TRANSACTIONS ON EMERGING TELECOMMUNICATIONS TECHNOLOGIES
*Trans. Emerging Tel. Tech.* 2017; **00**:1–17
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 [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, 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 [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 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 [9], while also been known for reducing the interference between cognitive clusters [10], and improving routing [8].

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 being 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.



*Prepared using ettauth.cls [Version: 2012/06/19 v2.10]*

if only one or two nodes are members of a cluster [11]. 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 [12, 13].

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 [14] and [15]. 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) [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 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 [14]. 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 first review briefly 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 common channel within each cluster is the foremost question to answer. [16, 17, 18] propose different clustering schemes and enforce that every cluster possesses at least one CC. The clustering scheme in [11] looks for a network partition which improves the accuracy of spectrum sensing without considering robustness. In [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. [12] proposes a cluster structure which imprroves energy efficiency. Furthermore, [20] 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 [21]. However, none of the above mentioned schemes provide robustness of the clusters against random primary user activity.

The authors of [22] 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. [23] 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 [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%-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 [12, 20]. In [14] 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, 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 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


*Trans. Emerging Tel. Tech.* 2017; **00**:1–17 
DOI: 10.1002/ett
*Prepared using ettauth.cls*

licensed channels.[†] We adopt the unit disk model [24] for both primary and CR users' transmission. Thus, if a CR node $i$ locates within the transmission range of an active primary user $p$, $i$ is not allowed to use the channel which is being used by $p$. We assume the primary users 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 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 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 $\mathrm{Nb}(i)$.

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. $K(C) = \bigcap_{i \in C} K_i$. Table I summarizes all parameters and their assumed relevance in our system model.

**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 |
| $\delta$ | 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 |
| $\mathcal{S}$ | 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 $C_i$ |
| $M$ | the cardinality of set $\mathcal{S}$ which contains all the possible clusters (Sec. 4) |
| $p(C_i)$ | the weight with respect to cluster $C_i$ (Sec. 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 (Sec. 5) |
| $e$ | cost function in the formulated game (Sec. 5.2.2) |

### 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 primary users. 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 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 trade-off 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:*

*1. The intersection of any two 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 $f(C) = K(C)$, i.e. it is given as the number of jointly available CCs for all nodes of that cluster. The desired size $\delta$ is within $[\delta_1, \delta_2]$, which is pre-decided based on the capability of the CR users and the tasks to be conveyed.*

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

4. *When the cluster size is larger or smaller than the range* $[\delta_1, \delta_2]$*,* $f(C)$ *is defined as 0, i.e. 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 above definition, we distinguish between the real number of CCs per cluster, which is given as $K(C)$, and the contribution of each formed cluster towards the objective function, which is given by function $f(C)$. If the cluster size is within range, the two parameters are the same, but otherwise the distinction is enforce 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 $\lambda$ 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

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, i.e., the locations of primary users and their working channels, and the 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 which 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 $\mathcal{S} = \{C_1, C_2, \ldots, C_M\}$ [§]. Then the problem as defined in Definition 1 can be formulated as a binary linear programming problem and can be solved by many available solvers:

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

$x_i, i \in \{1, 2, \cdots M\}$ is a set of binary optimization variables. Being either 1 or 0, $x_i$ denotes whether the $i$-th cluster $C_i$ in $\mathcal{S}$ is chosen or not. $e_{ij}, i \in \{1, 2, \cdots M\}$ and $j = \{1, \ldots, N\}$ is a set of constants which indicate whether the CR node $j$ resides in the cluster $C_i$, i.e., $e_{ij} = 1$ means node $j$ resides in the cluster $C_i$, $e_{ij} = 0$ indicates $j$ doesn't belong to that cluster .

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

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 two implications. First, the sum being larger than zero indicates that every node should be involved. Second, the sum equals 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 which constitute the CRN. In the second term in the bracket, we propose $p(C_i)$ which is a size-related weight for cluster $C_i, i \in 1, \cdots, M$. $p(C_i)$ is positively related with the difference between $C_i$'s size and the desired size $\delta$. When $x_i$ is 1 ($C_i$ is chosen) but $|C_i|$ doesn't equal to the desired cluster size $\delta$, the second item is negative, which contradicts the direction of the optimization. Thus the second item discourages the appearance the clusters whose sizes deviate from the desired cluster size $\delta$. The weight $p(C_i)$ with respect to different cluster sizes is as follows,

$$p(C_i) = \begin{cases} 0 & \text{if } |C_i| = \delta \\ \rho(1) & \text{if } |C_i| = \delta \pm 1 \\ \rho(2) & \text{if } |C_i| = \delta \pm 2 \\ \vdots & \\ \rho(\sigma) & \text{if } |C_i| = \delta \pm \sigma \end{cases}$$

where $\rho$ is an increasing function and there is $0 < \rho(1) < \rho(2) < \cdots < \rho_(\sigma)$. Note that here we adopt the desired cluster size in stead of the range for cluster sizes which is described in Definition 3.1. Because in implementation, we usually need a large range to guarantee a feasible result for the optimization problem. By adopting the desired size and the cost for deviating from it, we actually set a large range of cluster sizes and meanwhile a better control of the resulted sizes. Also to guarantee the feasibility of the problem, we allow the singleton cluster to contribute the objective function.

