TRANSACTIONS ON EMERGING TELECOMMUNICATIONS TECHNOLOGIES
Trans. Emerging Tel. Tech. 2017; **00**:1–15
DOI: 10.1002/ett

RESEARCH ARTICLE

Robust Clustering for Ad Hoc Cognitive Radio Network

Di Li[1]*, Erwin Fang[2], James Gross[3]

RWTH Aachen University[1], Swisscom (Schweiz) AG[2], KTH Royal Institute of Technology[3]

ABSTRACT

***Correspondence**
Chair of Communication and Distributed Systems Ahornstrasse 55 - building E3 52074 Aachen Germany
Email: li@umic.rwth-aachen.de

1. INTRODUCTION

Cognitive radio (CR) is a promising approach to mitigate the increasing scarcity of radio spectrum [**?**] 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, while unlicensed users may access the spectrum when it is not utilized. This can be realized by so-called opportunistic spectrum access, i.e., unlicensed users access the spectrum only after validating that the channel is currently unoccupied. In the context of cognitive radio, licensed users are also called primary users (PU), while unlicensed users are often referred to as secondary users and constitute a cognitive radio network (CRN)*. For CRN, accurate spectrum sensing is critical, and the rate of false negatives, i.e., the likelihood of misdetecting active primary users, needs to be minimized [**?**]. 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 [**?**, **?**]. Thus, clustering of secondary nodes is regarded as a necessary condition to realize cooperative spectrum sensing [**?**] 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 [**?**], improve routing performance [**?**], and improve the network lifetime and coverage [**?**]. For cognitive radio networks, apart from improving the sensing accuracy, clustering also improves spectrum utilization among several cognitive radio networks by allowing for coordination in particular when CRNs have to vacate channels [**?**], while also been known for reducing the interference between cognitive clusters [**?**], and improving routing [**?**].

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

On the other hand, the larger the cluster size is, the lower is in general the set of CCs that all nodes of a cluster observe as unoccupied by primary users. This is due to the fact that in general, secondary nodes at different spatial locations will be able to sense the activity of different primary users due to different channel characteristics. Thus, a trade-off arises for the formation of robust cognitive radio clusters: On the one hand, a low number of nodes in a cluster is desirable, as it generally provides more nodes with a common observation of primary user 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, in particular

*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 to nodes typically in the context of the network topology.

if only one or two nodes are members of a cluster [**?**]. One therefore needs to strike a balance between the *size* of a cluster and the *number* of common channels per cluster, to balance robustness and sensing accuracy. Cluster size plays furthermore a role in transmit power consumption, i.e., the cluster size affects the transmit power consumption under certain routing schemes [**?**, **?**].

In this paper, we analytically study the above mentioned trade-off 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 as well as distributed algorithms. We propose an alternate metric to measure cluster robustness in contrast to previous works [**?**] and [**?**]. We claim that cluster robustness can not 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 primary user activity. Our proposed distributed scheme extends our previous work ROSS (Robust Spectrum Sharing) [**?**] by additionally incorporating control over the size of a cluster. Throughout this paper, we call the newly proposed distributed schemes *variants of ROSS*. The rest of the paper is organized as follows: In Section 2, we review related work in particular 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 in [**?**]. Our system model as well as the problem statement with respect to the robust clustering problem are presented in Section 3. The main contribution, the centralized and distributed solutions are introduced in Section 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 review briefly state-of-the-art regarding clustering in CRN, with an emphasis on robust clustering.

With regard to forming clusters in CRN, deciding on the common channel within each cluster is the foremost question to answer. [**?**, **?**, **?**] propose different clustering schemes and enforce that every cluster possesses at least one CC. The clustering scheme in [**?**] looks for a network partition which improves the accuracy of spectrum sensing without considering robustness. In [**?**] clusters are formed by deciding on the cluster heads, where the transmit power for the long-haul transmission between the cluster heads is minimized. [**?**] proposes a cluster structure which imprroves energy efficiency. Furthermore, [**?**] proposes a strategy on how to decide on the CCs and access multiple CCs within clusters. An event-driven clustering scheme is proposed for cognitive radio sensor networks in [**?**]. However, none of the above mentioned schemes provide robustness of the clusters against random primary user activity.

The authors of [**?**] propose a clustering algorithm which aims at speeding up the process of re-clustering in case that primary user activity eliminates all CCs. However, this work does not consider cluster robustness in the first place, but rather focuses on reactive measures. [**?**] presents 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 SOC is proposed in [**?**], 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%-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 which accesses abundant channels may form a cluster solely by itself, 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 [**?**, **?**]. In [**?**] we propose a distributed clustering scheme ROSS (Robust Spectrum Sharing) 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 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. Summarizing, both this work and as well as the presentation of SOC define robustness just to be the number of CCs per cluster, and don't consider the sustainability of clusters against increasing activity of primary users, leaving the issue of cluster robustness open.

3. SYSTEM MODEL AND PROBLEM FORMULATION

We consider a set of CR users $\mathcal{N}$ and a set of primary users distributed over a given area. A set of licensed channels $\mathcal{K}$ is available for the primary users. The CR users are allowed to transmit on channel $k \in \mathcal{K}$ only if no primary user is detected to be occupying channel k. CR users conduct spectrum sensing independently and sequentially on all licensed channels.[†] We adopt the unit disk model [**?**] for both primary and CR users' transmission. Thus, if a CR

[†] We assume that every node can detect the presence of an active primary user on each channel with certain accuracy. The spectrum availability can be validated with a certain probability of detection. While we do argue that too small cluster

Trans. Emerging Tel. Tech. 2017; **00**:1–15
DOI: 10.1002/ett
Prepared using ettauth.cls

node i locates within the transmission range of an active primary user p, i is not allowed to use the channel which is being used by p. We assume the primary users to change their operation channels slowly, 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 primary users is limited and CR users have different locations, different CR users have different views of the spectrum availability, i.e., 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 i, 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 there is one dedicated control channel which is used to exchange signaling messages during the clustering process. This control channel could be one of the ISM bands or other reserved spectrum which is exclusively used for transmitting control messages.[‡] Over the control channel, a secondary user i can exchange its spectrum sensing result K_i with all its one hop neighbors $\mathrm{Nb}(i)$. In the following, we refer to licensed channels as channels in general, and will explicitly mention the dedicated control channel if necessary.

