based cooperative spectrum sensing in cognitive radio systems. Paper presented at: 2007 IEEE International Conference on Communications; 2007; Glasgow, UK.
6. Kawadia V, Kumar PR. Power control and clustering in ad hoc networks. Paper presented at: Twenty-Second Annual Joint Conference of the IEEE Computer and Communications (INFOCOM'03); 2003; San Francisco, CA.
7. Krebs M, Stein A, Lora MA. Topology stability-based clustering for wireless mesh networks. Paper presented at: 2010 IEEE Global Telecommunications Conference (GLOBECOM'10); 2010; Miami, FL. https://doi.org/10.1109/GLOCOM.2010.5683417
8. Abbasi AA, Younis M. A survey on clustering algorithms for wireless sensor networks. *Comput Commun*. 2007;30(14-15):2826-2841. https://doi.org/10.1016/j.comcom.2007.05.024
9. Willkomm D, Bohge M, Hollós D, Gross J, Wolisz A. Double hopping: A new approach for dynamic frequency hopping in cognitive radio networks. Paper presented at: 2008 IEEE 19th International Symposium on Personal, Indoor and Mobile Radio Communication (PIMRC'08); 2008; Cannes, France.
10. Passiatore C, Camarda P. A centralized inter-network resource sharing (CIRS) scheme in IEEE 802.22 cognitive networks. Paper presented at: Proceedings of the 10th IFIP Annual Mediterranean Ad Hoc Networking Workshop; 2011; Sicily, Italy. https://doi.org/10.1109/Med-Hoc-Net.2011.5970487
11. Wu Q, Ding G, Wang J, Li X, Huang Y. Consensus-based decentralized clustering for cooperative spectrum sensing in cognitive radio networks. *Chin Sci Bull*. 2012;57(28-29):3677-3683. https://doi.org/10.1007/s11434-012-5074-6
12. Zhang H, Zhang Z, Dai H, Yin R, Chen X. Distributed spectrum-aware clustering in cognitive radio sensor networks. Paper presented at: 2011 IEEE Global Telecommunications Conference (GLOBECOM'11); 2011; Kathmandu, Nepal.
13. Jorio A, El Fkihi S, Elbhiri B, Aboutajdine D. An energy-efficient clustering routing algorithm based on geographic position and residual energy for wireless sensor network. *J Comput Netw Commun*. 2015;2015. https://doi.org/10.1155/2015/170138. https://www.hindawi.com/journals/jcnc/2015/170138/

