Versatile Robust Clustering of Ad Hoc Cognitive Radio Network

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

***Abstract*—Cluster structure in CRN facilitates cooperative spectrum sensing, routing and other functionalities. The availability of unlicensed channels which are available for every member in a cluster decides the survival of that cluster from licensed users' influence. Thus in order to be robust against licensed users, there should be more unlicensed channels in the clusters. In the process of forming clusters, every secondary user needs to decide with whom to form a cluster, or which cluster to join. Congestion game model is adopted to analyse this process, which not only contributes the algorithm design directly, but also provides guarantee of convergence into Nash Equilibrium and convergence speed. Our proposed distributed clustering scheme outperforms the comparison scheme in terms of robustness against primary users, convergence speed and volume of control messages. Furthermore, the proposed clustering solution is versatile to fulfil other requirements such like fast convergence and cluster size control. Besides, we prove the clustering problem to be NP-hard, and also propose the centralized solution. The extensive simulation supports our claims.**

***Index Terms*—Cognitive Radio, Cluster, Robust, game theory, congestion game, distributed, centralised, size control.**

I. Introduction

COGNITIVE radio (CR) is a promising technology to solve the spectrum scarcity problem [1]. Unlicensed users access the spectrum allocated to them whenever there is information to be transmitted. In contrast, unlicensed users can only access the licensed spectrum after validating the channel is unoccupied by licensed users. This refers to the process of sensing a particular channel and verifying (with a previously specified probability of error) that it is not used by a primary user currently. In this hierarchical spectrum access model [2], the licensed users are also called primary users (PU), and the CR users are known as secondary users and constitute the cognitive radio networks (CRN).

As to the operation of CRN, efficient spectrum sensing is identified to be critical to the success of cognitive radio networks [3]. Cooperative spectrum sensing is able to effectively cope with noise uncertainty and channel fading, thus remarkably improves the sensing accuracy [4]. Collaborative sensing relies on the consensus of CR users within certain area, and decreases considerably the false sensing reports caused by fading and shadowing of reporting channel. In this regard, clustering is regarded as an effective method in cooperative spectrum sensing [5], [6], as a cluster forms adjacent secondary users as a collectivity to perform spectrum sensing together. Clustering is also efficient to enable all CR devices within the same cluster to stop payload transmission on the operating channel and initiate the sensing process, so that all the CR users[1] within the one cluster are able to vacate the channel swiftly when primary users are detected by at least one CR node residing in the cluster [7]. With cluster structure, as CR users can be notified by cluster head (CH) or other cluster members about the possible collision, the possibility for them to interfere neighbouring clusters is reduced [8]. Clustering algorithm has also proposed to support routing in cognitive ad-hoc networks [9].

The communication within a cluster is conducted in the spectrum which is available for every member in that cluster. Usually there are multiple unlicensed channels available for all the members in a cluster, which are referred as *common control channels* (CCC). When one or several members can not use one certain CCC because primary users are detected to appear on that channel, this channel will be excluded from the set of CCCs, in particular, if this channel is the working channel, then all the cluster members switch to another channel in the set of CCCs. In the context of CRN, as the activity of primary users is controlled by licensed operators which are generally not known to CR users, the availability of CCCs for the formed clusters is totally decided by primary users' activity. In other words, the availability of CCCs for clusters is passive and can not guaranteed. In CRN, one cluster survives the influence of primary users when at least one CCC is available for that cluster. As the channel occupation by primary users is assumed to be uncontrollable to the CR users, a cluster formed with more CCCs will survive with higher probability. Thus the number of CCCs in one cluster indicates robustness of it when facing ungovernable influence from primary users. As a result, how to form the clusters plays an important role on the robustness of clusters in CRN.

To solely pursue cluster robustness against the primary users' activity, i.e., to achieve more common channels within clusters, the ultimately best clustering strategy is ironically that each node constitutes one single node clusters. Apparently this contradicts our motivation of proposing cluster in cognitive radio network. This contradiction indicates that, the robustness discussed in terms of number of common channels carries little meaning when the sizes of formed clusters are not given consideration. Besides, cluster size plays import roles in certain aspects. For instance, cluster size is one decisive factor in power preservation [10], [11], and it also influences the

D. Li was with RWTH, Germany.
Erwin. Fang is with ETH, Switzerland.
J. Gross is with KTH, Sweden.
Manuscript received xxxx xx, 20xx; revised xxxx xx, 20xx.

[1] User and node are used interchangeably in this paper, in particular, user is used when its networking or cognitive ability are discussed or stressed, and node is used when the network topology is discussed.

accuracy of cooperative spectrum sensing [12]. Hence, cluster size should be given consideration when discussing cluster robustness against primary users.

In this paper, a decentralized clustering approach ROSS (RObust Spectrum Sharing) is proposed to cover the issues of robustness and size control of clusters in CRN. ROSS is able to form clusters with desired sizes, and the generated clusters are more robust than other clustering scheme which has claims on cluster robustness, i.e., more secondary users residing in clusters against increasing influence from primary users. Compared with previous work, ROSS involves much less control messages, and the generated clusters are significantly more robust. We also propose the light weighted versions of ROSS, which involve less overheads and thus are more suitable for mobile networks. Throughout this paper, we refer the clustering schemes on the basis of ROSS as the *variants of ROSS*, i.e., the light weighted versions, or the versions with size control feature.

The rest of paper is organized as follows. After reviewing related work in section II, we present our system model in Section III. Then we introduce our clustering scheme ROSS and its variants in section IV. The clustering problem is given through analysis and a centralized scheme is proposed in section V. Extensive performance evaluation is in section VI. Finally, we conclude our work and point out direction future research in section VII.

II. Related Work

Prior to the emergence of open spectrum access, as an important method to manage network, clustering has been proposed in for ad hoc networks [13], [14], [15], wireless mesh networks [9], and wireless sensor networks [9] and . In ad hoc and mesh networks, the major focus of clustering is to preserve connectivity (under static channel conditions) or to improve routing. In the context of sensor networks, the emphasis of clustering has been on longevity and coverage. Overhead generated by clustering in ad hoc network is analysed in [16], [17].

As to cognitive radio networks, clustering schemes are also proposed, which target different aspects. Work [12] improves spectrum sensing ability by grouping the CR users with potentially best detection performance into the same cluster. Clustering scheme [10] obtains the best cluster size which minimizes power consumption caused by communication within and among clusters. [10] proposes clustering strategy in cognitive radio network, which looks into the relationship between cluster size and power consumption and accordingly controlling the cluster size to decrease power consumption. Cogmesh is proposed in [18] to construct clusters by the neighbour nodes which share local common channels, and by interacting with neighbour clusters, a mesh network in the context of open spectrum sharing is formed. Robustness issue is not considered by this clustering approach. [19] targets on the QoS poisoning and energy efficiency. This approach first decides on the relay nodes which minimize transmission power consumption, then the chosen nodes become cluster heads and clusters are formed in a dynamic coalition process. This work emphasis on power efficiency and doesn't take into account the channel availability and the issue of robustness of the formed clusters. In [6], [20], the channel available to the largest set of one-hop neighbours is selected as common channel which yields a partition of the CRN into clusters. This approach minimizes the set of distinct frequency bands (and hence, the set of clusters) used as common channels within the CRN. However, bigger cluster sizes generally lead to less options within one cluster to switch to if the common channel is reclaimed by a primary node. Hence, this scheme does not provide robustness to formed clusters. [21] deploys cluster structure in order to implement common channel control, medium access with multiple channel and channel allocation. The node with the maximum number of common channels within its k-hop neighborhood is chosen as cluster head, but how to avoid one node appearing in multiple clusters is not given consideration.

Clustering robustness is considered in [22], [23]. The authors propose a distributed scheme where the metric is the product of cluster size and the number of common control channels. This scheme involves both cluster size and number of CCCs, but it is inherently flawed. With the metric, cluster could be formed only due to one factor of the two, e.g. a spectrum rich node will exclude its neighbour to form a cluster by itself. Besides, this scheme leads to a high variance on the size of clusters, which is not desired in certain applications as discussed in [10], [21].

III. System Model

Let us consider a two dimensional area where primary and secondary users coexist together. They share the licensed channels according to the spectrum overlay model, where secondary transmitters are only allowed to transmit when the primary users are detected as being idle, and they should vacate the working channel when the presence of primary user is detected. The set of primary users and secondary users are presented by $\mathcal{P}$ and $\mathcal{N}$ separately, there are $|\mathcal{P}| = P$ and $|\mathcal{N}| = N$. Cognitive radio ad hoc network is constituted by all the secondary users in $\mathcal{N}$. The collection of non-overlapping licensed frequency bands is denoted as $\mathcal{F}$ with $|\mathcal{F}| = F$, which is shared by the the primary and secondary users. We assume that primary users have a relatively low variation in activity (periods of activity and inactivity in the range of seconds or minutes). Primary users access the allocated channels in $\mathcal{F}$ according to its need without sending any explicit notification to secondary users.

Secondary users conduct spectrum sensing independently, and every secondary user conducts spectrum sensing on all licensed channels sequentially. The sensing duration and frequency on one channel is a technical problem [24], but we assume that every node can to certain extend accurately and agilely detect the presence of primary user on each channel. [2] The available channels sensed on secondary user i is denoted by V_i and there is $|V_i| \le F$.

[2]The spectrum availability can be validated with a certain probability of detection. Spectrum sensing/validation is out of the scope of this paper.

One dedicated control channel is assumed to be available for all the secondary nodes to exchange control messages with neighbours in the process of cluster formation. The control channel could be in ISM band or other reserved spectrum which is exclusively used for transmitting control messages. Note that the assumption of dedicated control channel is to simplify the discussion so that we can focus on the kernel of this paper, the robustness of clusters. Actually, the control messages involved in the clustering process can be conveyed on available licensed channels through rendezvous process by channel hopping [25], [26]. Over the control channel, secondary users exchange their spectrum sensing results V_i with one hop neighbours. Secondary users have the same transmission range on both licensed and control channel, and the communication links are reciprocal.