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 common channels which are 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, i.e. $K(C) = \bigcap_{i \in C} K_i$. We also summarize all used notation in Table I.

3.1. Robust Clustering Problem in CRN

As introduced in Section 1, robustness of the clusters is their ability to uphold with the influence of the active primary users, and it is represented by the number of secondary users which are not included in any cluster. To achieve better robustness, we propose that cluster should be formed by increasing the number of CCs. Meanwhile, the sizes of the formed clusters should be regulated, i.e., they don't diverge from the desired cluster size greatly.

Definition 1. *Robust clustering problem in CRN.*

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

As to a cognitive radio network where the set of CR nodes is $\mathcal{N}$, the robust clustering problem is to determine a set of clusters $\mathcal{T}$, where

1. *the intersection of any two clusters in $\mathcal{T}$ is an empty set*
2. *the union of all nodes in all clusters in $\mathcal{T}$ is $\mathcal{N}$*
3. *the sum over $f(C)$ for all clusters C is maximized, where the number of common channels for cluster C is denoted as $f(C)$*
4. *all cluster sizes are within the range $[\delta_1, \delta_2]$, with $\delta_1, \delta_2 \in \mathbb{Z}^+$ and $\delta_1 \leq \delta_2$. When the cluster size is larger or smaller than the range $[\delta_1, \delta_2]$, $f(C)$ is defined as 0.*
5. *the size of C in $\mathcal{T}$ is allowed to be 1.*

The decision version of this problem is to determine whether there exists a set of clusters, say $\mathcal{X}$, so that $\cup_{C \in \mathcal{X}} C = \mathcal{N}$, and $\sum_{C \in \mathcal{X}} f(C) \geqslant \lambda$ where λ is a positive integer number. We have the following theorem on the problem's complexity.

Theorem 3.1. *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

When the global knowledge of the CRN i.e., the locations of primary users and their working channels, and the

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 [**?**, **?**], i.e., two neighboring nodes establish communication on the same channel.

locations of secondary users and available channels on them, we can propose a centralized scheme. We obtain the set of $\mathcal{S}$ which contains all the clusters in $\mathcal{N}$, i.e., $\mathcal{S} = \{C_1, C_2, \ldots, C_i, \ldots, C_{|\mathcal{S}|}\}$ [§] and there is $\bigcup_{1 \le i \le |\mathcal{S}|} C_i = \mathcal{N}$. The proposed centralized solution formulates the problem in Definition 1 as an optimization problem which is solved with standard software packages. The optimization decides on the clusters according to the following optimization formulation, which is a binary linear programming problem and can be solved by many available solvers.

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

y_i and x_{ij} are two binary variables. Being either 1 or 0, y_i denotes whether the i-th cluster C_i in $\mathcal{S}$ is chosen or not. x_{ij} indicates whether the CR node j resides in the cluster C_i, i.e., $x_{ij} = 1$ means node j resides in the cluster C_i. N is the total number of CR users in network $\mathcal{N}$, M is the number of clusters in $\mathcal{S}$.

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

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

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

where constant $q_{ij} = |K(C_i)|$ when node $j \in C_i$, and $q_{ij} = 0$ when node $j \notin C_i$. $p(C_i)$ is the size-related weight, which is positively related with the difference between C_i's size and the desired size. Assuming δ is the desired size, then weight p is decided with respect to cluster sizes 1,2, $\cdots$, σ as follows,

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

where $0 < \rho_1 < \rho_2 < \cdots < \rho_\sigma$.

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

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

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

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

The difficulty of using this method lies in obtaining the set $\mathcal{S}$. In the worst case, i.e., every CR node communicates directly with any other node and the CRN forms a full connected graph, the size of $\mathcal{S}$ 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 the proximity of the available spectrum in their neighborhood after a series of interactions with their neighbors. The variants of ROSS consist of two cascaded phases: *cluster formation* and *membership clarification* as shown in Figure 1.

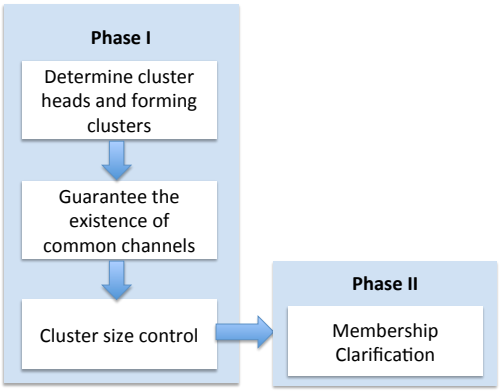


Figure 1. Processing steps of ROSS

In the first phase, clusters are formed quickly and every CR user becomes either cluster head or cluster member, cluster size control is also implemented . In the second phase, non-overlapping clusters are formed in a way that the CCs of relevant clusters are mostly increased.

5.1. Phase I - Cluster Formation

Before conducting clustering, we assume spectrum sensing and neighbor discovery are completed, and neighboring nodes exchange their channel availabilities via the dedicated control channel. As a result, every CR node is

Trans. Emerging Tel. Tech. 2017; **00**:1–15 © 2017 John Wiley & Sons, Ltd.
DOI: 10.1002/ett
Prepared using ettauth.cls

aware of the available channels for themselves and their neighbors. In this phase, cluster heads are determined after a series of comparisons with their neighbors. Two metrics are proposed to characterize the proximity in terms of available spectrum between CR node i and its neighborhood, which will be used to decide on cluster heads.

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

Individual connectivity degree d_i and neighborhood connectivity degree g_i together form the *connectivity vector* (d_i, g_i). Connectivity vector is calculated by every secondary user after channel availability is obtained, and is then broadcasted. Figure 2 shows a CRN, where dashed edge indicates the end nodes are within each other's transmission range, the number along the dashed line is the number of common channels between the two ends. The sets of the indices of the available channels sensed by each node are: $K_A = \{1, 2, 3, 4, 5, 6, 10\}, K_B = \{1, 2, 3, 5, 7\}$, $K_C = \{1, 3, 4, 10\}, K_D = \{1, 2, 3, 5\}$, $K_E = \{2, 3, 5, 7\}, K_F = \{2, 4, 5, 6, 7\}$, $K_G = \{1, 2, 3, 4, 8\}, K_H = \{1, 2, 5, 8\}$. Each node's connectivity vector is calculated and shown.