14. Li D, Gross J. Robust clustering of ad-hoc cognitive radio networks under opportunistic spectrum access. Paper presented at: 2011 IEEE International Conference on Communications (ICC'11); 2011; Kyoto, Japan.
15. Liu S, Lazos L, Krunz M. Cluster-based control channel allocation in opportunistic cognitive radio networks. *IEEE Trans Mob Comput*. 2012;11(10):1436-1449.
16. Zhao J, Zheng H, Yang GH. Spectrum sharing through distributed coordination in dynamic spectrum access networks. *Wirel Com Mob Comput*. 2007;7(9). https://doi.org/10.1002/wcm.v7:9
17. Chen T, Zhang H, Maggio G, Chlamtac I. CogMesh: A cluster-based cognitive radio network. Paper presented at: 2007 2nd IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks (DySPAN'07); 2007; Dublin, Ireland.
18. Baddour K, Ureten O, Willink T. Efficient clustering of cognitive radio networks using affinity propagation. Paper presented at: Proceedings of 18th International Conference on Computer Communications and Networks (ICCCN'09); 2009; San Francisco, CA. https://doi.org/10.1109/ICCCN.2009.5235219
19. Wu D, Cai Y, Zhou L, Wang J. A cooperative communication scheme based on coalition formation game in clustered wireless sensor networks. *IEEE Trans Wirel Commun*. 2012;11(3):1190-1200. https://doi.org/10.1109/TWC.2012.012712.111049
20. Asterjadhi A, Baldo N, Zorzi M. A cluster formation protocol for cognitive radio ad hoc networks. Paper presented at: 2010 European Wireless Conference (EW'10); 2010; Lucca, Italy. https://doi.org/10.1109/EW.2010.5483442
21. Ozger M, Akan OB. Event-driven spectrum-aware clustering in cognitive radio sensor networks. Paper presented at: Proceedings of the 2013 IEEE INFOCOM; 2013; Turin, Italy. https://doi.org/10.1109/INFCOM.2013.6566943
22. Mansoor N, Muzahidul Islam AKM, Zareei M, Baharun S, Komaki S. Construction of a robust clustering algorithm for cognitive radio ad-hoc network. Paper presented at: Proceedings of the 10th International Conference on Cognitive Radio Oriented Wireless Networks (CROWNCOM); April 21-23, 2015; Doha, Qatar.
23. Clark B, Colbourn C, Johnson D. Unit disk graphs. *Ann Discrete Math*. 1991;48:165-177. https://doi.org/10.1016/S0167-5060(08)71047-1
24. Zhang Y, Yu G, Li Q, Wang H, Zhu X, Wang B. Channel-hopping-based communication rendezvous in cognitive radio networks. *IEEE/ACM Trans Networking*. 2014;22(3):889-902. https://doi.org/10.1109/TNET.2013.2270443
25. Gu Z, Hua Q-S, Dai W. Fully distributed algorithms for blind rendezvous in cognitive radio networks. Paper presented at: Proceedings of the 15th ACM international symposium on Mobile ad hoc networking and computing (MobiHoc'14); 2014; Philadelphia, PA.
26. Li D, Fang E, Gross J. Versatile Robust Clustering of Ad Hoc Cognitive Radio Network. ArXiv e-prints, 1704.04828.
27. MacKenzie A, Wicker S. Game theory in communications: Motivation, explanation, and application to power control. Paper presented at: 2001 IEEE Global Telecommunications Conference (GLOBECOM'01); 2001; San Antonio, TX. https://doi.org/10.1109/GLOCOM.2001.965533
28. Neel JO. Analysis and Design of Cognitive Radio Networks and Distributed Radio Resource Management Algorithms [PhD dissertation]. Blacksburg, VA: Virginia Polytechnic Institute and State University; 2006.
29. Wang B, Wu Y, Liu KR. Game theory for cognitive radio networks: an overview. *Comput Netw*. 2010;54(14):2537-2561.
30. Chee BJS, Franklin Jr C. *Computing Cloud Technologies and Strategies of the Ubiquitous Data Center*. 1st ed. New York, NY: CRC Press Inc; 2010.
31. Ackermann H, Rglin H, Vcking B. Pure Nash equilibria in player-specific and weighted congestion games. *Theor Comput Sci*. 2009;410(17):1552-1563. https://doi.org/10.1016/j.tcs.2008.12.035
32. Li D, Fang E, Gross J. Robust Clustering in Cognitive Radio Network with Cluster Size Control. 2017. https://arxiv.org/abs/1704.04828
33. Li X-Y, Wang Y, Wang Y. Complexity of data collection, aggregation, and selection for wireless sensor networks. *IEEE Trans Comput*. 2011;60(3):386-399. https://doi.org/10.1109/TC.2010.50
34. Onus M, Richa A, Kothapalli K, Scheideler C. Efficient broadcasting and gathering in wireless ad-hoc networks. Paper presented at: Proceedings of the 8th International Symposium on Parallel Architectures, Algorithms and Networks (ISPAN'05); 2005; Las Vegas, NV. https://doi.org/10.1109/ISPAN.2005.43
35. Garey MR, Johnson DS. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. New York, NY: W. H. Freeman; 1979.

