Versatile Robust Clustering of Ad Hoc Cognitive Radio Network

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

Abstract—Cluster structure in cognitive radio networks facilitates cooperative spectrum sensing, routing and other functionalities. The 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.

1 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,

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

1. The term *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.

cluster size is one decisive factor in power preservation [10], [11], and it also influences the 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 *variants of ROSS*, i.e., the fast versions, or that with size control feature.

The rest of paper is organized as follows. After reviewing related work in section 2, we present our system model in Section 3. Then we introduce our clustering scheme ROSS and its variants in section 4. The clustering problem is given through analysis and a centralized scheme is proposed in section 5. Extensive performance evaluation is in section 6. Finally, we conclude our work and point out direction future research in section 7.

2 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 and sensor networks [9]. In ad hoc and mesh networks, the major focus of clustering is to preserve connectivity (under static channel conditions) or to improve routing. In 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].

3 System Model

We consider a set of cognitive radio users $\mathcal{N}$ and a set of primary users distributed on a two-dimensional Euclidean plane. These users share a number of non-overlapping licensed channels according to the spectrum overlay model. The set of these licensed channels is denoted as $\mathcal{K}$. As secondary users, the CR users are allowed to transmit on a channel $k \in \mathcal{K}$ only if no primary user is detected being accessing channel k. Further, we consider a *cognitive radio ad-hoc network* which consists of all secondary users and does not contain any primary user.

Secondary users conduct spectrum sensing independently and sequentially on all licensed channels. The sensing duration and frequency on one channel is a research topic [24], and we assume that every node can detect the presence of primary user on each channel with certain accuracy. [2] We denote $K_i \subseteq \mathcal{K}$ as the set of available channels for i.

We adopt the unit disk model [35] for the transmission of both primary and CR users. Both primary users and CR users have fixed transmission ranges respectively, and the all the channels are regarded to be identical in terms of signal propagation. If a CR node locates within the transmission range of primary user p, that CR node is not allowed to use the channel $k(p)$.

We assume that in addition to the licensed channels, there is one dedicated control channel. This control channel could be in ISM band or other reserved spectrum which is exclusively used for transmitting control messages. Actually, the control messages involved in the clustering process can be transmitted on available licensed channels through a rendezvous process by channel hopping [25], [26], i.e., two neighbouring nodes establish communication on the same channel. Over the control channel, a secondary user i can exchange its spectrum sensing result K_i to any $i' \in \mathrm{Nb}(i)$. It is available for any secondary node i to exchange control messages with any other node in its proximity (or neighborhood) $\mathrm{Nb}(i)$ during the cluster formation phase. $\mathrm{Nb}(i)$ is simply defined as the set of nodes located within the transmission range of i.

If a secondary user i is not in the transmission range of a primary user p, i can certainly not detect the presence of p. As

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

the transmission range of primary users is limited and secondary users are located at different locations, different secondary users may have different views of the spectrum availability, i.e., for any $i, i' \in \mathcal{N}$, $K_i = K_{i'}$ does not necessarily hold. As the assumed $0/1$ state of connectivity is solely based on the Euclidean distance between secondary users,

A cognitive radio network (CRN) can be represented as an undirected graph $G = (\mathcal{N}, E)$, where $E \subseteq \mathcal{N} \times \mathcal{N}$ such that $\{i, i'\} \in E$ if, and only if, there exists a channel $k \in \mathcal{K}$ with $k \in K_i \cap K_{i'}$. Note that we consider the channel availability only for *one* snapshot of time. For the rest of this paper the word channel is referred to licensed channel, if the control channel is not explicitly mentioned.

3.1 Clustering

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

———— No modifications from here on ... ————-

Cluster is denoted as C_i when its cluster head is i. We denote the set of common control channels for cluster C with set K_C. $K_C = \cap_{i \in C} K_i$ is the set of common control channels for cluster C. Clustering is performed periodicity, as secondary users are mobile and primary users change their working channels or move.

TABLE 1
Notations in robust clustering problem (*when sub*)

Symbol	Description
$\mathcal{N}$	collection of secondary users
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 CCCs of cluster $C(i)$
h_C	the cluster head of a cluster C
δ	desired cluster size
S_i	a set of claiming clusters, each of which includes debatable node i after phase I
d_i	individual connectivity degree of CR node i
g_i	social connectivity degree of CR node i

4 Distributed Coordination Framework: Clustering Algorithm

In this section, we present the clustering scheme ROSS. ROSS utilizes the proximity 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.

4.1 Phase I - Cluster Formation

After conducting spectrum sensing and communication with neighbours, every CR node is aware of the available channels at itself and all its neighbours. For each CR user, two metrics are proposed to characterize the spectrum proximity between the CR user and its neighborhood. As to CR node i, there are,

- *Individual connectivity degree* d_i: $d_i = \sum_{j \in \mathrm{Nb}(i)} |K_i \cap K_j|$, 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: $g_i = |\bigcap_{j \in \mathrm{Nb}(i) \cup i} K_j|$, 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.