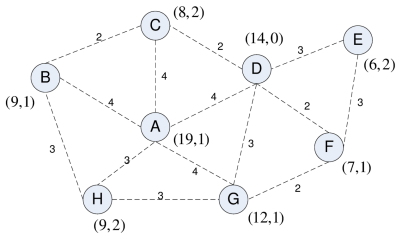


Figure 2. Connectivity graph of a CRN and the connectivity vector (d_i, g_i) for each node. Primary users are not shown.

5.1.1. Determining Cluster Heads and Forming Clusters

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 neighbors. When CR node i has lower individual connectivity degree than any of its neighbors except for those which have already been identified as cluster heads, node i becomes a cluster head. If there is a CR node j in i's neighborhood, and has the same individual connectivity degree as i, i.e., $d_j = d_i$ and $d_j < d_k, \forall k \in \text{Nb}(j) \setminus \{\Lambda \cup i\}$ where Λ denotes cluster 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, i.e., the one with smaller node ID becomes the cluster head. The node which is identified as cluster head broadcasts a message to notify its neighbors of this news, then its neighbors which are not cluster heads become its cluster members. [¶] During the whole phase I, whenever a CR node becomes cluster head (thus accordingly forms a cluster), or its cluster's composition is changed, the cluster head broadcasts the new information about its cluster, which includes the sets of available channels on itself and all its cluster members. The pseudo code for the cluster head decision and the initial cluster formation is shown in Algorithm 1 in appendix.

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 $M > |\mathcal{K}| \cdot N$, and broadcasts the new individual connectivity degree. When node i is associated with multiple clusters, i.e., i has received multiple notifications from different cluster heads, d_i is still set to be M. The manipulation of the individual connectivity degree of the cluster members accelerates the decision on the cluster heads.

5.1.2. The Existence of Common Channels

After executing Algorithm 1, certain formed clusters may not possess any CCs. As decreasing cluster size increases CCs within a cluster, for those clusters without CCs, certain nodes need to be excluded to obtain at least one CC. The sequence of removing is performed according to an ascending list of nodes, which are sorted according to 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 will be removed first. When there are multiple nodes having the same amount of common channels with the cluster head, the node whose absence brings in more common channels will be removed. If this criterion meets a tie, the tie can be broken by removing the node with smaller node ID. It is possible that the cluster head removes all its neighbors before obtaining CCs, which results in a singleton cluster which is composed by itself. The pseudo code for this procedure is shown in Algorithm 2. As for the nodes which 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 afterwards according to Theorem 5.1.

5.1.3. Cluster Size Control in Dense CRN

Both analysis and simulation [**?**] 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

[¶] The presence of cluster heads in the neighborhood of a newly formed cluster head will be explained in Section 5.1.2 and 5.1.3)

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. The desired size δ is decided based on the capability of the CR users and the tasks to be conveyed. We set the threshold as $t \cdot \delta$, where the constant parameter t is dependent on the network density and CR nodes' transmission range. We adopt t to be between 1 and the ratio of the average neighborhood size and desired size. When t is smaller, e.g., $t = 1$, the formed cluster in the phase I will be δ. For the cluster whose members are included by other clusters, the size of this cluster will be smaller than δ after the following membership clarification phase. If t is chosen large, e.g., $t \cdot \delta$ equals to the size of neighborhood, the mechanism of adjusting the cluster size will not work any more.

The cluster head removes the cluster members sequentially according to the following principle, the absence of one cluster member leads to the maximum increase of CCs in the cluster. The removed nodes restore their original individual connectivity degrees. This process ends when each cluster's size is smaller or equal to $t \cdot \delta$. This procedure is similar with that in Section 5.1.2, thus Algorithm 2 can be reused.

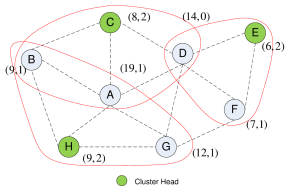


Figure 3. Clusters formation after the phase I of ROSS. Nodes A, B, D are debatable nodes as they belong to multiple clusters.

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

Lemma 5.1. *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.

Lemma 5.2. *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 M. 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 degrees* is still smaller than these nodes. Note that when the removed node becomes cluster head, it will not include its former cluster head i, so that i doesn't become cluster member and its *individual connectivity degrees* doesn't change. □

Lemma 5.3. *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 can be easily obtained from 5.2 considering the N is the number of all the secondary users in the CRN. Based on these Lemmas, we can easily obtain Theorem 5.1,

Theorem 5.1. *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 Lemma 5.1 and Lemma 5.3, the theorem is proved.

As Algorithm 1 is executed concurrently by different secondary users, the required time can be considerably reduced. If we apply Algorithm 1 to the CRN in Figure 2, the outcome can be found in Figure 3. 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 $\{H, B, A, G\}$.

5.2. Phase II - Membership Clarification

As to the resulted clusters shown in Figure 3 after running phase I of ROSS, we notice that nodes A, B, D are included in more than one cluster. We refer to these nodes as *debatable nodes* as their cluster affiliations are not decided. The clusters which include the debatable node i are called *claiming clusters* of node i, and the set of these clusters is denoted as S_i. The debatable nodes 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)

When a debatable node i decides one cluster $C \in S_i$ to stay and leaves the other its claiming clusters, the principle for i is that its move should result in the greatest increase of CCs in all its claiming clusters. As node i has been notified of the spectrum availability on all the nodes in each claiming cluster, node i is able to calculate how many more CCs can be produced in a claiming cluster if i leaves that cluster. Then node i decides on the cluster $C \in S_i$, if i leaving cluster C results in less increased CCs than leaving any other claiming clusters in S_i. When