The difficulty of using this method lies in obtaining the set $\mathcal{S}$. In the worst case, i.e., if every CR node can communicate directly with any other node, then the CRN forms a full 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 two 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.


DOI: 10.1002/ett
*Prepared using ettauth.cls*

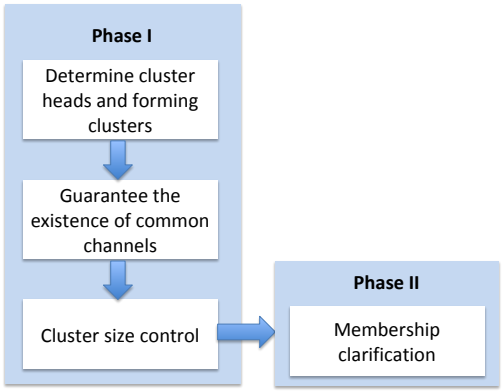



**Figure 1.** Processing steps of ROSS

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 non-overlapping 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 spectrum sensing and neighborhood discovery have been completed. Furthermore, neighboring nodes have exchanged already their channel availabilities via the dedicated control channel. As a 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* $d_i$: $d_i = \sum_{j \in \text{Nb}(i)} |K_i \cap K_j|$. $d_i$ 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* common channels among all neighbors of $i$.
- *Neighborhood connectivity degree* $g_i$: In contrast, $g_i$ is the number of CCs which are *jointly* available for $i$ and all of its neighbors, thus $g_i = |\bigcap_{j \in \text{Nb}(i) \cup i} K_j|$. 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 one end node is within the other's transmission range, while the number along the dashed line is the number of common channels between the two end nodes. In this example, the sets of available channels on the nodes are: $K_A = \{1, 2, 3, 4, 5, 6, 10\}$, $K_B = \{1, 2, 3, 5, 7\}$, $K_C = \{1, 3, 4, 10\}$, $K_D = \{1, 2, 3, 5\}$, $K_E = \{2, 3, 5, 7\}$, $K_F = \{2, 4, 5, 6, 7\}$, $K_G = \{1, 2, 3, 4, 8\}$, $K_H = \{1, 2, 5, 8\}$. Figure 2 shows in particular the resulting connectivity vector per node.

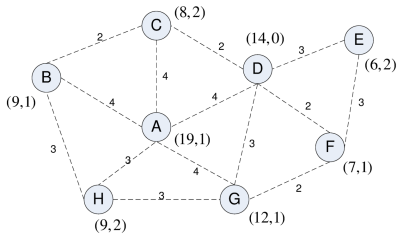



**Figure 2.** Illustration of the resulting connectivity vector $(d_i, g_i)$ for each node of an example CRN.

### 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 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, which has the same individual connectivity degree as $i$, i.e., $d_j = d_i$ 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, 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 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, in particular the new/updated sets of available channels regarding itself and all its cluster members. Pseudo code regarding this process, i.e. the cluster head decision and the initial cluster formation, is provided as Algorithm 1 in the appendix.

After a CR node, say $i$, receives notification that there is a new cluster head in its neighborhood, $i$ sets its individual

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

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, i.e., $i$ 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 Common Channels

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 common channels between them 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 elimination results in more common channels 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 pseudo code for this procedure is given as 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, see also Theorem 5.1.

### 5.1.3. Cluster Size Control in Dense CRN

Both analysis and simulation [27] 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 $\delta$, 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, e.g., $t = 1$, the formed cluster in phase I is $\delta$. For a cluster which has members included by other clusters, the size of that cluster will be smaller than $\delta$ after the membership clarification phase. If $t$ is chosen large, e.g., $t \cdot \delta$ equals 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.

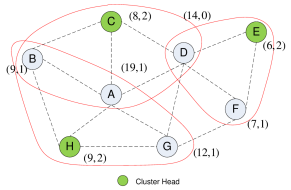



**Figure 3.** Cluster formation after phase I of ROSS. Nodes $A, B, D$ are debatable nodes as they belong to multiple clusters.