As to a pair of secondary users, when the distance between them is smaller than secondary users' transmission range, we assume the pair is able to communicate on both control channel and licensed channel, and the both are considered to be neighbours of each other. When a secondary user sits outside primary users' transmission ranges, it can not detect their presence. As the transmission range of primary users is limited and secondary users are at different locations, secondary users have different views of the spectrum availability (apart from the fact that there might be false negatives in the sensing process), i.e., $V_i \neq V_j$, for $i \neq j$. As the assumed $0/1$ state of connectivity is solely based on the distance between secondary users, CRN can be represented by a graph $\mathcal{G}(I, \mathcal{E})$, where $\mathcal{E} = \{(i, j, v) | i, j \in \mathcal{N} \wedge v \in V_i \wedge v \in V_j\}$ is wireless link between any secondary node i and its neighbour j with licensed channel v. Due to relatively low primary user dynamics, time index is omitted here. For secondary node i, its neighborhood Nb_i consists of all the secondary users locating within its transmission range, regardless whether common licensed channels exist or not. In the rest of this paper, *channel* only refers to the licensed common control channel unless the dedicated control channel is particularly mentioned.

A. Clustering

We give the description of cluster in the context of CRN, all the clusters discussed in this paper comply with it. A cluster C is composed with one cluster head and cluster members, which satisfies the following conditions:

- Cluster head H_C is able to communicate with any cluster member *directly*, i.e., for any cluster member $i \in C$, there is $i \in \mathtt{Nb}_{H_C}$.
- There exists as least one common control channel in the cluster, i.e., $\cap_{i \in C} V_i \neq \emptyset$.

Cluster head coordinates the activities of cluster members, i.e., notifies all the members to vacate a channel if primary user's presence is detected by one cluster member, or notify the members to use a different CCC for payload communication. Cluster is denoted as C_i, the subscript i represents the cluster head. We denote the set of common control channels for cluster C with set K_C. $K_C = \cap_{i \in C_i} V_i$, and $k_C = |K_C|$ is the number of common control channels for cluster C. Clustering is performed periodicity, as secondary users may move with certain velocity, or the most current clusters can not maintain due to lack of CCCs when primary users' operation is intense.

TABLE I
NOTATIONS IN ROBUST CLUSTERING PROBLEM

Symbol	Description
$\mathcal{P}$, $\mathcal{N}$	collection of primary and secondary users
P, N	the number of primary users and secondary users
$\mathcal{F}$	set of non-overlapping channels in the scenario
$\mathtt{Nb}_i$	node i' neighborhood
C	a cluster
C_i	a cluster whose cluster head is i (in Section V, C_i is the ith cluster among the legitimate clusters.)
V_i	set of available channels on CR node i
V_C	set of available common channels of cluster C
V_{C_i}	set of available common channels of cluster C_i
H_C	cluster head of a cluster C
δ	desired cluster size
S_i	set of claiming clusters, each of which includes debatable node i after phase I
K_{C_i}	set of common channels within cluster C_i
k_{C_i}	number of common channels of cluster C_i

IV. Distributed Coordination Framework: Clustering Algorithm

In this section, we present the clustering scheme ROSS. Briefly speaking, ROSS utilizes the similarity of the available spectrum within a local area to form clusters. ROSS consists of two cascaded phases: *cluster formation* and *membership clarification*. We will describe them sequentially.

A. Phase I - Cluster Formation

After conducting spectrum sensing and communication with neighbours, every CR node is aware of the available channels on itself and all its neighbours. Two metrics for each CR node are proposed to characterize the channel availability between the node and its neighborhood. As to CR node i, there are,

- *Individual connectivity degree* D_i: $D_i = \sum_{j \in \mathtt{Nb}_i} |V_i \cap V_j|$, which denotes the sum of the pairwise common channels of node i. It is an indicator of node i's adhesive property to the CRN.
- *Social connectivity degree* G_i: $G_i = |\bigcap_{j \in \mathtt{Nb}_i \cup i} V_j|$, which is the number of common channels of CR node i and its neighbours. G_i represents the ability of i to form a robust cluster with its neighborhood.

Individual connectivity degree D_i and social connectivity degree G_i together form the *connectivity vector*. Figure 1 illustrates an example CRN where each node's connectivity vector is calculated and shown.

After introducing connectivity vector, we proceed to introduce the first phase of algorithm ROSS, which can be sketched as follows, at first, cluster heads are determined, then clusters are formed on the basis of the cluster heads' neighbourhoods.

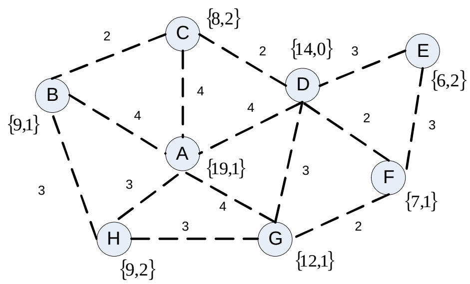


Fig. 1. Connectivity graph and the connectivity vector $\{D_i, G_i\}$ on each node. The available channels sensed by each CR node are: $V_A = \{1,2,3,4,5,6,10\}, V_B = \{1,2,3,5,7\}, V_C = \{1,3,4,10\}, V_D = \{1,2,3,5\}, V_E = \{2,3,5,7\}, V_F = \{2,4,5,6,7\}, V_G = \{1,2,3,4,8\}, V_H = \{1,2,5,8\}$. Dashed lines indicates two end nodes are within transmission range of each other. Each edge is labelled by the number of common channels between the two ends.

1) Determining Cluster Heads and Form the Initial Clusters: In this phase, each CR node comparing its connectivity vector with its neighbours to decide whether it is cluster head or not. When CR node i has bigger individual connectivity degree than any neighbours except for those which have already become cluster heads, then node i becomes clusters head. In other words, CR node i becomes cluster head if $D_i > D_k, \forall k \in \mathtt{Nb}_i \setminus CHs$ (CHs donate the cluster heads existing in $\mathtt{Nb}_i$). If there is another CR node j in its neighborhood, which has the same individual connectivity degree with i, i.e., $D_j = D_i$ and $D_j > D_k, \forall k \in Nb_j \setminus \{CHs \cup i\}$, then the node out of $\{i, j\}$ with higher social connectivity degree becomes cluster head, and the other one becomes one member of it. If $G_i = G_j$ as well, node ID is used to break the tie, i.e., the one with smaller node ID takes precedence and becomes cluster head. The node which becomes cluster head broadcasts a message on its eligibility of being cluster head to notify its neighbours, and claims its neighbourhood as its cluster. After receiving the notification from a cluster head, a CR node, e.g., i, is aware that it becomes one member of a cluster, then i sets its individual connectivity degree to zero. Then node i broadcasts its new individual connectivity degree to all its neighbours.

When a CR node i is associated to multiple clusters i.e., i receives multiple notifications on cluster head eligibility from different CR nodes, D_i is still set to zero. When a cluster member is excluded from a cluster (this may happen in the following steps to guarantee the existence of CCC or to control cluster size), that CR user's individual connectivity degree is restored to the original value which is further broadcast to its neighbours. The purpose of this temporary manipulation on individual connectivity degree is to let the CR nodes out side of the cluster be possible to become cluster heads, so that every CR node either becomes cluster head or a member of at least one cluster. In other words, with this manipulation on its individual connectivity degree, every CR node in the network will be involved in the clustering process, the proof will be given later. The final states of all the CR users in the CRN are described in the following theorem. Figure 1 provides an example about how do CR nodes decide cluster heads. Node B and H have same individual connectivity degree, $D_B = D_H$, but as $G_H = 2 > G_B = 1$, node H becomes cluster head. In Figure 1, the cluster C_H is $\{H, B, A, G\}$.

The pseudo code for deciding cluster head and forming initial clusters is in Algorithm 1 in appendix. We have the following theorem to show that as long as a secondary user's connectivity the CRN is not zero, the secondary user will always be integrated into a certain cluster.

Theorem IV.1: *Given a CRN, every secondary user is included into at least one cluster within N steps.*

The Proof is in Appendix 17.

According to Theorem IV.1, we can assign reasonable amount of time for phase I to complete.

2) Guarantee Availability of Common Control Channel: After deciding itself being cluster head, CR node broadcasts to notify its neighbours on control channel, meanwhile, i's initial cluster is formed immediately, which is i's neighborhood except for those nodes which have become cluster heads, i.e., $C_i = (\mathtt{Nb}_i \setminus CHs) \cup i$. Note this is the initial cluster, as it is possible that the formed cluster doesn't pose any common control channels. This problem can be solved with the following method.

As smaller cluster size increases the number of CCCs within the cluster, certain nodes are eliminated until there is at least one common channel. The elimination of nodes is performed according to an ascending list of nodes sorted by their number of common channels with the cluster head, which means, the cluster member which has the least number of common channels with the cluster head is excluded first. If there are nodes having the same number of common channels with cluster head, the node whose elimination brings in more common channels will be excluded. If this criterion meets a tie, the tie will be broken by deleting the node with smaller ID. It is possible that the cluster head excludes all its neighbours and resulting into a singleton cluster composed by itself. The pseudo code for cluster head to obtain at least one common channel is shown in Algorithm 2. As to the nodes eliminated in this procedure, they become either cluster heads or get included into other clusters later on according to Theorem IV.1.

3) Cluster Size Control in Dense CRN: In the introduction section, we have stated that cluster size should be given consideration to justify the concept of robustness of clusters, i.e., without specifying requirement on cluster sizes, small clusters will be generated to obtain more CCCs. Except for cooperative sensing, clusters need to conduct some other functionalities, and the cluster size should fall in a desired range [27], [28]. Here we illustrate the pressing necessity to control cluster size when CRN becomes dense via theoretical analysis and simulation.