Trans. Emerging Tel. Tech. 2017; **00**:1–15 © 2017 John Wiley & Sons, Ltd.
DOI: 10.1002/ett
Prepared using ettauth.cls

there comes a tie between two claiming clusters, i chooses to stay in the cluster whose cluster head shares the most CCs with i. When a tie still exists, node i chooses to stay in the claiming cluster which has the smallest size. Node IDs of cluster heads will be used to break tie in the end if necessary. The pseudo code of this algorithm is given in 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, and the members of these claiming clusters can be changed by other debatable nodes' moves. As a result, the debatable node which have made decision may not be content with its original move. There is concern that 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, modeling and analyzing the interactions among individuals. A game consists of three elements: a set of players, a selfish utility for each player, and a feasible strategy space for each player. In a game, the players are rational and intelligent decision makers, which are related with one explicit formalized incentive expression (the utility or cost). Game theory provides standard procedures to study its equilibriums [**?**]. In the past few years, game theory has been extensively applied to problems in communication and networking [**?**, **?**]. Congestion game is an attractive game model which describes the problem where participants compete for limited resources in a non-cooperative manner, it has the good property that Nash equilibrium can be achieved after finite steps of best response dynamic, i.e., each player chooses the strategy to maximize/minimize its utility/cost with respect to the other players' strategies. The framework of the congestion game has been used to model certain problems in internet-centric applications or cloud computing, where self-interested clients compete for the centralized resources and meanwhile interact with each other. For example, server selection is involved in distributed computing platforms [**?**], or users downloading files from cloud, etc.

To formulate the debatable nodes' membership clarification into the desired congestion game, we reexamine this process from a different/opposite perspective. From the new perspective, the debatable nodes are not included in any cluster and they need to decide on one cluster to join. When a debatable node i join 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.

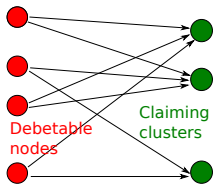


Figure 4. Debatable nodes and claiming clusters

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

- $\mathcal{P}$, the set of players in the game, which are the debatable nodes in our problem.
- $\mathcal{R} = \cup S_i, i \in \mathcal{P}$, the 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, then only one piece of resource will be allocated to i.
- The cost function $f(C)$ as to a resource C. $f(C) = \Delta|K^i(C)|, C \in S_i$, which represents the decreased number of CCs in cluster C when debatable node i joins C. As to cluster $C \in S_i$, the decrease of CCs caused by 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 non-decreasing 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 [**?**]. 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 [**?**]. In this game, every player greedily updates its strategy (choosing one claiming cluster to join) if joining a different claiming cluster minimizes the decrease of CCs $\sum_{i:C\in S_i}\Delta|K^i(C)|$, and a player's strategy in the game is exactly the same with the behaviour of a debatable node in the membership clarification phase.

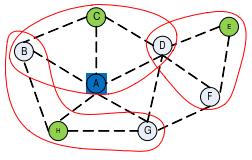
As to singleton congestion game, there exists a pure equilibria which can be reached with the best response update, and the upper bound for the number of steps before

convergence is $n^2 * m$ [**?**], where n is the number of players, and m is the number of resources. In our problem, the players are the debatable nodes, and the resources are the claiming clusters. Thus the number of steps can be expressed as $\mathcal{O}(N^3)$. In fact, the upper bound for the number of steps which 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% to 60%. On the other hand, the number of clusters heads is dependent on the network density and the CR node's transmission range as mentioned in Section 5.1. The simulation in [**?**] shows the cluster heads account for from 3.4% to 20% of the total CR nodes with the increase of network density. Furthermore, as the game played locally and in parallel i.e., a debatable node can only interact with a few claiming clusters, which greatly accelerates the execution speed.

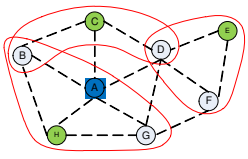
5.2.3. Distributed Fast Algorithm (DFA)

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 only once. The debatable nodes consider their claiming clusters to include all their debatable nodes, thus the membership of claiming clusters is static and all the debatable nodes can make decisions simultaneously without considering the change of membership of their claiming clusters. As ROSS-DFA is quicker than ROSS-DGA, the former is especially suitable for the CRN where the channel availability changes frequently. To run ROSS-DFA, debatable nodes execute only one loop in Algorithm 3.

Now we apply both ROSS-DGA and ROSS-DFA to the network in Figure 3 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 5. In Figure 5(a), node A staying in $C(C)$ and leaving $C(H)$ brings 2 more CCs to S_A, which is more than that brought by another strategy shown in 5(b). After the decisions made similarly by the other debatable nodes B and D, the final clusters are formed as shown in Figure 6.



(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 $C(C)$, $\Delta|K(C(C))| + \Delta|K(C(H))| = 1$

Figure 5. Membership clarification: possible cluster formations caused by node A's different choices

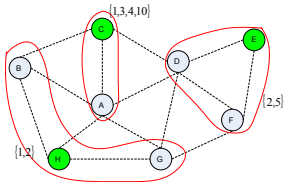
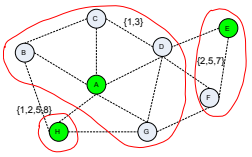


Figure 6. Final formation of clusters. Common channels are shown beside corresponding clusters.

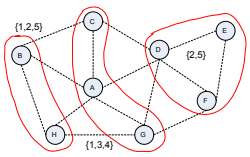
6. PERFORMANCE EVALUATION

The schemes involved in the simulation are as follows,

- ROSS without size control: ROSS-DGA, ROSS-DFA.
- ROSS with size control, i.e., ROSS-δ-DGA and ROSS-δ-DFA where δ is the desired cluster size. In the following, we refer to the above mentioned four schemes as the variants of ROSS.
- SOC [**?**], a distributed clustering scheme pursuing cluster robustness.
- Centralized robust clustering scheme. As shown in Section 4, the centralized robust clustering scheme is formulated as an integer linear optimization problem and is solved by MATLAB with the function $bintprog$.



(a) Generated by SOC



(b) Generated by the centralized clustering scheme

Figure 7. Final clusters formed by the centralized clustering scheme and SOC.

As to the CRN shown in Figure 2, the resulting clusters by the centralized scheme and SOC are shown in Figure 7. We now investigate the performances of the schemes in the following aspects.