We have the following lemma to show every secondary user will eventually be either integrated into a cluster or become 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 $J$. 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 $i$ doesn't become cluster member and so its *individual connectivity degree* 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 follows from Lemma 5.2 considering that $N$ is the number of all the secondary users in the CRN. Based on the above lemmas, we have:

**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,


*Trans. Emerging Tel. Tech.* 2017; **00**:1–17 © 2017 John Wiley & Sons, Ltd.
DOI: 10.1002/ett
*Prepared using ettauth.cls*

Theorem 5.1 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. 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

After running phase I of ROSS, we notice that nodes $A, B, D$ 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 which 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 $\mathcal{S}_i$ if it chooses a claiming cluster and leaves the other claiming clusters. In case of 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 by Algorithm 3. After deciding its membership, debatable node $i$ notifies all its claiming 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 an 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 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 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 [28]. Over the last decade, game theory has been extensively applied to problems in communication and networking [29, 30]. Congestion game is an interesting game model which describes the problem where participants compete for limited resources in a non-cooperative manner. It has the good property that a 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 server selection in distributed computing platforms [31], or users downloading files from cloud, etc.

To formulate the debatable nodes' membership clarification into a congestion game, we reexamine this process from a different perspective. Thus, debatable nodes are not included in any cluster and they need 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.

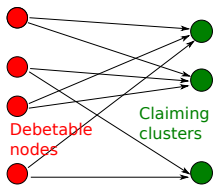



**Figure 4.** Illustration of debatable nodes and claiming clusters

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 $\Gamma = (\mathcal{P}, \mathcal{R}, \sum_{i, i \in \mathcal{P}}, e)$ 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, only a single resource will be allocated to $i$.
- The cost function $e(C)$ regarding resource $C$. $e(C) = \Delta|K^i(C)|, C \in S_i$, 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 $C$.

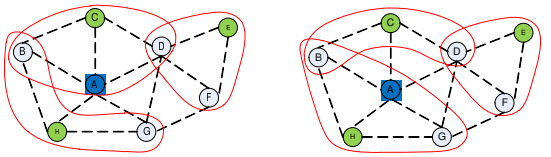
When the utility function is decided purely by the amount of players accessing the resource, the game is a canonical congestion game [32]. In our game, as the channel availability on debatable nodes (players) is different, the loss of CCs (cost) caused by a debatable node could also be different. Hence, this congestion game is player specific [32]. 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 equilibrium which can be reached with the best response update, while the upper bound for the number of steps before convergence is $n^2 * m$ [32], 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 cluster heads is dependent on the network density and the CR node's transmission range, as mentioned in Section 5.1. The simulation in [33] shows that the cluster heads account for from 3.4% to 20% of the total CR nodes with increasing network density. Furthermore, as the game is played locally and in parallel i.e., a debatable node can only interact with a few claiming clusters, the execution speed is significantly reduced.

### 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, 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 two possible strategies of node $A$ are illustrated in Figure 5. In Figure 5(a), 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 5(b). After similar decisions are made 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

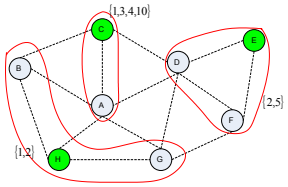



**Figure 6.** Final formation of clusters. Common channels are shown as well as the corresponding clusters.

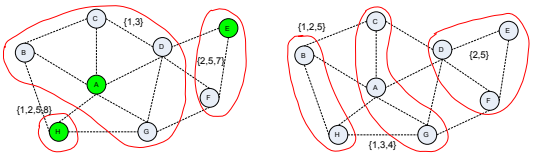
## 6. PERFORMANCE EVALUATION

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 two 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, while 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. We therefore 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, i.e., ROSS-$\delta$-DGA and ROSS-$\delta$-DFA where $\delta$ is the desired cluster size;
- SOC [15], a distributed clustering scheme pursuing cluster robustness;


*Trans. Emerging Tel. Tech.* 2017; **00**:1–17 
DOI: 10.1002/ett
*Prepared using ettauth.cls*