How to cite this article: Li D, Fang E, Gross J. Robust clustering for ad hoc cognitive radio network. *Trans Emerging Tel Tech*. 2018;e3285. https://doi.org/10.1002/ett.3285

APPENDIX A

PSEUDOCODE FOR ALGORITHMS 1, 2, and 3

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

Input: $d_j, g_j, j \in \text{Nb}(i) \setminus \Lambda$, Λ denotes the set of cluster heads among $\text{Nb}(i)$. Empty sets τ_1, τ_2.
Result: Returning 1 means i is cluster head, and d_j is set to $0, j \in \text{Nb}(i) \setminus \Lambda$. Returning 0 means i is not cluster head.

if $\nexists j \in \text{Nb}(i) \setminus \Lambda$, such that $d_i \geq d_j$ **then**
  return 1;
end
if $\exists j \in \text{Nb}(i) \setminus \Lambda$, such that $d_i > d_j$ **then**
  return 0;
else
  if $\nexists j \in \text{Nb}(i) \setminus \Lambda$, such that d_j equals 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 equals g_i **then**
    $\tau_2 \leftarrow j$
  end
end
if ID_i is smaller than any $\text{ID}_j, j \in \tau_2 \setminus i$ **then**
  return 1;
end
return 0;

Algorithm 2: ROSS phase I: cluster head guarantees the availability of CC (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 CC or satisfies the requirement on cluster size

1 **while** $K_C = \emptyset$ **do**
2   **while** $|C| > t \cdot \delta$ **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 min**$(|\cap_{j \in C \setminus i} K_j| - |\cap_{j \in C} K_j|)$ **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 $C \in S_i$, afterwards node i notifies all its claiming clusters in S_i about its affiliation decision.

while i has not chosen a cluster, or i has chosen cluster $\tilde{C}$, but $\exists C' \in S_i (C' \neq \tilde{C})$, which has $|K(C' \setminus i)| - |K(C')| < |K(\tilde{C} \setminus i)| - |K(\tilde{C})|$ **do**
  if $\exists$ only one $C \in S_i$, $C = \mathbf{arg\,min} \sum_{C' \in S_i \setminus C} (|K(C' \setminus i)| - |K(C')|)$ **then**
    return C;
  else
    $\exists$ multiple C satisfying the above condition, then $\tau_1 \leftarrow C$;
  end
  if $\exists$ only one $C \in \tau_1$, $C = \mathbf{arg\,min} (K_{h(C)} \cap K_i)$ **then**
    return C;
  else
    $\exists$ multiple C satisfying the above condition, then $\tau_2 \leftarrow C$;
  end
  if $\exists$ only one $C \in \tau_2$, $C = \mathbf{arg\,min} |C|$ **then**
    return C;
  else
    return $\mathbf{arg\,min}_{C \in \tau_2} h(C)$;
  end
end

APPENDIX B

PROOF OF LEMMA 1

Proof. We consider a CRN that is represented by a connected graph. To simplify the discussion, we assume that secondary users have unique individual connectivity degrees.

For the sake of contradiction, let us assume that there exist a secondary user α that is not included into any cluster. Then, there exists a node $\beta \in \mathrm{Nb}(\alpha)$ such that $d_\alpha > d_\beta$ (otherwise α becomes cluster head). In this case, according to Algorithm 1, β is not included into any cluster because, otherwise $d_\beta = M$, a large positive integer, which contradicts to $d_\alpha > d_\beta$. Now, we distinguish between 2 cases. If β becomes cluster head, node α is included, the assumption that α is not included in any cluster 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 a neighboring node γ with $d_\gamma < d_\beta$. So far, when no cluster head is identified, the unclustered nodes, ie, α and β, form a linked list, where their individual connectivity degrees monotonically decrease. However, this list will not continue growing 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 A1.

In this example, node ω is at the tail of a 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 neighbors (here, we assume that every newly formed cluster has CCs) including the nodes that precede ω in the list. The nodes that join a cluster set their individual connection degrees to J, which makes the node immediately precede in the list to become a cluster head.

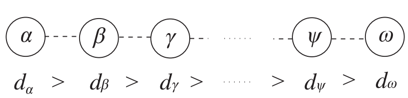

FIGURE A1 The node series discussed in the proof of Theorem 2. The deduction begins from node α

In this way, cluster heads are generated from the tail to the head in the list, and every node in the list is in at least one cluster, which contradicts the assumption that α is not included in any cluster. □

APPENDIX C

PROOF OF THEOREM 1

Proof. To prove that the robust clustering problem is NP-hard, we reduce the *maximum weighted k-set packing problem*, which is NP-hard when $k \geqslant 3$ to the robust clustering problem to show the latter is at least as hard as the former.[35] Given a collection of sets of cardinality at most k and with weights for each set, the maximum weighted packing problem is that of finding a collection of disjoint sets of maximum total weight. The decision version of the weighted k-set packing problem is as follows.