- **The average number of CCs per non-singleton cluster.** Non-singleton cluster refers to the cluster whose cluster size is larger than one. Previous work [**?**] and [**?**] claim that the larger average number of CCs over all the clusters indicates robustness , from which we see two flaws. First, the unclustered CR nodes (synonym of singleton clusters) should not be considered when calculating the average number of CCs, as singleton clusters don't contribute to the collaborative computing or sensing. Second, the average number of CCs doesn't necessarily indicate the robustness of individual clusters, because the ability for a cluster to sustain

Trans. Emerging Tel. Tech. 2017; **00**:1–15 © 2017 John Wiley & Sons, Ltd.
DOI: 10.1002/ett
Prepared using ettauth.cls

also depends on cluster size and the locations of the cluster members, but these information can not be illustrated in the average number of CCs. In the performance evaluation, we will examine the metric of average number of CCs per non-singleton cluster, which excludes the bias brought in by the unclustered CR nodes. Moreover, we will examine whether this metric reflects the robustness of the clusters.

- **Robustness of the clusters against newly added PUs.** Robustness is illustrated by the number of the unclustered CR nodes in the CRN, after the CRN being challenged by the increasing number of PUs. This metric indicates the robustness of the clusters, i.e., as to the clusters formed for a given CRN and spectrum availability, how many CR nodes can still be benefited from the clusters when the spectrum availability decreases.
- **Cluster sizes.** We investigate the distribution of CRs residing in the formed clusters with different sizes.
- **Amount of control messages involved.** We investigate the number of control messages involved in the clustering process.
- **Influence from inaccurate spectrum sensing.** The above simulations are conducted under the assumption of perfect spectrum sensing. As spectrum sensing in practise is subject to errors, we are interested to see the performance of the distributed schemes when the spectrum sensing is not perfect. The false negative in spectrum sensing, which misdetects the presence of the active primary users, is harmful to the primary users and should be avoided as much as possible. On the contrary, a false positive i.e., which reports the presence of active primary users when there are actually none, only decreases the available spectrum for the secondary users. In this regard, we assume only the false negative exists in the spectrum sensing.

  We assume when a secondary user is within the transmission range of an active primary user, the probability that it misdetects and thus regards a channel as being available equals to the rate of false negative. The secondary users make clustering decisions based on the imperfect spectrum sensing. After the clustering process is completed, we correct the spectrum availability with the ground truth. Then certain formed clusters may be affected as their CCs which are obtained due to false negative will be revoked.

The simulation consists of two parts, first we investigate the performance of centralized scheme and the distributed schemes in a small network, as there is no polynomial time solution available to solve the centralized problem. In the second part, we investigate the performance of the proposed distributed schemes in the CRN with different scales and densities. The following simulation setting is the same for both simulation parts. CRs and PUs are deployed on a two-dimensional Euclidean plane. The number of licensed channels is 10, each PU is operating on each channel with probability of 50%. The other parameters i.e., the number of CR and PU, and their transmission ranges are given in the beginning of the respective simulation sections. The constant t which is used to control cluster size for ROSS (discussed in Section 5.1.3) is 1.3. CR users are assumed to be able to sense the existence of primary users and identify available channels. All primary and CR users are assumed to be static during the process of clustering. The simulation is written in C++, and the performance results are averaged over 50 randomly generated topologies, and the confidence interval corresponds to 95% confidence level.

6.1. Centralized Schemes vs. Decentralized Schemes

In this part of simulation, there are 10 primary users and 20 CR users dropped randomly (with uniform distribution) in a square area where side length is A. A is a positive value and the transmission ranges of both primary and CR users are set to $A/3$. By doing this, we try to abstract from the influence of any given physical layer technology, and A can be given a concrete value in practice when the physical layer technology is decided. When clustering scheme is executed, around 7 channels are available on each CR node. The desired cluster size δ is 3. As for the centralized scheme, the parameters used in the *punishment* for choosing the clusters with undesired sizes are set as follows, $\rho_1 = 0.4$, $\rho_2 = 0.6$.

6.1.1. CCs in Non-singleton Clusters

Figure 8 shows the centralized schemes outperform the distributed schemes. SOC achieves the most CCs among the distributed schemes, because SOC groups the neighboring CRs which share the most abundant spectrum together, without considering the number of them. As a result, SOC also generates the most unclustered CRs. As to the variants of ROSS, we notice that the greedy mechanism increases CCs in non-singleton clusters significantly.

6.1.2. Cluster Size

Figure 9 depicts the empirical cumulative distribution of the CRs in clusters of different sizes. The centralized schemes don't produce unclustered CR nodes in the simulation. The unclustered nodes generated by ROSS-DGA/DFA account for 3% of the total CR nodes, as comparison, 10% of nodes are unclustered when applying SOC. ROSS-DGA and ROSS-DFA with size control feature generate 5%-8% unclustered CR nodes, which is due to the cluster pruning procedure (discussed in Section 5.1.2 and Section 5.1.3).

In terms of cluster size, the clusters resulted from centralized schemes and ROSS with cluster size control mechanism have little deviation from the desired cluster size. 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

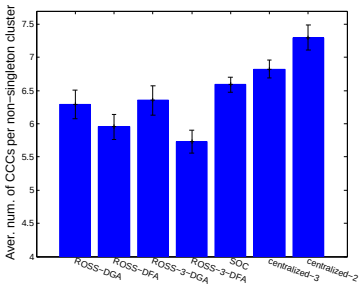


Figure 8. Average number of CCs of non-singleton clusters

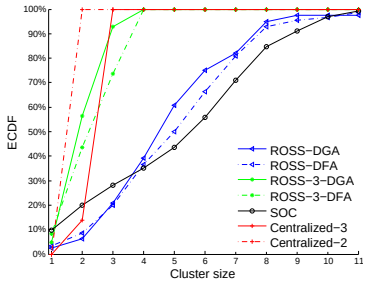


Figure 9. Cumulative distribution of CRs residing in clusters with different sizes

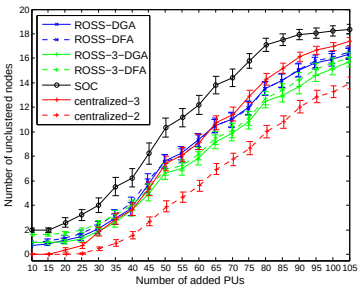


Figure 10. Number of unclustered CRs with decreasing spectrum availability

SOC are 4.5, 5, and 5.5, and the 90% percentiles for the three schemes are 8, 8, and 9, the corresponding sizes resulted by ROSS are closer to the desired size.