(a) Generated by SOC

(b) Generated by the centralized clustering scheme

**Figure 7.** Final clusters formed by SOC as well as the centralized clustering scheme.

- 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 non-singleton cluster.** Previous work [15] and [14] claim that the larger average number of CCs over all the clusters indicates robustness. As mentioned, this interpretation has several shortcomings: First, 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 a cluster, because the ability of a cluster to sustain primary user 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 primary user activity. We thus are interested in this effect as a second measure for robustness. In particular, we are interested in the number of CR nodes which are still part of a cluster after exposing the clusters to primary user activity.
- **Cluster sizes.** We investigate the distribution of the sizes of the formed clusters. This metric reflects the above mentioned size constraints, i.e. 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 performs under imperfect sensing accuracy. In case of erroneous channel sensing, false negatives harm primary users while 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 primary user activity is only correcly detected by a CR node in transmission range with a certain probability, i.e. 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 two parts: First we investigate the performance of the centralized scheme and the distributed schemes for a small network, as the run-time 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 evalutions: CRs and PUs are deployed on a two-dimensional Euclidean plane. The number of licensed channels is 10, each PU is operating on each channel with probability of 50%. The constant $t$ which is used to control cluster size for ROSS (discussed in Section 5.1.3) is 1.3. CR users are assumed to sense the existence of primary users 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 i.e., 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 a 95% confidence level.

### 6.1. Centralized Scheme vs. Decentralized Schemes

We start with the comparison of the centralized scheme versus various distributed ones. For this, we consider 10 primary users 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 $\delta$ as 3. As for the centralized schemes, we set the following parameters numerically: $\rho_1 = 0.4$, $\rho_2 = 0.6$.

We start with the consideration of the CCs in all non-singleton 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 which 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 (i.e. the ROSS-DGA variants) maximize the CCs in non-singleton clusters significantly.

Figure 9 provides further insights into the performance comparison. Here, we depict the empirical cumulative distribution function of the size of the clusters. The centralized schemes don't result in any singleton clusters in the 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%-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 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, i.e., the 50% percentiles for ROSS-DGA, ROSS-DFA and SOC are 4.5, 5, and 5.5, and the 90% percentiles for the three schemes are 8, 8, and 9. Thus, the corresponding sizes resulting from ROSS are closer to the desired size.

Next, we consider the robustness of clusters if facing random primary user activity. We thus extend the simulation by adding more primary users sequentially into the area of the CRN, leading to a decreasing spectrum availability. While 10 primary users are in the network at start, some extra19 batches of primary users are added sequentially, each batch including 5 primary users that are placed randomly in the area. These added primary users choose then an active channel also at random. Figure 10 shows the corresponding average number of unclustered CR nodes as a result of this significant increase in primary user activity. The figure reveals that the centralized scheme with a desired size of 2 leads to the best robustness, while SOC leads to the worst one. Surprisingly, the centralized scheme with desired size of 3 doesn't 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 which are more likely to be maintained despite the increasing primary user activity.

Alternatively, we can consider the total share of users (still) residing in a cluster after the addition of the primary users 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 non-singleton clusters doesn't necessarily reflect the robustness of clusters, i.e., SOC obtains the most CCs among the distributed schemes, but the resulting clusters are vulnerable to primary user 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 [34], without distinguishing broadcast or uni-cast control messages.

As to ROSS, in the first phase the maximal number of broadcasts is $N$ according to 5.1. 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 three 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 [35], and apply ROSS to generate cluster heads which serve as the backbone. In the stage of information aggregation, all the nodes transmit information to the cluster heads which 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 which 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 II 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 which are involved in the variants of ROSS is still smaller than the 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. We are here 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 III, the desired size equals to 60% of the average number of neighbors. The


*Trans. Emerging Tel. Tech.* 2017; **00**:1–17 
DOI: 10.1002/ett
*Prepared using ettauth.cls*

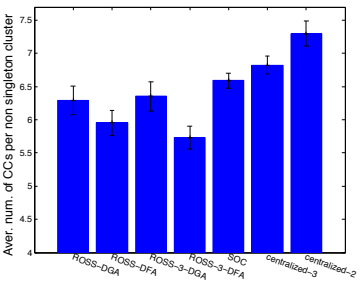



**Figure 8.** Average number of CCs of non-singleton clusters

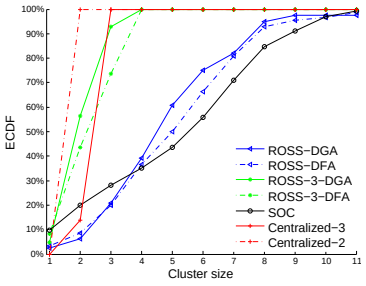



**Figure 9.** Cumulative distribution of CRs residing in clusters with different sizes

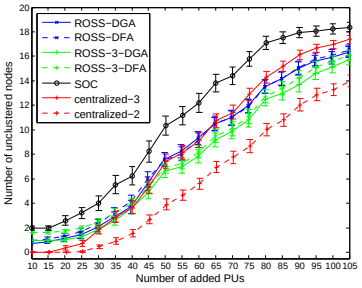



**Figure 10.** Number of unclustered CRs with decreasing spectrum availability

**Table II.** Quantity of control messages

| Scheme | Message Overhead | Number of messages | Content and size of the message |
|---|---|---|---|
| ROSS-DGA, ROSS-$\delta$-DGA | $\mathcal{O}(N^3)$ [a] | $N + n^2m$ (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-$\delta$-DFA | $\mathcal{O}(N)$ [b] | $N + n$ (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 [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 $\{i, C\}, i \in C, i \in \mathcal{N}$.