Assuming the CR and primary users are evenly distributed and primary users occupy the licensed channels randomly, then both CR nodes density and channel availability in the CRN can be seen as homogeneous. Based on Algorithm 1, cluster heads are the CR nodes which possess the biggest individual connectivity degrees in their neighborhood respectively, and they are surrounded by CR nodes. The clusters formed are the neighborhood of cluster heads, which is decided by the transmission range and network density. When the CRN is extremely dense, consider a cluster with cluster head of CR node i, based on the rule for cluster head selection Algorithm 1, the nearest cluster head could locate just outside node i's neighborhood or its transmission range. An instance of this situation is shown in Figure 2. In the figure, black dots represent cluster heads, the circles denotes the transmission ranges of those cluster heads. Cluster members are not shown in the figure. Let l be the length of side of simulation plan

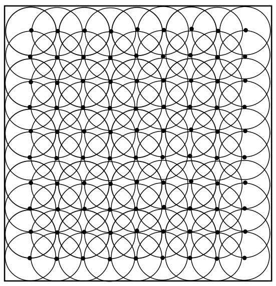


Fig. 2. Clusters formation in extremely dense CRN. Black dots are cluster heads, other cluster members are not drawn.

square, and r be CR's transmission radius. Based on the aforementioned analysis and geometry illustration as shown in Figure 2, we can give an estimate on the maximum number of generated clusters. The estimated number of cluster heads is the product of the number of cluster heads in row and line, $l/r * l/r = l^2/r^2$.

Now we verify the estimation with simulation. We distribute CR users and primary users randomly on a square plan, and set $r = 10, l = 50$. Network density is increased by adding more CR users. Figure 3 shows the number of formed clusters. With the increase of CR users in the network, network density increases linearly (see the right hand side Y axis, which indicates the number of neighbours.), and the number of formed clusters also increases and approaches to the the upper bound of 25 which complies with the estimation. For each network scale, simulation is run for 50 times. The confidence rate is 95% in the figure.

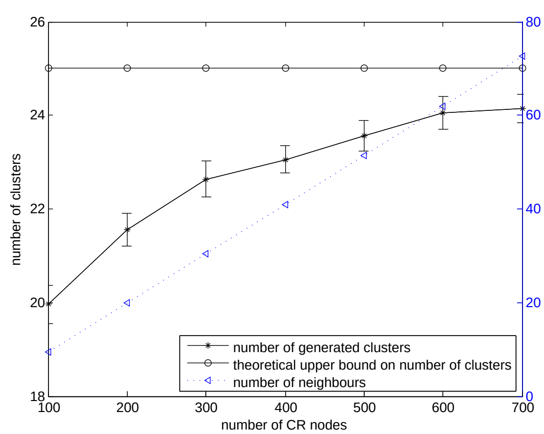


Fig. 3. The correlation between the number of formed clusters and network density. Note that the number of neighbours denotes the network density.

Both the analysis and simulation show that when applying ROSS, the cluster size also increases with the increase of network density. In case of dense network where cluster size is large, there is substantial burden on cluster heads to manage the cluster members, which is a challenge for resource limited cluster heads. As a result, certain measures are needed to prevent the network size to increase with the increasing network density. This task falls on cluster head. To control cluster size, cluster heads prune their cluster members if sizes are greater than the desired size δ. Given desired size as δ, cluster head excludes members sequentially, whose absence leads to the maximum increase of common channels within the cluster. This process ends when the size of resultant cluster is at most δ and at least one CCC is available. This procedure is similar with that to guarantee CCCs in cluster, thus the algorithm is also given in Algorithm 2. The nodes eliminated in this procedure later become a member or cluster head according to Theorem IV.1.

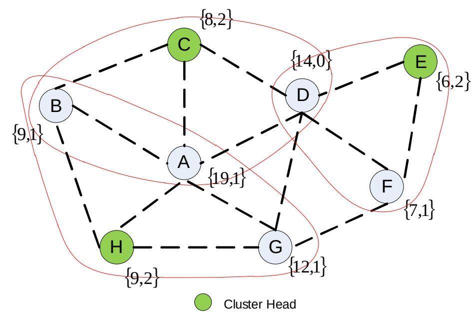


Fig. 4. Clusters formation after the first phase of ROSS. There are some nodes being debatable nodes, i.e., belonging to more than one cluster.

B. Phase II - Membership Clarification

After applying ROSS phase I on the example in Figure 1, we get the clusters shown in Figure 4. We notice there are several nodes, i.e., A, B, D, are included into more than one cluster. We refer these nodes as *debatable nodes* as their cluster affiliations are not clear, and the clusters which include debatable node i are called *claiming clusters* of node i, and are represented as S_i. Actually, debatable nodes extensively exist in CRN with larger scale. Figure 5 shows the percentage of debatable nodes increases with the scaling of CRN network.

Debatable nodes should be exclusively associated with one cluster and removed from the other claiming clusters, this procedure is called cluster membership clarification. We will introduce the solution for cluster membership clarification in the following.

1) Distributed Greedy Algorithm: After Phase I, debatable nodes, e.g., i needs to decide which cluster $C \in S_i$ to stay, and leaves the others. The principle for debatable node i to choose one claiming cluster is to result in the greatest increase of common channels in all its claiming clusters. Node i communicates with all the cluster heads whose clusters are in S_i, and is aware of the vector of common channels of the claiming clusters, then i is able to calculate how many more common channels in one certain claiming cluster if i leaves that cluster. Based on this calculation, i decides in which claiming cluster to stay and leaves the other claiming clusters.

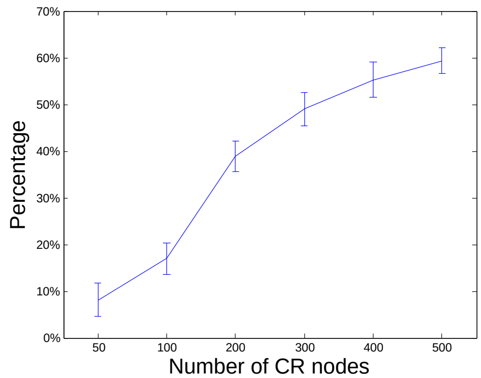


Fig. 5. The percentage of debatbale nodes among all CR nodes after the first phase of ROSS.

If there exists one cluster $C \in S_i$, by leaving from which the clusters in S_i obtain the minimum increased common channels than leaving any other claiming clusters, then i chooses to stay in cluster C. When there comes a tie in terms of the increase of common channels among multiple claiming clusters, i chooses to stay in the cluster whose cluster head shares more common channels with i. In case there are multiple claiming clusters demonstrating the same on the aforementioned metrics, node i chooses to stay in the smallest claiming cluster. IDs of cluster heads will be used to break tie if the previous rule doesn't decide on the unique cluster to stay.

Algorithm for debatable node i to decide which claiming cluster to stay is described as Algorithm 3. To conduct Algorithm 3, debatable node i needs to know the necessary information about its claiming clusters, i.e., V_C, V_{H_C}, $|C|$,$C \in S_i$, which are respectively the set of available channels in C, the set of available channels on C' cluster head H_C, and C' cluster size. Node i decides which cluster to stay based on Algorithm 3, then notifies all its claiming clusters, and retrieves the updated information of the necessary information V_C, V_{H_C}, $|C|$, where $C \in S_i$.

This procedure raises the concern on the infinite chain effect that debatable nodes update their choices based on other debatable nodes' choices, and this process never ceases. Consider the following example, where debatable node i locates in cluster $C \in S_i$, and C has more than one debatable node except for i. Assuming that i makes decision on which cluster to stay, which is followed by the other debatable nodes j to decides its affiliation, and there is $j \in C \in S_i$. The choice of j may change C's members, i.e., j leaves cluster C, which could possibly triggers node i to alter its previous decision. Thus, we must answer this question raised when implementing ROSS-DGA. In the following we show that the process of membership clarification can be formulated into a singleton congestion game, and a equilibrium state is reached after a finite number of best response updates.

2) Bridging ROSS-DGA with Congestion Game: To formulate the problem of membership clarification for the debatable nodes in the context of a game, we see this process from a different (or opposite) perspective. In the new perspective, the debatable nodes are regarded as isolated and don't belong to any cluster, which means the clusters they used to belong to become their neighbouring clusters. Then as to each debatable node, the previous problem to decide which clusters to leave becomes a new problem that which cluster to join. In this new problem, debatable node i (note now $i \notin S_i$) chooses one cluster C out of S_i to join if the decrement of common channels in cluster C is the smallest in S_i, and the decrement of CCCs in cluster C is $\sum_{C\in S_i} \Delta|K_C| = \sum_{C\in S_i}(|K_C| - |K_{C\cup i}|)$. The relation between debatable nodes and claiming clusters is shown in Figure 6.

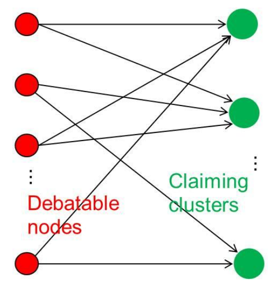


Fig. 6. Debatable nodes and claiming clusters

In the following, the debatable nodes constitute the players, and the 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 can be represented by a 4-tuple $\Gamma = (\mathcal{P}, \mathcal{R}, (\sum_i)_{i\in\mathcal{N}}, \Delta|K_C^i|)$, where elements in Γ are given as below,

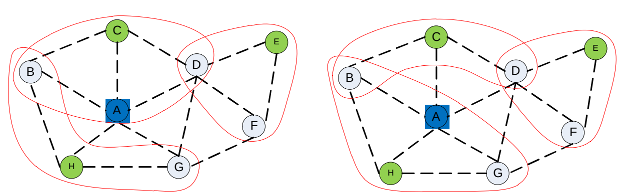
- $\mathcal{P}$, the set of players of the game, which are the debatable nodes after phase I in our clustering problem.
- $\mathcal{R} = \cup S_i, i \in \mathcal{P}$, denotes the set of resources for players to choose, S_i is the set of claiming clusters of node i. $\mathcal{R}$ is the set of claiming clusters after phase I in our clustering problem.
- As to the strategy space $\sum_i$ of player $i \in \mathcal{N}$, there is $\sum_i \subseteq 2^{[S_i]}$. As one debatable node is supposed to choose one claiming cluster in our problem, thus only one resource is allocated for i, accordingly this congestion game is a singleton game.
- The utility (cost) function $f(C)$ of resource $C \in R$, (or to say $f(r)$ of resource $r \in \mathcal{R}$) is $\Delta|K_C^i|$ which represents the decrement of CCCs in cluster C caused by debatable node i' joining in it. As to cluster $C \in S_i$, the decrement of CCCs caused by enrolment of debatable nodes is $\sum_{i:C\in S_i, i\to C} \Delta|K_C^i|$. $i \to C$ means i joins in cluster C. Obviously this function is non-decreasing with respect to the number of nodes joining in cluster C.
  The utility function is not purely decided by the number of players (debatable nodes) as that in a canonical congestion game, as in this game the channel availability on debatable nodes is different. Given two same sized groups of debatable nodes, when the nodes are not completely the same (neither are the channel availabilities on these nodes), the cost happened on one claiming cluster could be different if the two groups of debatable nodes join in that cluster respectively. Hence, this game is called player specific. In this game, every player greedily updates its

strategy (choosing one claiming cluster to join) if joining in a different claiming cluster minimizes the decrement of CCCs $\sum_{i:C\in S_i} \Delta|K_C^i|$, player's strategy in the game is exactly the same with the behaviour of debatable node in membership clarification phase, which is described by Algorithm 3.