6.1.3. Robustness of the formed clusters

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

Considering the number of secondary users residing in a cluster, ROSS based schemes benefit 5%, 30% and 230% more secondary users from neighborhood cooperation than SOC, when the numbers of newly added PR are 10, 40 and 80 respectively. The above observation also shows the average number of CCs of non-singleton clusters doesn't necessarily illustrate the robustness of cluster, i.e., SOC obtains the most CCs among the distributed schemes, but the resulted clusters are vulnerable.

6.1.4. Control Signaling Overhead

In this section we compare the overhead of signaling involved in different clustering schemes. We count the number of *transmissions of control messages* as message complexity [**?**], without distinguishing broadcast or unicast control messages. In Section 5, this metric is synonymous with the *the number of updates*.

As to ROSS, in the first phase the maximal number of broadcast is N according to 5.1. The upper bounds for the transmissions are n^2m and n for ROSS-DGA and ROSS-DFA respectively. Scheme SOC consists of three rounds, and in each round every node needs to broadcast to do comparisons and cluster mergers. The centralized scheme is conducted at the centralized control device, which involves information aggregation and clustering decision dissemination. To analyze the centralized scheme's message complexity, we adopt the backbone structure proposed in [**?**], and apply ROSS to generate cluster heads which serve as the backbone. In the process of information aggregation, all the nodes transmit information to the cluster heads which forward the messages to the controller. In the process of dissemination, all the cluster heads and the debatable nodes broadcast the clustering result, thus the upper bound for the number of broadcast is $N + m + n$.

The number of control messages which are involved in ROSS variants and the centralized scheme is related with the number of debatable nodes. Figure 11 shows the percentage of debatable nodes with different network densities. Table II shows message complexity, quantitative

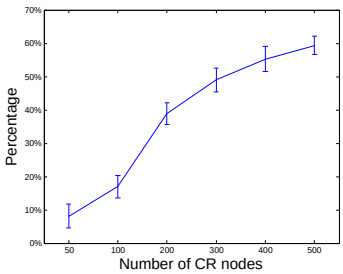


Figure 11. The percentage of debatable nodes after phase I of ROSS.

amount of control messages, and size of control messages. Figure 12 shows the analytical result of the amount of transmissions involved in different schemes.

6.2. Comparison among the Distributed Schemes

In this section we investigate the performances of the proposed distributed clustering schemes with different network scales and densities. The transmission range of

Trans. Emerging Tel. Tech. 2017; **00**:1–15 © 2017 John Wiley & Sons, Ltd.
DOI: 10.1002/ett
Prepared using ettauth.cls

Table II. Signalling overhead

Scheme	Message Complexity	Quantitative number of messages	Content and size of the message
ROSS-DGA, ROSS-δ-DGA	$\mathcal{O}(N^3)$ (worst case)	$N + n^2m$ (upper bound)	PhaseI: ID, d_i,g_i, which are 3 bytes; PhaseII: Cluster head i broadcasts channel availability to all members, where are $\|C(i)\|\|\mathcal{K}\|$ bytes
ROSS-DFA, ROSS-δ-DFA	$\mathcal{O}(N)$ (worst case)	$N + n$ (upper bound)	
SOC	$\mathcal{O}(N)$	$3N$	Every CR node i broadcasts channel availability on all cluster members, which is $\|C(i)\|\|\mathcal{K}\|$ bytes
Centralized	$\mathcal{O}(N)$	$N + n + m$ (upper bound)	clustering result, which is $2N$ bytes [a]

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

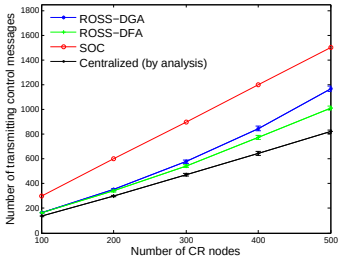


Figure 12. Quantitative amount of control messages.

CR is $A/5$, PU's transmission range is $2A/5$. The initial number of PU is 30. The desired sizes adopted are listed in the Table III, which is about 60% of the average number of neighbors.

Table III

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

6.2.1. Number of CCs per Non-singleton Clusters

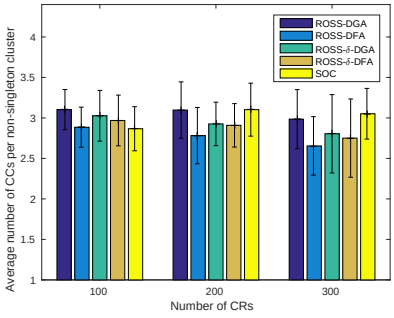


Figure 13. Average number of CCs of non-singleton clusters

The average number of CCs of the non-singleton clusters is shown in Figure 13. Here we don't see the obvious difference between different schemes on the number of CCs.

6.2.2. Robustness of the Formed Clusters

Here we see how robust clusters are when exposed to the increasing influence of PUs. We increase primary users' activity by importing 20 batches of PUs sequentially in CRN, where each batch includes 10 PUs. Figure 14 and 15 show when $N = 100$ and 200, with the same intensity of PUs' activities, more unclustered CR nodes are resulted from SOC than the variants of ROSS, When $N = 300$ as shown in Figure 16 and new PUs are not many, ROSS-DGA/DFA generate slightly more unclustered CR nodes than SOC, but SOC's performance deteriorates quickly when the number of PUs continue increasing. From Figure 14 to 16, we can see that significantly less unclustered CR nodes are generated by ROSS with size control mechanism. Besides, the greedy mechanism moderately strengthens the robustness of the clusters.

6.2.3. Cluster Size Control

Figure 20 shows when network density increases, i.e., N changes from 100 to 300 in the same area, the number of clusters resulted from SOC increases linearly, whereas that by ROSS increases by a smaller margin. This result coincides with the analysis in Section 5.1.3. To see the distribution of the sizes of formed clusters, for each network density, we depict empirical cumulative distribution of CR nodes which are in clusters with different sizes in Figures 17 18 19 respectively.

The cluster sizes resulted from the variants of ROSS are in proximity to the desired size, i.e., as shown in Figures 17, 90% of CR nodes are in the clusters whose sizes are between 3 and 9, while for SOC, only 17% of nodes are in the clusters with these sizes. Similarly, when $N = 200$ and desired size is 12 as shown in Figure 18, 80% of nodes are in the clusters whose sizes are between 6 and 18, meanwhile only 30% of nodes are in the clusters with these sizes when SOC is executed. The clusters sizes from ROSS-δ-DGA and ROSS-δ-DFA concentrates more around the desired size than that from ROSS-DGA and ROSS-DFA. In contrary, the clusters from SOC demonstrates obvious divergence on cluster sizes.