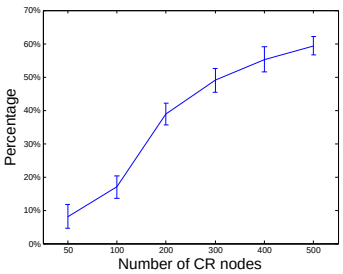



**Figure 11.** Percentage of debatable nodes after ROSS phase I

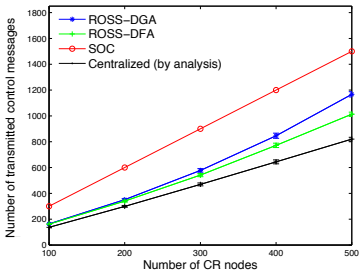



**Figure 12.** The number of control messages required for clustering, which is based on the second column of Table II

transmission range of CR is now set to $A/5$ while the primary user transmission range is set to $2A/5$. The initial number of primary users is set to 30.

**Table III.** The average numbers of neighbors and the chosen desired sizes with respect to different network scales

| Number of CRs | 100 | 200 | 300 |
|---|---|---|---|
| Average num. of neighbors | 9.5 | 20 | 31 |
| Desired size $\delta$ | 6 | 12 | 20 |

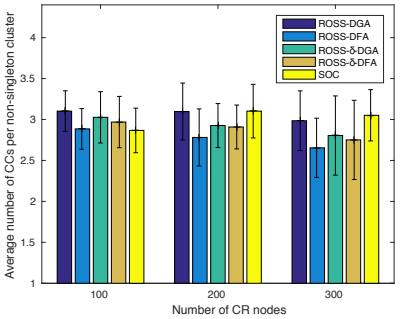



**Figure 13.** Average number of CCs of non-singleton clusters in case of increasing the number of CR nodes.

We start again with considering the average number of CCs over all non-singleton clusters, 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.

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

We next turn to the size of the formed clusters under the different distributed schemes. For this we study in Figure 20 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 sub-linearly in case of the variants of ROSS. This result coincides with the analysis in Section 5.1.3. We furthermore consider the empirical distribution function of the size of the formed clusters, for each considered network density, in Figures 17, 18 and 19 respectively.

The empirical distribution functions (ECDF) associated with the cluster sizes show that the cluster sizes resulting from the variants of ROSS are clearly influenced by the chosen desired size, i.e., as shown in Figures 17, 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, while 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 18), 80% of nodes are in clusters whose sizes are between 6 and 18, while only 30% of nodes are in clusters of similar sizes when SOC is executed. The cluster sizes from ROSS-$\delta$-DGA and ROSS-$\delta$-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. We furthermore 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 formed by SOC are furthermore affected by the sensing errors significantly. More unclustered nodes are generated, and a lot of small clusters are formed, e.g., 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 mathematical description of the problem and prove NP hardness of it. Both centralized and distributed clustering solutions are proposed. With the increasing tensity of the primary users' activity, our proposed schemes generates clusters which 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 resulted 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, although it requires a centralized device and the involved control message packet is large. Our proposed distributed scheme is also more robust against the erroneous spectrum sensing compared with the comparison 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 primary users' activity.


*Trans. Emerging Tel. Tech.* 2017; **00**:1–17 
DOI: 10.1002/ett
*Prepared using ettauth.cls*