As to singleton congestion game, there exists pure equilibria which can be reached with best response update, and the number of steps before convergence is $O(n^2 * m)$ [29], 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 (or clusters heads). Thus the number of steps is $O(N^3)$. In fact, the actual number of steps is much smaller than N^3. The amount of debatable nodes is illustrated in Figure 4, which is between 10% to 80% of the total number of CR nodes in the network. The number of clusters heads, as discussed in the part of cluster size in Section IV-A, is decided by the network density and the CR node's transmission range. Only a small part of the CR nodes become cluster heads, as in the example shown in Figure 3, the number of clusters is only 3.4% to 20% of the total number of CR nodes.

3) Distributed Fast Algorithm (DFA): The convergence speed of DGA is large recalling that the number of steps is of $O(N^3)$. Here we propose a faster algorithm DFA which is especially suitable for CRN where channel availability change dynamically and re-clustering is necessary. In DFA, each debatable node executes only one iteration of Algorithm 3 (by setting 'the current value' in Line 14 to zero). Every cluster includes all its debatable nodes, thus the membership is static and debatable nodes can make decisions simultaneously without considering the change of membership of its claiming clusters.

For example, in Figure 4, 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's clarification is illustrated in Figure 7. In Figure 7(a), node A staying in C_C and leaving C_H brings 2 more CCC into S_A, which is more than that brought by another strategy showed in 7(b). After the decisions made similarly by the other debatable nodes B and D, the final clusters formed are shown in Figure 8.



(a) Node A stays in cluster C_C, quits C_H, $\Delta|K_{C_C}|+\Delta|K_{C_H}|=2$ (b) Node A stays in cluster C_H, quits C_C, $\Delta|K_{C_C}|+\Delta|K_{C_H}|=1$

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

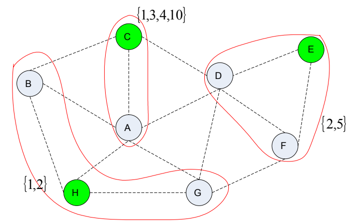


Fig. 8. Final formation of clusters, common channels for each cluster is shown.

V. Centralized Clustering Scheme

The centralized clustering scheme aims to form clusters with desired sizes, meanwhile the total number of common channels of all clusters is maximized. In the following, we refer this problem as *centralized clustering* for short, and the problem definition is as follows,

Definition 1: *Centralized clustering in CRN.*

Given a cognitive radio network $\mathcal{N}$ where nodes are indexed from 1 to N sequentially. Based on certain correlation, certain secondary users constitute one cluster C. $1 \le |C| \leqslant k$ where $|C|$ is size of cluster C and k is positive integer. We name the collection of such clusters as $\mathcal{S} = \{C_1, C_2, \ldots, C_{|\mathcal{S}|}\}$ (the subscript i is the unique index of cluster in $\mathcal{S}$, not the ID of cluster head of relevant cluster), $\mathcal{S}$ has following properties: $\bigcup_{1\le i\le l} C_i = N$ and $V_{C_i} \neq \emptyset$ for any i which satisfies $1 \le i \le l$.

Following condition distinguish the centralized clustering problem discussed in this thesis. The number of common channels is denoted as f which is $|V_C|$ if $|V_C| > 1$, and $f = 0$ if $|V_C| = 1$. The question of this problem is to find a subcollection $\mathcal{S}' \subseteq \mathcal{S}$, so that $\bigcup_{C_j\in\mathcal{S}'} C_j = N$, and $C'_j \cap C_j = \emptyset$ for $C'_j, C_j \in \mathcal{S}'$, so that $\sum_{C\in\mathcal{S}'} f$ is maximized. The decision version of centralized clustering in CRN is to ask whether exist $\mathcal{S}' \subseteq \mathcal{S}$, so that $\sum_{C\in\mathcal{S}'} f \geqslant \lambda$ where λ is a real number.

A. Complexity of Clustering Problem

In the following part of this section, we will discuss the complexity of centralized clustering problem and provide a centralized solution for it. We put the definition of weighted k-set packing problem here as it will be used in the analysis on the complexity of our problem.

Definition 2: *Weighted k-set packing.*

Given a set $\mathcal{G}$ which contains finite number of positive integers, and a collection of set $\mathcal{Q} = \{s_1, s_2, \cdots, s_m\}$, where for each element s_r, $1 \le r \le m$, there is $s_r \subseteq \mathcal{G}$, $1 \le |s_r| \le k$, and s_r has an associated weight which is positive real number. The question is whether exists a collection $\mathcal{S} \subseteq \mathcal{Q}$, where $\mathcal{S}$ contains only disjoint sets and the total weight of sets in $\mathcal{S}$ is greater than λ. Weighted k-set packing is NP-hard when $k \geqslant 3$. [30]

Theorem V.1: *CRN clustering problem is NP-hard, when the maximum size of clusters $k \geqslant 3$.*

The proof is in Appendix 17.

B. Centralized Optimization

As there is no efficient algorithm to solve clustering problem in CRN, we adopt binary linear programming to solve the problem. Note that binary linear programming is in NP-complete.

Given a CRN $\mathcal{N}$ and desired cluster size δ, we obtain a collection of clusters $\mathcal{G}$ which contains all the *legitimate* clusters, and the sizes of these clusters are $1, 2, \ldots, \delta$. Legitimate clusters are the clusters which satisfy the conditions in Section III-A. Note that the legitimate clusters include the singleton ones, which guarantees the partition of any network is feasible.

With $n = |\mathcal{N}|, g = |\mathcal{G}|$, we construct a $g \times n$ matrix Q_{gxn}. The element of matrix Q is q_{ij}, where the subscript i denotes a legitimate cluster, and j denotes a CR node, actually, j can be seen as the node ID. There are $i \in \{1, 2, \cdots, g-1, g\}$, and $j \in \{1, 2, \cdots, n-1, n\}$. Element $q_{ij} = |k_{C_i}|$ if node $j \in C_i$, and $q_{ij} = 0$ if $j \notin C_i$. Note that C_i means ith cluster in the collection $\mathcal{G}$, doesn't denote the cluster where cluster head is i, this notation is only valid in this subsection. In other words, each non-zero element q_{ij} denotes the number of CCC of the cluster i where node j resides.

$$Q = \begin{array}{c} \\ 1 \\ 2 \\ \vdots \\ i \\ \vdots \\ \vdots \\ g \end{array} \begin{array}{c} \begin{array}{cccccccc} 1 & 2 & 3 & \cdots & j & \cdots & n-1 & n \end{array} \\ \begin{pmatrix} k_1 & k_1 & 0 & \cdots & \cdots & \cdots & 0 & 0 \\ k_2 & 0 & k_2 & \cdots & \cdots & \cdots & 0 & 0 \\ \vdots & & & \vdots & & \vdots & & \\ 0 & k_i & 0 & \cdots & \cdots & \cdots & k_i & 0 \\ \vdots & & & \vdots & \cdots & \cdots & & \vdots \\ \vdots & 0 & 0 & \cdots & \cdots & \cdots & k_{i'} & 0 \\ k_g & & & \vdots & \cdots & \cdots & & \vdots \end{pmatrix} \end{array}$$

Fig. 9. Matrix Q, its rows correspond to all legitimate clusters, and columns correspond to the CR nodes in the CRN.

We build another $g \times n$ binary matrix X, which illustrates the clustering strategy. The element of matrix X is binary variable $x_{ij}, i = 1, \ldots, g, j = 1, \ldots, n$. $x_{ij} = 1$ denotes cluster i is one partition chosen by the clustering scheme, $x_{ij} = 0$ means this partition is not adopted. Note that matrix Q contains only constant elements, and matrix X contains only binary variables.

Now, we can formulate the optimization problem as follows,

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

The objective is the sum of two parts, the first part is the sum of products of cluster size and the corresponding number of CCCs, the second part is the *punishment* for choosing the clusters whose sizes are not δ. We notice that the first part is the metric adopted by the scheme SOC [22], in fact, the second part is particularly designed to eliminates the drawbacks of SOC, i.e., a large number of singleton clusters, or a few very large clusters which access affluent unlicensed spectrum. In practical computation, we minimize the opposite of the sum of the products of cluster size and the corresponding number of CCCs, thus the punishment is positive. The first constraint restricts each node j to reside in exactly one cluster. The second constraint regulates that when ith legitimate cluster C_i chosen, the number of elements which equal to 1 in the ith row is $|C_i|$.

Now we explain how is the mechanism of the punishment in the objective function. w_i is an auxiliary binary variable, which denotes whether cluster C_i is chosen by the solution, in particular,

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

and p is regulated as follows,

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

where $\alpha_i > 0$ and increases when $|C_i|$ diverges from δ. Because of w_i, any chosen cluster brings certain *punishment*. Function p denotes that when the chosen cluster's size is desired size δ, the punishment is zero. When the chosen cluster's size diverges from δ, the objective function suffers *loss*. Choice of α_i affects the resultant clusters.

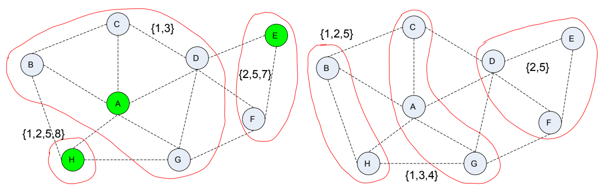
This is a integer linear optimization problem, which is solved by function *bintprog* provided in MATLAB. Note that the proposed centralized solution is heuristic. There are two reasons for pursuing the heuristic scheme, first, the problem of centralized clustering is NP hard, and there is no efficient solution to solve it. The second reason is, the collection of legitimate clusters is dependant on the network topology and spectrum availability in the network, thus to each specific CRN, the space of solution is different.