The limitation of distributed scheme ROSS is it doesn't generate clusters whose sizes exceed the cluster head's neighborhood. The reason is with ROSS, cluster heads

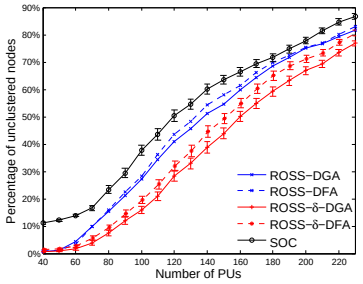


Figure 14. 100 CRs

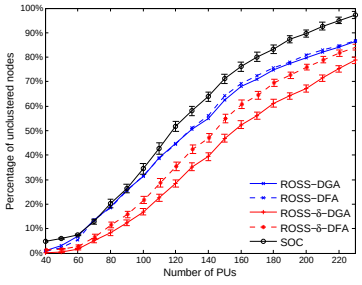


Figure 15. 200 CRs

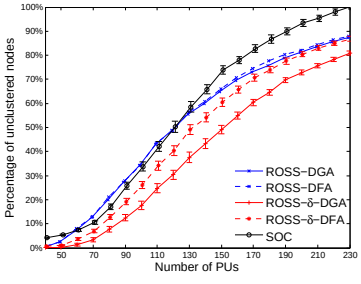


Figure 16. 300 CRs

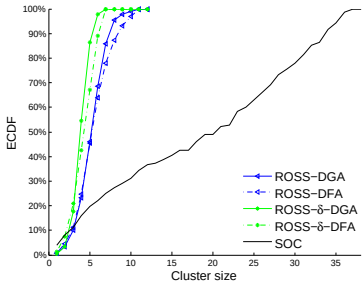


Figure 17. 100 CRs, 30 PUs in network

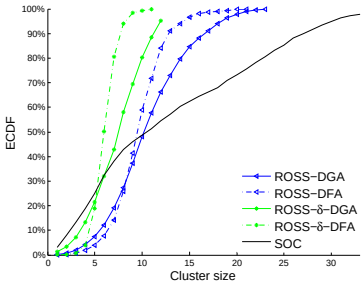


Figure 18. 200 CRs, 30 PUs in network

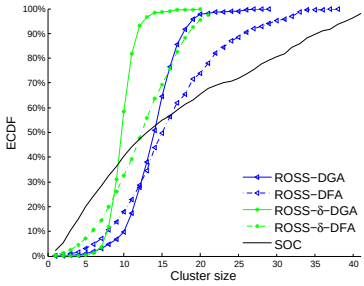


Figure 19. 300 CRs, 30 PUs in network

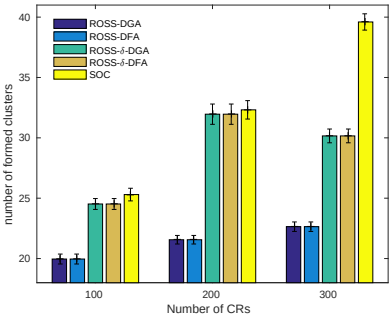


Figure 20. The number of formed clusters.

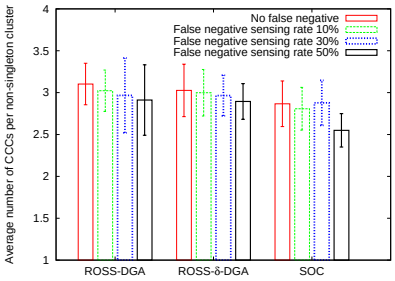


Figure 21. The number of CCs per non-singleton cluster with the presence of spectrum sensing false negative

form clusters on the basis of their neighborhood, thus don't involve the nodes out of the neighborhood.

6.2.4. The Performance with False Negative in Spectrum Sensing

Figure 21 shows the average number of CCs decrease slightly when the false negative rate increases. The size distribution of the ROSS-DGA, ROSS-δ-DGA and SOC is shown in Figure 22. For all the schemes, when false negatives increase, the number of singleton clusters and smaller clusters increases accordingly. The clusters resulted from SOC are affected by the sensing errors heavily. More unclustered nodes are generated, and a lot of small clusters are formed, e.g., when false negative rate is 30%. In contrary, ROSS variants are resilient in terms of unclusterd nodes and cluster sizes. We can conclude that due to the negotiation within neighborhoods, ROSS variants successfully rules out the channels obtained due to false negative sensing.

6.3. Insights Obtained from the Simulation

The simulation with large CRN network confirms the conclusion drawn from the small CRN network. First, the average number of CCs per cluster, which is adopted as metric for cluster robustness, is not able to tell the robustness against the increasing influence of the primary users. Second, the centralized clustering scheme forms the clusters which satisfy the requirement on cluster size, and the resulted clusters are robust against PUs' increasing activity, besides, it involves the smallest control overhead in the process of clustering. Third, as distributed schemes,

Trans. Emerging Tel. Tech. 2017; **00**:1–15 © 2017 John Wiley & Sons, Ltd.
DOI: 10.1002/ett
Prepared using ettauth.cls

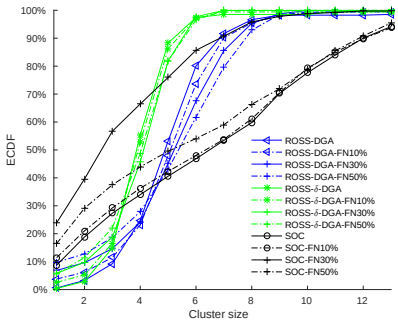


Figure 22. 100 CRs with false negative in spectrum sensing, 30 PUs in network

the variants of ROSS outperform SOC considerably in the following four aspects.

- The variants of ROSS generate less unclustered nodes than SOC for a given CRN, and the resulted clusters are more robust than SOC when PUs become more active.
- The amount of signaling overhead involved in ROSS is about half of that needed for SOC, and the signaling messages are much shorter that the latter.
- Compared with SOC, the clusters generated by ROSS don't appear with a wide span of sizes. ROSS with size control mechanism result in the clusters whose sizes are in proximity to the desired size.
- The variants of ROSS are more resilient against the erroneous spectrum sensing.