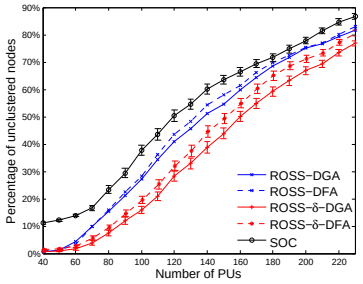



**Figure 14.** Percentage of Unclustered CR nodes when $N = 100$

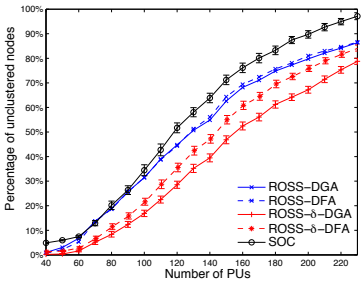



**Figure 15.** Percentage of Unclustered CR nodes when $N = 200$

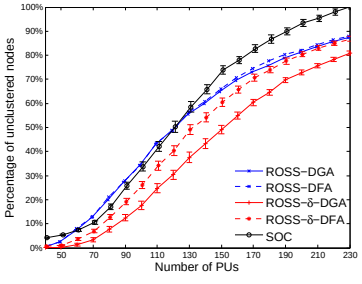



**Figure 16.** Percentage of Unclustered CR nodes when $N = 300$

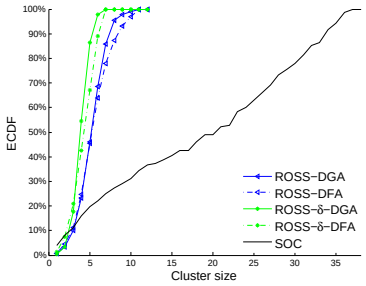



**Figure 17.** Empirical distribution function associated with cluster sizes when there are 100 CRs and 30 PUs in network

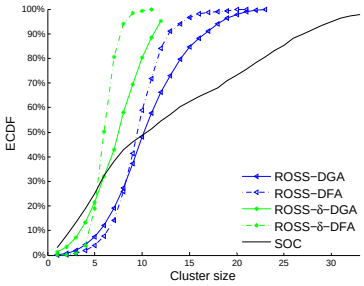



**Figure 18.** Empirical distribution function associated with cluster sizes when there are 200 CRs and 30 PUs in network

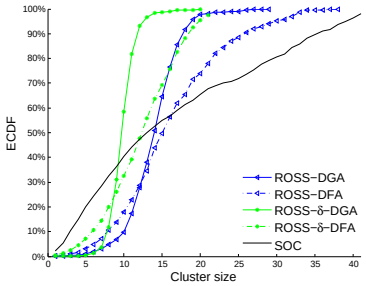



**Figure 19.** Empirical distribution function associated with cluster sizes when there are 300 CRs and 30 PUs in network

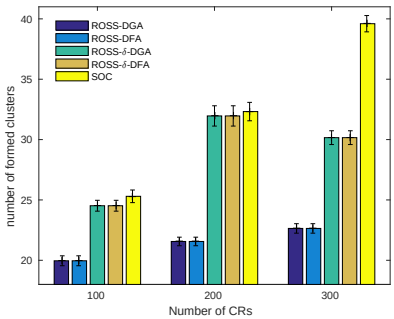



**Figure 20.** The number of formed clusters.

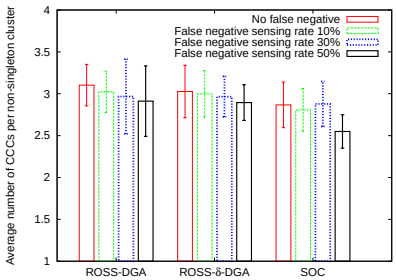



**Figure 21.** The number of CCs per non-singleton cluster with the presence of spectrum sensing false negative

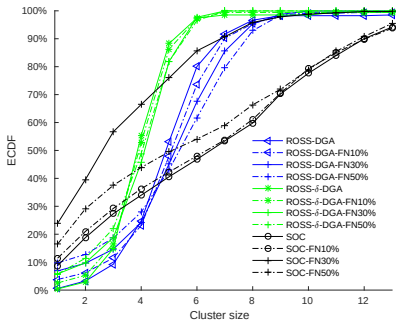



**Figure 22.** Empirical distribution function associated with cluster sizes, where there are 30 PUs and 100 CRs with false negative in spectrum sensing

# Appendices

## A. PEUDO CODE FOR ALG. 1, 2, 3

## B. PROOF OF LEMMA 5.1

*Proof*

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

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

**Input:** $d_j, g_j, j \in \text{Nb}(i) \setminus \Lambda$, $\Lambda$ denotes the set of cluster heads among $\text{Nb}(i)$. Empty sets $\tau_1, \tau_2$.
**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 $d_j == d_i$ **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** $\text{ID}_i$ 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 $\tau_1, \tau_2$
**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 $i \in C \setminus h(C)$, $i = \textbf{arg min}(|K_{h(C)} \cap K_i|)$ **then**
4  $C = C \setminus i$;
5  **else**
6  $\exists$ multiple $i$ which satisfies $i = \textbf{arg min}(|K_{h(C)} \cap K_i|)$;
7  $\tau_1 \leftarrow i$;
8  **end**
9  **if** $\exists$ only one $i \in \tau_1$, $i = \textbf{arg max}(|\cap_{j \in C \setminus i} K_j| - |\cap_{j \in C} K_j|)$ **then**
10  $C = C \setminus i$;
11  **else**
12  $C = C \setminus i$, where $i = \underset{i \in \tau_1}{\textbf{arg min}}\ \text{ID}_i$
13  **end**
14  **end**
15 **end**