1) Example of the Centralized Optimization: To make the solution easier to understand, we look into the toy example of the CRN in Figure 1.

We let the cluster size δ to be 3. A collection of clusters $\mathcal{G}$ is obtained, which contains all the clusters satisfying the conditions of cluster in Section III-A and the sizes of clusters are 1, 2 or 3. $\mathcal{G} = \{\{A\}, \{B\}, \ldots, \{B, C\}, \{B, A\}, \{B, H\}, \cdots, \{B, A, C\}, \{B, H, C\}, \{A, D, C\}, \cdots\}$, $g = |\mathcal{G}| = 38$. α_1 and α_1 are set as 0.2 and 0.8.

The clustering result of centralized programming is $\{\{D, E, F\}, \{A, C, G\}, \{H, G\}\}$, the numbers of CCCs are 2, 3, 3. As comparison, the solution resulted from ROSS is $\{\{B, H, G\}, \{C, A\}, \{D, E, F\}\}$, the numbers of CCCs are 2, 4, 2. By applying SOC, the clustering result is $\{A, B, C, D, G\}, \{E, F\}, \{H\}$, and the numbers of CCCs are 2, 3, 4. The clusters formed by SOC and linear programming are shown in Figure 10.

As to the average number of common channel, the results of ROSS, LP and SOC are 2.66, 2.66, and 3 respectively. Note there is one singleton cluster C_H generated. When the



(a) Resulted from SOC (b) Resulted from Centralized Optimization

Fig. 10. Final clusters formed in the example CRN network from Figure 1

singleton cluster $\{E\}$ is excluded, the average number of common channels of SOC drops to 2.5.

VI. Performance Evaluation

In this section, we evaluate the performances of all the variants of ROSS, i.e., ROSS-DGA and ROSS-DFA, and that with cluster size control functionalities. The latter is referred as ROSS-x-DGA/ROSS-x-DFA, where x is the desired cluster size. Note that ROSS generates clusters on based of cluster heads' neighbourhood, thus the desired cluster size is smaller than the average neighbourhood size. We choose SOC as comparison scheme. To the best of our knowledge, SOC [22] is the only work emphasizing on the robustness of clustering structure from all previous work on clustering in CRN. The authors of [22] compared SOC with other schemes based on the average number of common channels within each cluster, on which SOC outperforms other schemes by 50%-100%. This is because the schemes except for SOC are designed either for ad hoc network without consideration of channel availability [15], or for CRN but just considering basic connection among CR nodes [6]. Hence, we only compare the two versions of our scheme ROSS-DGA and ROSS-DFA with SOC to show the merits of ROSS, and also compare with the centralized scheme to see the gap with the global optima. We will investigate the following metrics:

- *Average number of common channels per non-singleton cluster.* This metric shows the robustness of the current non-singleton clusters. Non-singleton cluster refers the cluster whose cluster size is larger than 1, and can also be seen as unclustered node. Note that this metric along is biased and misleading, because the CR nodes with more channels could be formulated as singleton clusters. This happens in SOC solution, whose objective is to improve the average number of common channel over *all* clusters, i.e., including the singleton clusters, thus many CR nodes with more channels are turned into clusters.
- *Number of unclustered CRs with moderate and vigorous intensity of primary users' activities.* This is a straight forward metric which reflects the robustness of clusters, as this metric directly shows how many nodes can make use of the cluster structure. With this metric, we investigate the performance of different schemes under different availability of spectrum in the CRN. When we vary the intensity of primary users' activity, e.g., from low to medium level by increasing the number of primary users, this metric is the antonym of *survival rate*, i.e., , how many nodes are still within a certain cluster when some clusters have collapsed due to the newly added primary users.
- *Cluster sizes.* Specific clusters size is pursued in many applications due to energy preservation and the system design [10]. We will present the distribution of CRs residing in the formed clusters, and the number of generated clusters through multiple simulations.
- *Amount of control messages involved.*

The simulation is conducted with C++. Certain number of CRs and PUs are deployed within a squire whose edge is 100 m. We adopt the round disk model to simulate transmission. Transmission ranges of CR and PU are 10 m and 30 m respectively. As to CRs, the CR node residing within another CR node's transmission range is seen as neighbour of that CR node. If CR node locating within one PU's transmission range, the CR node is not allowed to use the channel which is being used by that PR. The number of licensed channels in simulation is 10, each PU is operating on each channel with probability of 50%.

Simulation is divided into two parts, in the first part, we investigate the performance of centralized scheme, and the gap between the distributed schemes with the centralized scheme. As there is no polynomial time solution available to solve the centralized problem, the simulation of this part is conducted in a small network. In the second part, we investigate the performance of the proposed distributed schemes thoroughly in the networks with different scales and densities.

A. Centralized Schemes vs. Decentralized Schemes

Coinciding with the system model in Section III, 10 primary users and 20 CR users are dropped randomly (with uniform distribution) within a square area of size A^2, where we set the transmission ranges of primary and CR users to $A/3$. There are 10 available channels. With this setting, the average number of neighbours of one CR user is around 5. Each primary user randomly occupies one channel, and CR users are assumed to be able to sense the existence of primary users and identify available channels. When clustering scheme is executed, around 7 channels are available on each CR node. All primary and CR users are assumed to be static during the process of clustering. The desired cluster size is 3, the parameters used in the *punishment* for choosing the clusters with undesired sizes are set as follows, $\alpha_1 = 0.4$, $\alpha_2 = 0.6$. Performance results are averaged over 50 randomly generated topologies with equal parameters. The confidence interval shown in figure corresponds to 95% confidence level.

1) Number of Common Control Channels in Non-singleton Clusters: Figure 11 shows the average number of common channel of non-singleton clusters, as the singleton clusters (in other words unclustered nodes) don't execute any functionalities of clusters, which are has be discussed in Section I. From Figure 11, we can see centralized schemes outperform distributed schemes. SOC achieves the most number of CCC than all the variants of ROSS. The reason is, SOC is liable to group the neighbouring CRs which share the most abundant spectrum together, no matter how many of them are, thus the number of CCC of the formed clusters is higher. But this method leaves considerable number of CRs which have less spectrum not in any clusters. As to the variants of ROSS, the procedure of debatable nodes greedily looking for better affiliation improves the number of CCC, thus ROSS-DGA with and without size control outperform ROSS-DFA and its size control version respectively. We also notice that, the size control feature doesn't affect the number of CCC for both ROSS-DGA and ROSS-DFA. This is because the desired cluster size happens to be the average size of clusters generated by ROSS-DGA and ROSS-DFA, then the size control functionality doesn't play effect to increase the number of CCCs.

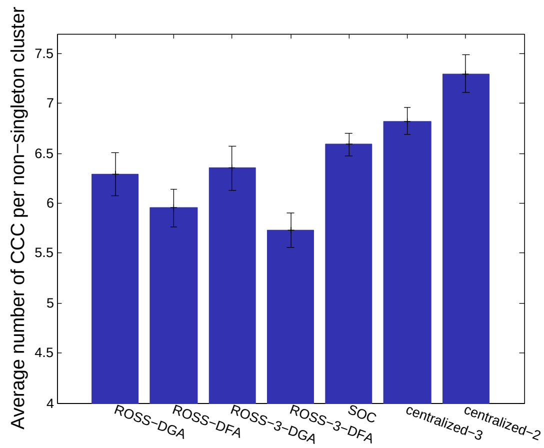


Fig. 11. Number of common channels for non-singleton clusters, the numbers in the names of schemes annotate the desired cluster size.

2) Survival Rate of Clusters with Increasing Primary Users: With the number of PRs in CRN increases, or their operation becomes more intensive, some CCCs will no longer be available. If there is no common control channels available any more because of the new added PRs, the cluster is regarded as destroyed and the former cluster member CRs become unclustered CRs or in other words singleton clusters.

We investigate the robustness of the formed clusters when they co-exist with primary users whose intensity of activities are varying. After the clusters are formed under the influence of the initial 10 PRs, extra 100 PRs are sequentially added into the network. The transmission range and channel occupancy of the new PU is the same with the previous ones, i.e., transmission range is $A/3$, and one channel out of 10 is randomly chosen to operate. Figure 12 shows the number of unclustered CRs with the increase of PRs, which indicates the vulnerability of clusters under varying surrounding of licensed spectrum.

We obtain three conclusions corresponding to three comparisons shown in this figure,

- Centralized scheme with cluster size of 2 produces the most robust clusters, and SOC results in the most vulnerable clusters. Centralized scheme with cluster size of 3 achieves less unclustered CRs than variants of ROSS when the number of PRs is 10∼30, when number of PUs is 30∼60, same amount of unclustered CRs are generated with variants of ROSS. When there are 75 and more new PRs, centralized scheme with cluster size of 3 results in more unclustered CR nodes than variants of ROSS. Size control feature makes both ROSS-DGA and ROSS-DFA outperform themselves without size control when number of new PRs is greater than 50.
  The reason that centralized scheme with cluster size of 3 does not completely excel variants of ROSS is due to the favourable achievement of it: the uniformly sized clusters. As distributed schemes, variants of ROSS generate considerable amount of smaller clusters which are more likely to survive when PRs' activities become intense. The comparison on cluster sizes will be given in details in VI-A3.
- ROSS with size control is better than the other two distributed schemes. The size control decreases the clusters size and makes the clusters more robust when under PRs' activity.
- Greedy algorithm improves survival rate. ROSS-DGA improves the survival rate of ROSS-DFA, so does ROSS-DGA with size control against ROSS-DFA with size control. This comply with the observation on number of CCC in section VI-A1. As the debatable CRs greedily update their affiliation with demanding clusters, and the metric for updating is the maximum increase of CCCs of the demanding clusters, the average number of common channels is improved (shown in Figure 11), then the robustness of clusters is enhanced. Meanwhile, sizes of more clusters become smaller also contributes more robustness.