Moreover, the ROSS variants with size control features achieve similar performance to the centralized scheme in terms of cluster size, and the cluster robustness is similar when applying the variants of ROSS and the centralized scheme respectively. Among 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 increased signaling overhead.

7. CONCLUSION

In this paper we investigate robust clustering problem in CRN thoroughly and propose both centralized and distributed clustering solutions. We give mathematical description of the problem and prove NP hardness of it. The proposed centralized scheme generate clusters which have long life expectancy against primary users, and the cluster sizes are close to the desired cluster size. Through simulation, the distributed schemes demonstrate similar performance with the centralized scheme in terms of cluster robustness, signaling overhead and cluster sizes. The distributed scheme outperforms the comparison distributed scheme by generating more robust clusters, generating clusters whose sizes are in a smaller range, and being more resilient against the erroneous spectrum sensing.

Appendices

A. PEUDO CODE FOR ALGORITHM 1, 2, 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 $\nexists j \in Nb(i) \setminus \Lambda$, *such that* $d_i \geq d_j$ **then**
  return 1;
end
if $\exists j \in Nb(i) \setminus \Lambda$, *such that* $d_i > d_j$ **then**
  return 0;
else
  if $\nexists j \in Nb(i) \setminus \Lambda$, *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 ID_i *is smaller than any* ID_j, $j \in \tau_2 \setminus i$ **then**
  return 1;
end
return 0;

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

For the sake of contradiction, let us assume there exist a secondary user α which is not included into any cluster. Then there exists a node $\beta \in \text{Nb}(\alpha)$ such that

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* $i \in \tau_1$, $i = \arg\max(|\cap_{j \in C \setminus i} K_j| - |\cap_{j \in C} K_j|)$ **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**

$d_\alpha > d_\beta$ (otherwise α becomes cluster head). In this case, according to Algorithm 1, β is not included in any clusters, because otherwise $d_\beta = M$, a large positive integer, which contradicts to $d_\alpha > d_\beta$. 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 a neighboring node γ with $d_\gamma < d_\beta$. So far, 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 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 23.

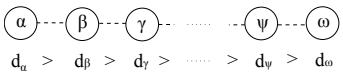


Figure 23. The node series discussed in the proof of Theorem 5.1, the deduction begins from node α

In this example, node ω is at the tail of 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 common channels), including the nodes which precede ω in the list. The nodes which join a cluster set their individual connection degrees

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.

1 **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**
2 **if** $\exists$ *only one* $C \in S_i$, $C = \arg\min(|K(C \setminus i)| - |K(C)|)$ **then**
3 return C;
4 **else**
5 $\exists$ multiple $C \in S_i$ which satisfies $C = \arg\min(|K(C \setminus i)| - |K(C)|)$;
6 $\tau_1 \leftarrow C$;
7 **end**
8 **if** $\exists$ *only one* $C \in \tau_1$, $C = \arg\max(K_{h(C)} \cap K_i)$ **then**
9 return C;
10 **else**
11 $\exists$ multiple $C \in S_i$ which satisfies $C = \arg\max(K_{h(C)} \cap K_i)$;
12 $\tau_2 \leftarrow C$;
13 **end**
14 **if** $\exists$ *only one* $C \in \tau_2$, $C = \arg\min |C|$ **then**
15 return C;
16 **else**
17 return $\arg\min_{C \in \tau_2} h(C)$;
18 **end**
19 **end**

to M, which makes the node immediately precede in the list to become a cluster head. 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.

□

C. PROOF OF THEOREM 3.1

Proof
To prove 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 the robust clustering problem to show the latter is at least as hard as the former. 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,

Definition 2. *Given a finite set $\mathcal{G}$ of non-negative integers where $\mathcal{G} \subsetneq \mathbb{N}$, and a collection of sets $\mathcal{Q} =$*

Trans. Emerging Tel. Tech. 2017; **00**:1–15 © 2017 John Wiley & Sons, Ltd.
DOI: 10.1002/ett
Prepared using ettauth.cls

$\{S_1, S_2, \cdots, S_m\}$ *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, *i.e.,* $\sum_{\forall S \in \mathcal{I}} \omega(S) > \lambda$.

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, i.e., a collection of sets $\mathcal{Q} = \{S_1, S_2, \cdots, S_m\}$, where the set $S_i, i \in \{1, 2, \ldots, m\}$ consists of positive integers. There is an integer weight $\omega(S_i)$ for S_i, 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. W.l.o.g. we let set $\cup_{i \in \{1,2,\ldots,m\}} S_i = \{1, 2, \ldots, N\} = \mathcal{P}$.

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 common channels 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 which is exclusive for this cluster. For the set S which contains only one element, i.e., $S = \{t\}$ where $t \in \mathcal{P}$, a cluster composed with two CR nodes will be created. In this case, one CR node's ID is t, the other CR node is the dummy node of the former and its ID is $t + N$. A number of $|\omega(S)|$ channels from the exclusive spectrum band for this cluster are allocated to these two CR nodes. Now we have constructed the clusters which correspond to all the sets in $\mathcal{Q}$. Note that every CR node is allowed to form a singleton cluster by itself, although its common channels don't contribute to the sum of $f(C)$.

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 in each constructed cluster, the CR nodes occupy the common channels which are exclusive to this cluster, this design of transformation eliminates the formation of the cluster which doesn't have a corresponding set in $\mathcal{Q}$. 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.

Now suppose there is a set of pairwise disjoint clusters which constitute the CRN $\mathcal{N}$, and the sum of $f(C)$ is greater than λ. After removing the singleton clusters, we can easily find the natural association between the remaining clusters and the sets in $\mathcal{Q}$. The clusters in the CRN correspond to the sets in $\mathcal{Q}$ according to the mapping between the node IDs in the clusters and the elements in the sets. In particular, the clusters which contain dummy CR nodes correspond to the sets which contain only one element. Then the sum of the weights of the corresponding sets equals to the sum of $f(C)$ and thus greater than λ.

We have now shown that our algorithm solves the weighted k-set packing problem using a black box for the robust clustering problem. Since our construction takes polynomial time, we can conclude that the robust clustering problem is NP-hard.

□