degrees and IDs. This assumption is justified as the neighborhood connectivity degrees and node IDs 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 $\alpha$ which is not included into any cluster. Then there exists a node $\beta \in \text{Nb}(\alpha)$ such that $d_\alpha > d_\beta$ (otherwise $\alpha$ becomes cluster head). In this case, according to Algorithm 1, $\beta$ 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 two cases: If $\beta$ becomes cluster head, node $\alpha$ is included, the assumption that $\alpha$ is not included in any cluster is not true. If $\beta$ is not a cluster head, then $\beta$ is not in any cluster, we can repeat the previous analysis made on node $\alpha$, and deduce that node $\beta$ has a neighboring node $\gamma$ with $d_\gamma < d_\beta$. So far, when no cluster head is identified, the unclustered nodes, i.e., $\alpha$, $\beta$ form a linked list, where their individual 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.

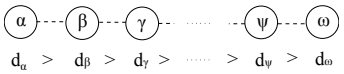



**Figure 23.** The node series discussed in the proof of Theorem 5.1, the deduction begins from node $\alpha$

In this example, node $\omega$ is at the tail of a list. As $\omega$ does not have neighboring nodes with lower individual connectivity degree, $\omega$ becomes a cluster head. Then $\omega$ incorporates all its one-hop neighbors (here we assume that every newly formed cluster has common channels), including the nodes which precede $\omega$ in the list. The nodes which join a cluster set their individual connection degrees to $J$, 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 $\alpha$ 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*,


*Trans. Emerging Tel. Tech.* 2017; **00**:1–17 © 2017 John Wiley & Sons, Ltd.
DOI: 10.1002/ett
*Prepared using ettauth.cls*

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

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

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

which is NP-hard when $k \geqslant 3$ [36], 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 $\mathcal{Q} = \{S_1, S_2, \cdots, S_m\}$ 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 $\lambda$, 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 $\lambda$ 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 $\lambda$. 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 $\lambda$.

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.

□

## REFERENCES


1. J. Mitola and G. Q. Maguire, "Cognitive radio: making software radios more personal," *IEEE Personal Communications*, vol. 6, no. 4, pp. 13–18, Aug 1999.