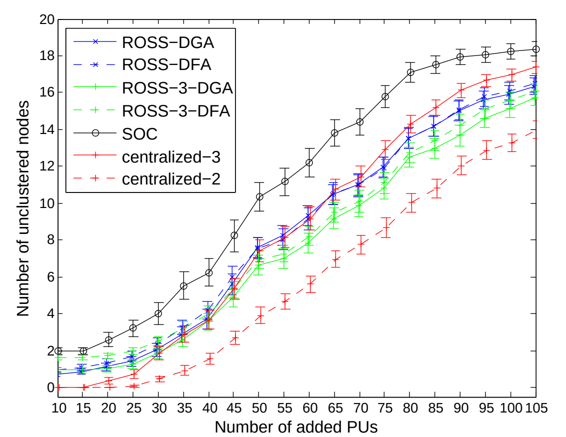


Fig. 12. Number of CRs which are not included in any clusters

3) Cluster Size Control: Figure 13 shows the number of CRs residing in certain sized clusters. The centralized schemes are able to form clusters which strictly satisfy the requirement on cluster sizes. When the desired size is 2, each generated cluster has two members. When the desired size is 3, in average only 3 CRs are formed into 2 node clusters. When ROSS-3-DFA is applied, most number of CRs are in 3 node clusters, nevertheless, slightly less nodes are found in 2 node and 4 node clusters, there are also considerable number of singleton clusters. ROSS-3-DGA decreases the clusters sizes and results in more 2 node clusters, the second most CRs are found in 3 node clusters. ROSS-DGA and ROSS-DFA generate rather even distribution of nodes with different sizes,

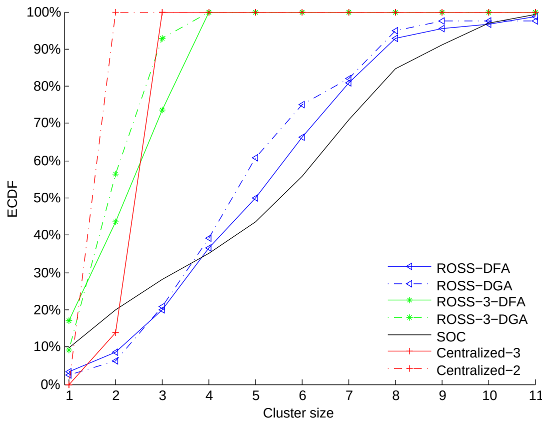


Fig. 13. Cumulative distribution of CRs residing in clusters with different sizes, as to ROSS with size control feature, the desired cluster size is 3. The average number of neighbours is 4.8.

whereas SOC results in more CRs unclustered or clusters of large sizes. Figure 13 shows distributed clustering schemes are not able to control cluster sizes perfectly, but ROSS-DGA and ROSS-DFA eliminate the clusters whose size diverges largely with the desired one, i.e., single node clusters and clusters with size of 13 and 14. Particularly, size control enable both ROSS-DGA and ROSS-DFA to achieve clusters whose sizes demonstrate certain homogeneity, i.e., cluster sizes vary from 1 to 4. But there are considerable number of single node clusters, which is due to the cluster pruning discussed in section IV-A3.

4) Control Signalling Overhead: In this section we compare the amount of control messages generated in different clustering schemes, e.g., centralized scheme, ROC, and the variants of ROSS.

In order to highlight the amount of control signalling only for clustering, we omit the process of neighbourhood discovery, which is premise for all clustering schemes. According to [32], the message complexity is defined as the number of messages used by all nodes. To have the same criterion to compare the overhead of signalling, we count *the number of transmissions of control messages*, without distinguishing they are sent with broadcast or unicast. This metric is identical to *the number of updates* we have discussed in Section IV.

As to ROSS, the control messages are generated in both phases. In the first phase, When a CR node decides itself to be the cluster head, it forms one cluster with its neighbourhood, each cluster head broadcasts one message containing its ID, cluster members and the set of CCCs in its cluster. In the second phase, debatable node informs its claiming clusters by broadcasting its affiliation, and the claiming cluster's cluster head broadcasts message about its new cluster if its cluster's members are changed. The total number of times for all CR nodes to send control messages, i.e., the total number of decisions related with clustering functionality, has been analysed in the part of convergence speed in Theorem IV.1 and Section IV-B2 respectively.

Comparison scheme SOC involves three rounds of execution. In the first two rounds, every CR node maintains one cluster and the final clustering solution is obtained in the third round. In each round, every CR node is involved in comparisons and cluster mergers.

The centralized scheme consists two phases. The first phase is information aggregation, where channel availability and neighbourhood of each CR node is sent to the centralized controller. The second phase is flooding, where the final clustering solution is disseminated over the network. Assume we use OSPF [31] to aggregate and disseminate information, and the best and worst complexity for is $O(E)$, where E is the number of edges in the graph which corresponds to the network. The minimum number of edges is $n-1$ when the nodes form a line and each node has at more two neighbours, and the maximum number is $n*(n-1)/2$ when the nodes form a complete graph. Thus the message complexity of the centralized scheme is $O(n)$ to $O(n^2)$.

The message complexity, quantitative analysis of the number of messages, and the size of control messages are shown in Table II,

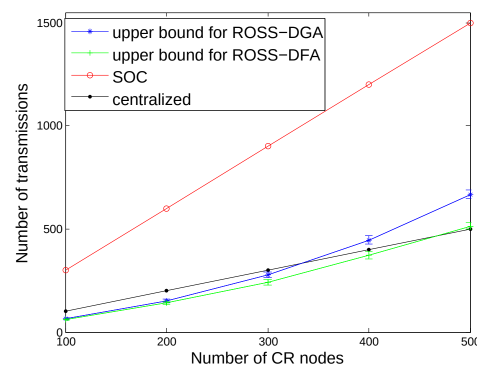


Fig. 14. Number of control messages, note that the curve for ROSS-DGA and ROSS-DFA is the upper bound for the number of messages.

B. Comparison between Distributed Schemes

In this section we investigate the performances of distributed clustering schemes in CRN with different network scales and densities. The transmission range of CR is $A/10$, PR's transmission range is $A/5$. The number of PU is 30. The number of CR is 100, 200 and 300, and the average number of neighbours of each CR is 9.5, 20, and 31.

1) Number of CCC per Non-singleton Clusters: Figure 15 illustrates the average number of CCCs of the non-singleton clusters. It shows when $N = 100$, variants of ROSS have 30% less CCCs than SOC, but this gap is decreased significantly when N is 200 and 300, i.e., when $N = 300$, number of CCCs achieved by ROSS variants (except for ROSS-x-DFA) is almost the same with that resulted from SOC.

This means SOC performs better on average number of CCCs per non-singleton clusters when network is sparse [3]. When the network becomes denser, even this metric favours SOC as discussed in the beginning of Section VI, ROSS-DGA achieves even more CCCs than SOC, and ROSS-DFA and ROSS-x-DGA increase the number of CCCs visibly.

[3]this is also observed in the evaluation in Section VI-A1 where $N = 20$

TABLE II
SINGALLING OVERHEAD. n-NUMBER OF CR NODES IN CRN, h: NUMBER OF CLUSTER HEADS, m: NUMBER OF DEBATABLE NODES, c: NUMBER OF DEMANDING CLUSTERS.

Scheme	Message Complexity	Quantitative number of messages	Content of message
ROSS-DGA, ROSS-x-DGA	$O(n^3)$	$h + 2 * m^2 c$ (upper bound)	$\mathtt{ID}_{H_C}$ and V_C for $h + m^2 c$ times, notification to join in one cluster for $m^2 c$ times
ROSS-DFA, ROSS-x-DFA	$O(n)$	$h + 2m$ (upper bound)	$\mathtt{ID}_{H_C}$ and V_C for $h + m$ times, notification to join in one cluster for m times
SOC	$O(n)$	$3 * n$	$\{V_i\}, i \in M \subseteq \mathtt{Nb}_i$
Centralized	$O(n) \sim O(n^2)$	$n \sim n^2$	$\{C\}$

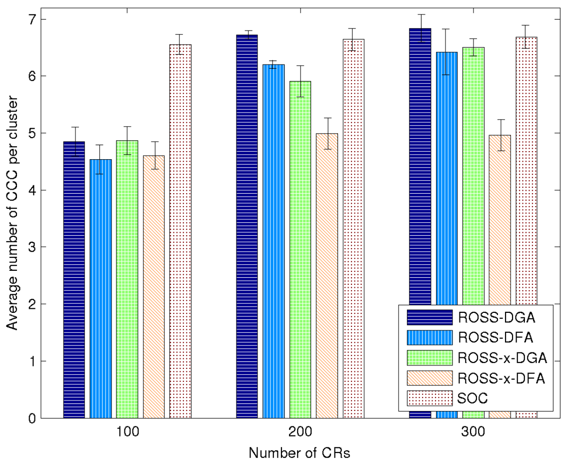


Fig. 15. Number of common channels of non-singleton clusters. As to ROSS with size control feature, we adopt $x = 6$ when $N = 100$, $x = 12$ when $N = 200$, $x = 21$ when $N = 300$, which is around $2/3$ of the number of average neighbours

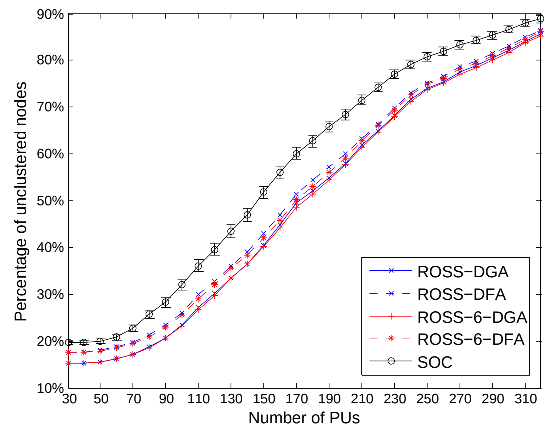


Fig. 16. Percentage of CRs which are not included in any clusters with the increasing number of primary users, $N = 100$

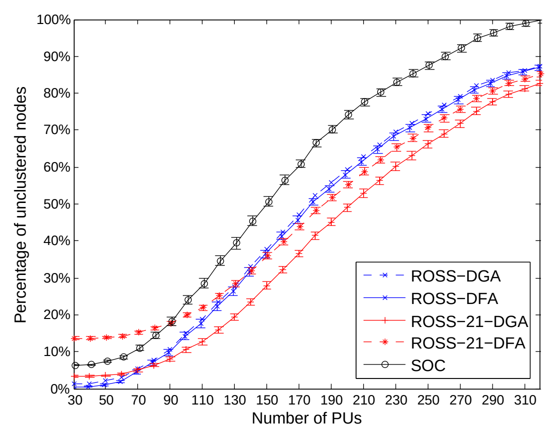


Fig. 17. Percentage of CRs which are not included in any clusters with the increasing number of primary users, $N = 300$

2) Survival Rate of Clusters with Increasing Primary Users: In this part of simulation, we investigate the robustness of clusters by increasing the PRs working on certain channels.