Definition 2. Given a finite set $\mathcal{G}$ of nonnegative integers where $\mathcal{G} \subsetneq \mathbb{N}$ and a collection of sets $\mathcal{Q} = \{S_1, S_2, \dots, S_m\}$, where $S_i \subseteq \mathcal{G}$ and $\max(|S_i|) \geq 3$ for $1 \leq i \leq m$. Every set S in $\mathcal{Q}$ has a weight $\omega(S) \in \mathbb{N}^+$. The problem is to find a collection $\mathcal{I} \subseteq \mathcal{Q}$ such that $\mathcal{I}$ contains only the pairwise disjoint sets and the total weight of these sets is greater than a given positive number λ, ie, $\sum_{\forall S \in \mathcal{I}} \omega(S) > \lambda$.

We will show that the weighted k-set packing problem $\leq_P$ CRN robust clustering problem. Given an instance of the weighted k-set packing problem, ie, a collection of sets $\mathcal{Q} = \{S_1, S_2, \dots, S_m\}$, where each set $S_i, i \in \{1, 2, \dots, m\}$ consists of positive integers. There is an integer weight $\omega(S_i)$ for S_i, and in the end, an integer λ completes the description of this instance. We will construct an instance of a CRN robust clustering problem within polynomial time. Without loss of generality, we let set $\cup_{i \in \{1,2,\dots,m\}} S_i = \{1, 2, \dots, N\} = \mathcal{P}$.

We will construct the CRN and the clusters as follows. For every set $S \in \mathcal{Q}$, there will be a corresponding cluster composed with CR nodes constructed. For the set whose size is larger than 1, the IDs of the constructed CR nodes are identical with the elements in it, and we locate the CR nodes so that any two of them can communicate directly when CCs are available on them. Besides, a set of channels with cardinality of $|\omega(S)|$ is allocated to all the CR nodes in this cluster, and the channels are on the spectrum band that is exclusive for this cluster. For the set S that contains only one element, ie, $S = \{t\}$, where $t \in \mathcal{P}$, a cluster composed with 2 CR nodes will be created, ie, one CR node's ID is t, the other CR node is the dummy node of the former and its ID is $t+N$. Afterwards, $|\omega(S)|$ channels that are exclusively allocated to this cluster are assigned to these 2 CR nodes. Now, we have constructed the clusters that correspond to every set in $\mathcal{Q}$. In the end, singleton clusters are formed with respect to every element in $\mathcal{P}$. Note that the CCs in these singleton clusters do not contribute to the sum of $f(C)$. The existence of the singleton clusters ensures that it is always possible to find out a group of clusters, which together constitute the whole CRN.

Actually, all the constructed CR nodes can be assumed to locate in a very small area so that each CR node is within the transmission scope of every other CR node. Note that each constructed cluster maintains certain CCs that are exclusive to other clusters. This rule in transformation makes the transformation for every set in $\mathcal{Q}$ to be feasible.

Given any instance $\mathcal{I}$ of the maximum weighted k-set packing problem, we can now formulate it to an instance $\mathcal{T}$ of the robust clustering problem. For each set in $\mathcal{I}$, we find out the corresponding clusters, as well as the singleton clusters, whose IDs are from $\mathcal{P} \setminus (\cup S_i), S_i \in \mathcal{I}$. Then, the sum of the weights of the corresponding sets is equal to the sum of $f(C)$ and, thus, greater than λ.

Based on the aforementioned construction and transformation, if robust clustering black box says yes to a clusters formation $\mathcal{T}$, then the corresponding $\mathcal{I}$ is an instance for the maximum weighted k-set packing problem. If the clusters formation $\mathcal{T}$ is not a solution for the robust clustering problem, then $\mathcal{I}$ is an instance for the maximum weighted k-set packing problem.

Since the construction and transformation from instance to clusters formation take polynomial time, we can conclude that the robust clustering problem is NP-hard. □