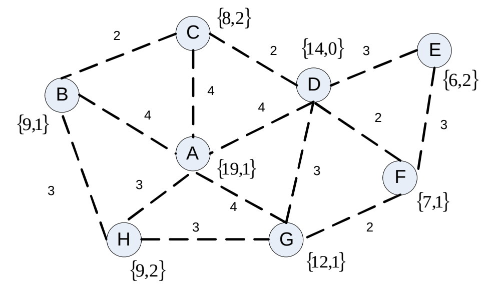


Fig. 1. Connectivity graph and the connectivity vector (d_i, g_i) at each node. The available channels sensed by each CR node are: $K_A = \{1,2,3,4,5,6,10\}, K_B = \{1,2,3,5,7\}, K_C = \{1,3,4,10\}, K_D = \{1,2,3,5\}, K_E = \{2,3,5,7\}, K_F = \{2,4,5,6,7\}, K_G = \{1,2,3,4,8\}, K_H = \{1,2,5,8\}$. Dashed 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.

After introducing connectivity vector, we proceed to introduce the first phase of algorithm ROSS. At first, cluster heads are determined, then clusters are formed on the basis of the cluster heads' neighbourhoods.

4.1.1 Determining Cluster Heads and Form the Initial Clusters

In this phase, each CR node decides whether it is cluster head by comparing its connectivity vector with its neighbours. When CR node i has bigger individual connectivity degree than any neighbours except for those which have already become cluster heads (the appearance of cluster heads will be explained in next subsubsection, they are denoted as CHs), then node i becomes clusters head. 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 $d_j > d_k, \forall k \in \mathrm{Nb}(j) \setminus \{\mathrm{CHs} \cup i\}$, 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. The pseudo code for deciding cluster head and forming initial clusters is in Algorithm 1 in appendix.

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. We manipulate the individual connectivity degree of the CR nodes, so that those locating outside of the cluster are possible to become cluster heads, then every CR node becomes either cluster head or one member of at least one cluster. 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 4.1: *Given a CRN, every secondary user is included into at least one cluster within N steps.*

The Proof is in Appendix 19. According to Theorem 4.1, we can assign reasonable amount of time for phase I to complete.

Let us apply Algorithm 1 to the example shown in Figure 1. Node B and H have the same individual connectivity degree, $d_B = d_H$, but as $g_H = 2 > g_B = 1$, node H becomes cluster head. Cluster C_H is $\{H, B, A, G\}$.

4.1.2 Guarantee Availability of Common Control Channel

After deciding itself being cluster head, CR node broadcasts to notify its neighbours on the dedicated 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., $C_i = (\mathrm{Nb}_i \backslash \mathrm{CHs}) \cup i$. Note this is the initial cluster, as it is possible for the formed cluster to pose no 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 control channel. The elimination of nodes is performed according to an ascending list of nodes which are sorted by their number of common channels with the cluster head. In other words, 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 restore their original individual connectivity degrees, and become either cluster heads or get included into other clusters later on according to Theorem 4.1.

4.1.3 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. When cluster size is large, there will be substantial burden on cluster heads to manage the cluster members, which is a challenge for resource limited cluster heads, thus 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 formed clusters 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 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 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, which is the product of the number of cluster heads in row and line, $l/r * l/r = l^2/r^2$.

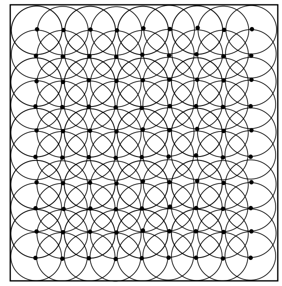


Fig. 2. Clusters formation in extremely dense CRN. Black dots are cluster heads, other cluster members are not drawn.

Now we verify the estimation with simulation. We distribute CR and primary users randomly on a square plain, r=10 and l=50. Network density is increased by adding more CR users. The number of formed clusters is shown in Figure 3. With the increase of CR users in the network, network density increases linearly (the Y axis label is 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 and the confidence interval is 95%.

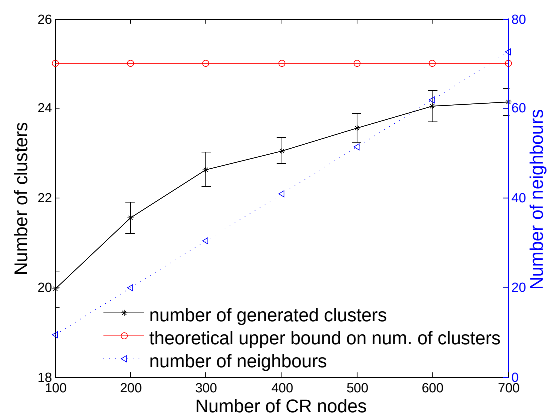


Fig. 3. The correlation between the number of formed clusters and network density. Note that the number of neighbours denotes the network density.