Figure 16 shows the increasing tread of singleton clusters, or to say, unclustered nodes, with the increase of PRs. SOC generates around 10 more singleton clusters than the variants of ROSS, which accounts for 10% of the total CR nodes. The confidence intervals of the variants of ROSS are not shown in the figure as they overlap, and we only show the average values. It can be seen that greedy algorithms result in slightly less singleton clusters than their counterparts.

Figure 17 shows a denser CRN where $N = 300$. SOC noticeably causes more singleton clusters than ROSS variants, except that ROSS-3-DFA results in more singleton clusters when PRs are few. The reason is that ROSS-3-DFA conducts cluster membership clarification for only once, which causes large number of singleton clusters, while, in ROSS-3-DGA increase the size of smaller clusters through debatable nodes' repeated updates thus drastically decreases the number of singleton clusters.

From the Figure 16 and 17, we can conclude that the greedy versions of ROSS are more robust than their counterpart variants of ROSS. When the network is denser, the improvement on cluster sizes and robustness by the greedy search in the membership clarification phase is more obvious.

3) Cluster Size Control: The number of formed clusters is shown in Fig. 18. When the network scales up, the number of formed clusters by ROSS increases by smaller margin. This result coincides with the analysis in Section IV-A3, that with ROSS, the number of formed clusters saturates when the network scales. When the network becomes denser, more clusters are generated by SOC compared with ROSS variants. To better understand the distribution of the sizes of formed clusters, we depict the cluster sizes with cumulative distribution. In this group of evaluation, the number of PRs is

30.

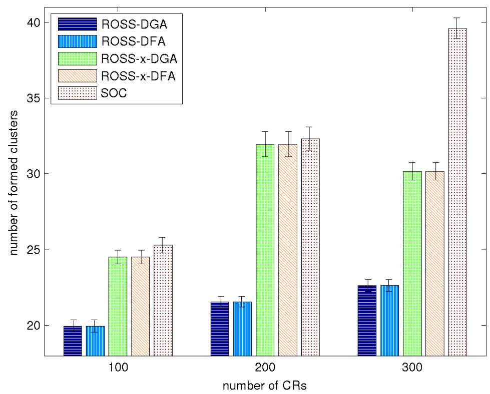


Fig. 18. Number of formed clusters. As to ROSS with size control feature, there are $x = 6$ when $N = 100$, $x = 12$ when $N = 200$, $x = 21$ when $N = 300$, which is around $2/3$ of the number of average neighbours.

Figures 19 20 21 illustrate the distribution of CR nodes which reside in clusters with certain sizes, where the CRN has different densities, i.e., the number of CRs is 100, 200 and 300 respectively. The desired cluster sizes are 5, 8 and 13, which are around 45% of the average number of neighbours in respective CRNs. When variants of ROSS are applied, the schemes with size control features generate clusters whose sizes are evenly distributed between 2 and the respective desired size. This means the size control feature effectively restricts the size of clusters, which is able to prevent the clusters from ungovernable growing with the increase of network density. The sizes of clusters generated by ROSS-DGA and ROSS-DFA span a wider range than that with feature control feature. But the range where most of the CRs (80% which exclude the CR nodes residing in small and big clusters) resides centres a value, which is roughly the half of the average number of neighbours. Overviewing the three Figures, we can see ROSS-DGA and ROSS-DFA show similar behaviour on cluster sizes. The clusters generated from SOC demonstrate randomness on cluster sizes, which diverge strongly.

VII. Conclusion

In this paper we design a distributed clustering scheme with the singleton congestion model, which forms robust clusters against primary users' effect. Through simulation and theoretical analysis, we find that distributed scheme achieves similar performance with centralized optimization in terms of cluster survival ratio and number of control messages. This paper investigates the robust clustering problem in CRN extensively, and proves the NP hardness of this problem. A Light weighted clustering scheme ROSS is proposed, on the basis of which, we propose the fast version scheme and the scheme which generate clusters with desired sizes. These schemes outperform other distributed clustering scheme in terms of both cluster survival ratio and the number of control messages.

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

Algorithm 1: Cluster head determination and initial cluster formation for Unclustered CR node i

Input: $D_j, G_j, j \in \mathrm{Nb}_i \setminus CHs$. Empty sets τ_1, τ_2
Result: 1 means i is cluster head, then $D_j, j \in \mathrm{Nb}_i \setminus CHs$ is changed as a big positive value M. 0 means i is not.

if $\nexists j \in Nb_i \setminus CHs$, *such that* $D_i \leq D_j$ **then**
  return 1;
end
if $\exists j \in Nb_i \setminus CHs$, *such that* $D_i < D_j$ **then**
  return 0;
else
  if $\nexists j \in Nb_i \setminus CHs$, *such that* $D_j == D_i$ **then**
    $\tau_1 \leftarrow j$
  end
end
if $\nexists j \in \tau_1$, *such that* $G_i \leq G_j$ **then**
  return 1;
end
if $\exists j \in \tau_1$, *such that* $G_i < G_j$ **then**
  return 0;
else
  if $\nexists j \in \tau_1$, *such that* $G_j == G_i$ **then**
    $\tau_2 \leftarrow j$
  end
end
if $\mathtt{ID}_i$ *is smaller than any* $\mathtt{ID}_j$, $j \in \tau_2 \setminus i$ **then**
  return 1;
end
return 0;

Proof of Theorem IV.1

Proof. Note the formed cluster can be a singleton cluster, i.e., cluster size is 1. To simplify the proof, we assume secondary users have unique individual connectivity degrees. This is fair as there are other metrics to break the tie according to Algorithm 1, i.e., the social connectivity degrees and node ID. Assuming there are some secondary users which are not assigned to any cluster and node α is one of them. As node α is not contained in any cluster, there must be at least one node $\beta \in Nb_\alpha$, with $D_\alpha < D_\beta$. Otherwise, node α is eligible to form a cluster. Then, as node β form a cluster and include node α, we can repeat the analysis on node α, and deduce that

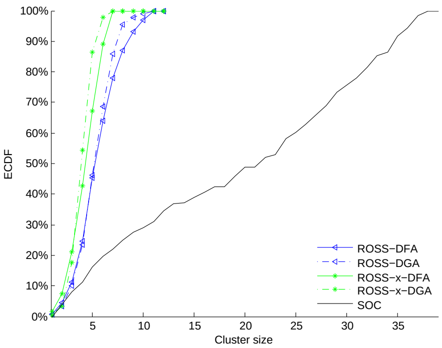


Fig. 19. 100 CRs, 30 PRs in network, the average number of neighbours of CR node is 9.5. The desired cluster size x is 6.

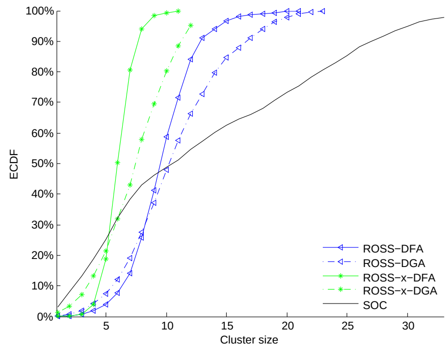


Fig. 20. 200 CRs, 30 PRs in network, the average number of neighbours of CR node is 20. The desired cluster size x is 12.

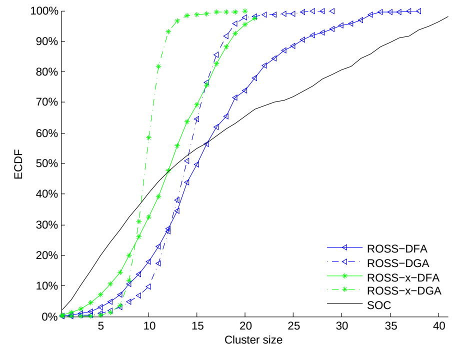


Fig. 21. 300 CRs, 30 PRs, the average number of neighbours is 30. The desired cluster size x is 21.

Fig. 22. Cumulative distribution of CRs residing in clusters with different sizes

Algorithm 2: Guarantee the availability of CCC / cluster size control by cluster head

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

/* When to guarantee available CCCs, execute from line 1, when to control cluster size, execute from line 2 */

1 **while** $V_C = \emptyset$ **do**
2 **while** $|C| > \delta$ **do**
3 **if** $\exists$ *only one* $i \in C \setminus H_C$, $i = \arg\min(|K_{H_C} \cap K_i|)$ **then**
4 $C = C \setminus i$;
5 **else**
6 $\exists$ multiple i which satisfies $i = \arg\min(|K_{H_C} \cap K_i|)$;
7 $\tau_1 \leftarrow i$;
8 **end**
9 **if** $\exists$ *only one* $i \in \tau_1$, $i = \arg\max(|\cap_{j \in C \setminus i} V_j| - |\cap_{j \in C} V_j|)$ **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**

Algorithm 3: Debatable node i decides its affiliation, chooses one claiming cluster to stay and leaves all the other claiming clusters

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.

if $\exists$ *only one* $C \in S_i$, $C = \arg\min(|K_{C \setminus i} - |K_C|)$ **then**
return C;
else
$\exists$ multiple $C \in S_i$ which satisfies $C = \arg\min(|K_{C \setminus i} - |K_C|)$;
$\tau_1 \leftarrow C$;
end
if $\exists$ *only one* $C \in \tau_1$, $C = \arg\max(V_{H_C} \cap V_i)$ **then**
return C;
else
$\exists$ multiple $C \in S_i$ which satisfies $C = \arg\max(V_{H_C} \cap V_i)$;
$\tau_2 \leftarrow C$;
end
if $\exists$ *only one* $C \in \tau_2$, $C = \arg\min |C|)$ **then**
return C;
else
return $\arg\min_{C \in \tau_2} \mathtt{ID}_{H_C}$;
end

node β has at least one neighbouring node γ with $D_\gamma > D_\beta$, and this series of nodes with monotonically increasing D_i might continue to grow and ceases finally because both of the individual connectivity degree and the total number of nodes are limited. The formed node series is shown as Figure 23.