2. A. Sahai, R. Tandra, S. M. Mishra, and N. Hoven, "Fundamental design tradeoffs in cognitive radio systems," in *Proc. of ACM TAPAS '06*.
3. J. Jacob, B. R. Jose, and J. Mathew, "Cellular automata approach for spectrum sensing in energy efficient sensor network aided cognitive radio," in *Proc. of ICECCS 2012*.
4. I. F. Akyildiz, B. F. Lo, and R. Balakrishnan, "Cooperative spectrum sensing in cognitive radio networks: A survey," *Phys. Commun.*, vol. 4, no. 1, pp. 40–62, Mar. 2011.
5. C. Sun, W. Zhang, and K. B. Letaief, "Cluster-based cooperative spectrum sensing in cognitive radio systems," in *proc. of IEEE ICC 2007*.
6. V. Kawadia and P. R. Kumar, "Power control and clustering in ad hoc networks," in *Proc. of INFOCOM '03*, 2003, pp. 459–469.
7. M. Krebs, A. Stein, and M. A. Lora, "Topology stability-based clustering for wireless mesh networks," in *IEEE GLOBECOM 2010*.
8. A. A. Abbasi and M. Younis, "A survey on clustering algorithms for wireless sensor networks," *Comput. Commun.*, vol. 30, no. 14-15, pp. 2826–2841, 2007.
9. D. Willkomm, M. Bohge, D. Hollós, J. Gross, and A. Wolisz, "Double hopping: A new approach for dynamic frequency hopping in cognitive radio networks," in *Proc. of PIMRC 2008*.
10. C. Passiatore and P. Camarda, "A centralized inter-network resource sharing (CIRS) scheme in IEEE 802.22 cognitive networks," in *Proc. of IFIP Annual Mediterranean Ad Hoc Networking Workshop 2011*.
11. Q. Wu, G. Ding, J. Wang, X. Li, and Y. Huang, "Consensus-based decentralized clustering for cooperative spectrum sensing in cognitive radio networks," *Chinese Science Bulletin*, vol. 57, 2012.
12. H. D. R. Y. Huazi Zhang, Zhaoyang Zhang1 and X. Chen, "Distributed spectrum-aware clustering in cognitive radio sensor networks," in *Proc. of GLOBECOM 2011*.
13. B. E. Ali Jorio, Sanaa El Fkihi and D. Aboutajdine, "An energy-efficient clustering routing algorithm based on geographic position and residual energy for wireless sensor network," *Journal of Computer Networks and Communications*, vol. 2015, 04 '15.
14. D. Li and J. Gross, "Robust clustering of ad-hoc cognitive radio networks under opportunistic spectrum access," in *Proc. of IEEE ICC '11*.
15. S. Liu, L. Lazos, and M. Krunz, "Cluster-based control channel allocation in opportunistic cognitive radio networks," *IEEE Trans. Mob. Comput.*, vol. 11, no. 10, pp. 1436–1449, 2012.
16. J. Zhao, H. Zheng, and G.-H. Yang, "Spectrum sharing through distributed coordination in dynamic spectrum access networks," *Wireless Com. and Mobile Computing*, vol. 7, no. 9, 2007.
17. T. Chen, H. Zhang, G. Maggio, and I. Chlamtac, "Cogmesh: A cluster-based cognitive radio network," *Proc. of DySPAN '07*.
18. K. Baddour, O. Ureten, and T. Willink, "Efficient clustering of cognitive radio networks using affinity propagation," in *Proc. of ICCCN 2009*.
19. D. Wu, Y. Cai, L. Zhou, and J. Wang, "A cooperative communication scheme based on coalition formation game in clustered wireless sensor networks," *IEEE Transactions on Wireless Communications,*, vol. 11, no. 3, pp. 1190 –1200, march 2012.
20. A. Asterjadhi, N. Baldo, and M. Zorzi, "A cluster formation protocol for cognitive radio ad hoc networks," in *Proc. of European Wireless Conference 2010*, pp. 955–961.
21. M. Ozger and O. B. Akan, "Event-driven spectrum-aware clustering in cognitive radio sensor networks," in *Proc. of IEEE INFOCOM 2013*.
22. N. Mansoor, A. K. M. Muzahidul Islam, M. Zareei, S. Baharun, and S. Komaki, "Construction of a robust clustering algorithm for cognitive radio ad-hoc network," in *Proc. of CROWNCOM 2015*.
23. N. Mansoor, A. Islam, M. Zareei, S. Baharun, and S. Komaki, "Construction of a robust clustering algorithm for cognitive radio ad-hoc network," in *Proc. of CROWNCOM 2015*.
24. B. Clark, C. Colbourn, and D. Johnson, "Unit disk graphs," *Annals of Discrete Mathematics*, vol. 48, no. C, pp. 165–177, 1991.
25. Y. Zhang, G. Yu, Q. Li, H. Wang, X. Zhu, and B. Wang, "Channel-hopping-based communication rendezvous in cognitive radio networks," *IEEE/ACM Transactions on Networking*, vol. 22, no. 3, pp. 889–902, June 2014.
26. Z. Gu, Q.-S. Hua, and W. Dai, "Fully distributed algorithms for blind rendezvous in cognitive radio networks," in *Proceedings of the 2014 ACM MobiHoc*, ser. MobiHoc '14.
27. D. Li, E. Fang, and J. Gross, "Versatile Robust Clustering of Ad Hoc Cognitive Radio Network," *ArXiv e-prints, 1704.04828*.
28. A. MacKenzie and S. Wicker, "Game theory in communications: motivation, explanation, and application to power control," in *Proc. of IEEE GLOBECOM 2001*.
29. J. O. Neel, "Analysis and design of cognitive radio networks and distributed radio resource management algorithms," Ph.D. dissertation, Blacksburg, VA, USA, 2006, aAI3249450.
30. B. Wang, Y. Wu, and K. R. Liu, "Game theory for cognitive radio networks: An overview," *Comput. Netw.*, vol. 54, no. 14, pp. 2537–2561, Oct. 2010.
31. B. J. S. Chee and C. Franklin, Jr., *Cloud Computing: Technologies and Strategies of the Ubiquitous Data Center*, 1st ed. CRC Press, Inc., 2010.
32. H. Ackermann, H. Rglin, and B. Vcking, "Pure Nash equilibria in player-specific and weighted congestion games," *Theoretical Computer Science*, vol. Vol. 410, no. 17, pp. 1552 – 1563, 2009.


DOI: 10.1002/ett
*Prepared using ettauth.cls*

33. D. Li, E. Fang, and J. Gross, "Robust Clustering in Cognitive Radio Network with Cluster Size Control," 2017.
34. X.-Y. Li, Y. Wang, and Y. Wang, "Complexity of data collection, aggregation, and selection for wireless sensor networks," *IEEE Transactions on Computers*, vol. 60, no. 3, pp. 386–399, 2011.
35. M. Onus, A. Richa, K. Kothapalli, and C. Scheideler, "Efficient broadcasting and gathering in wireless ad-hoc networks," in *Proc. of ISPAN 2005*.
36. M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979.