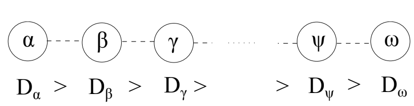


Fig. 23. The node series discussed in the proof for Theorem IV.1, the deduction begins from node α

Now we find the node ω is in the end of this series. As ω is the end node and does not have neighbouring nodes with bigger individual connectivity degree D, ω becomes cluster head and incorporate all its one-hop neighbours, including the node before it in the node series (here we assume that every new formed cluster has common channels). After that, the node recruited into cluster will set its connection degree D to zero, which enables the node further down in the list to become a cluster head. In this way, all the nodes in the series are included in at least one cluster in an inverse sequence. This result contradicts the initial assumption and proves the claim stated above. Meanwhile, through this proof, we know that within at most N steps, all nodes will become a part of certain clusters. □

Proof of Theorem V.1

Proof. To see centralized clustering problem is NP-hard, we reduce the NP-hard problem *weighted k-set packing* to it. To complete the reduction, we need to conduct following two steps:

- step 1: Show there exists a polynomial algorithm σ, by which any instance (e.g., $\mathcal{S}$) of a weighted k-set packing can be transformed to instance $\sigma(\mathcal{S})$ for centralized clustering.
- step 2: Show that $\mathcal{S}$ is a *yes* instance of weighted k-set packing if and only if $\sigma(\mathcal{S})$ is an *yes* instance for CRN clustering problem.

We continue using the notation introduced in the problem definition. Let set $\mathcal{G}$ contains N positive integer numbers which are indexed from 1 to N sequentially. Assume one instance $\mathcal{S}$ of weighted k-set packing is a collection of disjoint sets $Q = \{s_1, s_2, \cdots s_m\}$, and each set is composed by certain amount of elements in $\mathcal{G}$. ω indicates the weight for each set s, $\omega : \mathcal{S} \rightarrow \mathbb{Z}^+$.

The polynomial algorithm σ consists of three transformations.

- In the first transformation, for each set s_i of instance $\mathcal{S}$, the elements are duplicated, for instance, given $s_i = \{1, 4, 6\}$, the dummy set s'_i is $\{1, 1, 4, 4, 6, 6\}$. By doing this, we obtain the dummy sets and constitute the dummy instance $\mathcal{S}'$ based on $\mathcal{S}$. The purpose of this transformation is to eliminate the single element set in $\mathcal{S}$. The weight of set is unchanged after this transformation, i.e., $\omega(s_i) = \omega(s'_i)$. After this transmission, there is no set with only one element. This transformation requires $\sum_{s_i \in \mathcal{S}} |s_i|$ steps.
- In the immediate following second transformation, we transform the dummy instance $\mathcal{S}'$ to an instance for CRN clustering problem. Given an instance $\mathcal{S}'$, we retrieve all the elements which appear in it, and map each of those elements into one CR node, i.e., each integer corresponds to one CR node, particularly, that integer becomes the CR node's ID. As to duplicated elements, we also map them into a CR node, thus there exist CR nodes with the same ID. As a result, these CR nodes constitute a collection of CR nodes, but note that they have not constituted one CRN yet as there are not connections drawn among them. Connections in CRN under this context is decided by physical conditions, which says the corresponding CR nodes have common channels and close enough to communicate with each other. This transformation requires $2 \cdot \sum_{s_i \in \mathcal{S}} |s_i|$ steps.
- Mere isolated nodes don't constitute network, thus we add connections in CRN based on the sets in $\mathcal{S}'$ sequentially. For each set $s' \in \mathcal{S}'$, we add connection between two CR nodes if their IDs are in s'. There is also connection between the CR node and its dummy node. The number of common channels of the CR nodes equals to the weight of set s'. No connection exists between two CR nodes if their IDs don't appear in one set in Q. Afterwards, the CR node whose ID doesn't appear in any set in $\mathcal{S}'$ becomes single node clusters, according to the definition of clustering problem in CRN, the number of common channels is 0. This procedure requires $\sum_{s'_i \in \mathcal{S}'} |s_i|$ steps to map sets in $\mathcal{S}'$ into CRN and connections, and at most N steps to complement the single node clusters in CRN.
  The number of common channels of cluster f is non-decreasing function of cluster size, while, the weight of set in weighted k-set packing problem doesn't have this property. In weighted k-set packing, the weight of a set with smaller size could be larger than a set with more elements. But this difference doesn't hinder the transformation and we use an example to explain. Assuming two sets in $\mathcal{S}$ are $s_1 = \{1, 2\}$ and $s_2 = \{1, 2, 3, 4\}$, then their weights are 3 and 5 respectively. Their dummy sets are $s'_1 = \{1, 1, 2, 2\}$ and $s'_2 = \{1, 1, 2, 2, 3, 3, 4, 4\}$ and their new weights are 3 and 5 as before. The connections mapped to CRN are contradictory to reality, as the number of common channels of CR node group $\{1, 1, 2, 2\}$ can only be smaller than that of $\{1, 1, 2, 2, 3, 3, 4, 4\}$. We let this contradiction in the process of mapping happen because it will be eliminated later: no matter one instance $\mathcal{S}$ for weighted k-set packing results in *yes* or *no*, at most only one set of s'_1 and s'_2 is chosen, then we can safely delete the connections based on the deleted set from the CRN, and the contradiction is eliminated.

We have crossed the hurdle of finding one polynomial algorithm σ to transform instance of weighted k-set packing to an instance for clustering in CRN. Now we look into the step 2 in reduction.

When the instance $\mathcal{S}$ for weighted k-set packing contains one solution, i.e., there is a group of sets in $\mathcal{S}$, whose sum weights is greater than λ, then in the CRN which is mapped from $\mathcal{S}'$, the sum number of common channels of the clusters which correspond to the selected sets in $\mathcal{S}$ and $\mathcal{S}'$, is greater than λ. When there is no solution out of set $\mathcal{G}$ for weighted k-set packing, let's assume the maximum sum of weights of all instances is $\sum_{s_i \in \mathcal{S}} \omega(s_i) = \delta < \lambda$. The dummy set of each $s_i \in \mathcal{S}$ is mapped to cluster of CR nodes. Definition of CRN clustering regulates that the number of common channels is 0 when the cluster has only one node. As to $|s_i| = 1$, the mapped cluster has two nodes, with one of them is the dummy CR node. Then number of common channels is on longer 0 but equals to the weight of corresponding set s_i. Then the sum number of common channels of the clusters in CRN is $\delta < \lambda$, thus, there is no clustering solution for the mapped CRN.

After proving weighted k-set packing can be reduced to centralized clustering in CRN, we can say the latter problem is NP-hard. □

Acknowledgment

The authors would like to thank xxxx

PLACE
PHOTO
HERE

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

Erwin Fang Biography text here.

PLACE
PHOTO
HERE

James Gross Biography text here.

References

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

[2] Q. Zhao and B. Sadler, "A survey of dynamic spectrum access," *Signal Processing Magazine, IEEE*, vol. 24, no. 3, pp. 79–89, May 2007.

[3] A. Sahai, R. Tandra, S. M. Mishra, and N. Hoven, "Fundamental design tradeoffs in cognitive radio systems," in *Proceedings of the First International Workshop on Technology and Policy for Accessing Spectrum*, ser. TAPAS '06. New York, NY, USA: ACM, 2006.

[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, pp. 2511–2515.

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

[7] 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*.

[8] C. Passiatore and P. Camarda, "A centralized inter-network resource sharing (CIRS) scheme in IEEE 802.22 cognitive networks," in *Ad Hoc Networking Workshop (Med-Hoc-Net), 2011 The 10th IFIP Annual Mediterranean*, june 2011, pp. 17 –24.

[9] 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.

[10] 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*.

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

[12] Q. Wu, G. Ding, J. Wang, X. Li, and Y. Huang, "Consensus-based decentralized clustering for cooperative spectrum sensing in cognitive radio networks," *Chinese Science Bulletin*, vol. 57, no. 28-29, pp. 3677–3683, 2012.

[13] V. Kawadia and P. R. Kumar, "Power control and clustering in ad hoc networks," in *Proc. of INFOCOM '03*, 2003, pp. 459–469.

[14] C. R. Lin and M. Gerla, "Adaptive clustering for mobile wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 15, pp. 1265–1275, 1997.

[15] S. Basagni, "Distributed clustering for ad hoc networks," *Proc. of I-SPAN '99*, pp. 310 –315, 1999.

[16] J. Sucec and I. Marsic, "Clustering overhead for hierarchical routing in mobile ad hoc networks," in *Proc. of IEEE INFOCOM 2002*, vol. 3, 2002, pp. 1698 – 1706 vol.3.

[17] H. Wu and A. Abouzeid, "Cluster-based routing overhead in networks with unreliable nodes," in *Proceedings of IEEE WCNC 2004*, vol. 4, march 2004, pp. 2557 – 2562 Vol.4.

[18] T. Chen, H. Zhang, G. Maggio, and I. Chlamtac, "Cogmesh: A cluster-based cognitive radio network," *Proc. of DySPAN '07*, pp. 168 –178, apr. 2007.

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

[21] 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.

[22] L. Lazos, S. Liu, and M. Krunz, "Spectrum opportunity-based control channel assignment in cognitive radio networks," *Proc. of SECON '09*, pp. 1 –9, jun. 2009.

[23] 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.

[24] T. Yucek and H. Arslan, "A survey of spectrum sensing algorithms for cognitive radio applications," *IEEE Communications Surveys Tutorials*, vol. 11, no. 1, pp. 116–130, First 2009.

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

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

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

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

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

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

[31] E. Baccelli, J. A. Cordero, and P. Jacquet, "Optimization of critical data synchronization via link overlay rng in mobile ad hoc networks," in *In 7th IEEE International Conference Mobile Ad-hoc and Sensor Systems (MASS)*, 2010.

[32] 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.