Both the analysis and simulation show that when applying ROSS, the cluster size increases with the increase of network density. As a result, certain measures are needed to prevent the network size from increasing with the increasing network density. This task falls to the cluster heads. To control cluster size, cluster heads prune their cluster members. The desired size δ is decided based on the capability of the CR users and the tasks to be conveyed. Given desired size as δ, cluster head excludes members sequentially, whose absence leads to the maximum increase of common channels within the cluster.

Note that ROSS generates clusters on based of cluster heads' neighbourhood, thus the δ is smaller than the average neighbourhood size. As there are extensive overlaps between clusters, the threshold that the cluster size satisfies requirement should be larger than δ. 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 can reuse Algorithm 2.

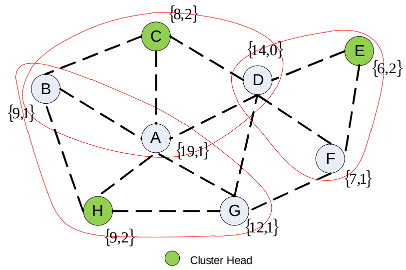


Fig. 4. Clusters formation after the first phase of ROSS. There are some nodes being debatable nodes, i.e., belonging to more than one cluster.

4.2 Phase II - Membership Clarification

After applying phase I of ROSS to the example in Figure 1, the resulted clusters are shown in Figure 4. We notice nodes A, B, D are included in 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 when the CRN network scales up.

Debatable nodes should be exclusively associated with only 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.

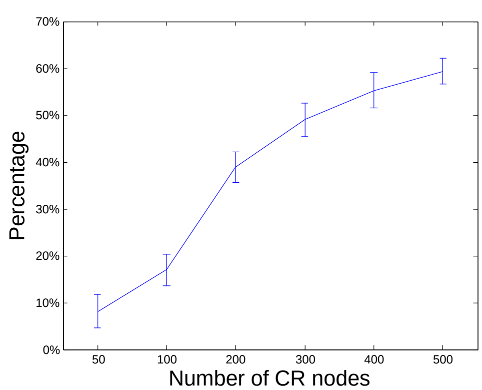


Fig. 5. The percentage of debatable nodes after phase I of ROSS.

4.2.1 Distributed Greedy Algorithm

After Phase I, debatable nodes, e.g., i needs to decide which cluster $C \in S_i$ to stay, and thereafter leaves the rest others in S_i. 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 of its claiming clusters, and is aware of the vector of common channels of the claiming clusters, then i is able to know how many more common control channels will be in one certain claiming cluster if i leaves that cluster. Based on this calculation, i decides on one claiming cluster to stay and leaves the other claiming clusters. If there exists one cluster $C \in S_i$, if i leaves this cluster brings the least increased common control 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 claiming cluster with smallest size. IDs of cluster heads will be used to break tie if the previous rule could not 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., K_C (the set of available channels in C), K_{h_C} (the set of available channels on C's cluster head h_C) and $|C|, C \in S_i$ (sizes of i's claiming clusters). After deciding which cluster to stay based on Algorithm 3, node i notifies all its claiming clusters, and retrieves the updated information of the necessary information K_C, K_{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. Thence, 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 is reached after a finite number of best response updates.

4.2.2 Bridging ROSS-DGA with Congestion Game

To formulate the problem of membership clarification for the debatable nodes in the context of a game, we observe this process from a different (or opposite) perspective. From the new perspective, the debatable nodes are regarded as isolated and don't belong to any cluster, which means their claiming clusters become their neighbouring clusters. Then for debatable nodes, the previous problem of deciding 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. The concern on convergence appears again as we have discussed in the previous subsubsection. We will give proof on convergence under game theoretic framework.

Game theory is a powerful mathematical tool for studying, modelling and analysing the interactions among individuals. A

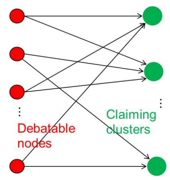


Fig. 6. Debatable nodes and claiming clusters

game consists of three elements: a set of players, a selfish utility for each player, and a set of feasible strategy space for each player. In a game, the players are rational and intelligent decision makers, which are related with one explicit formalized incentive expression (the utility or cost). Game theory provides standard procedures to study its equilibriums [29]. In the past few years, game theory has been extensively applied to problems in communication and networking [30], [31]. Congestion game is an attractive game model which describes the problem where participants compete for limited resources in a non-cooperative manner, it has good property that Nash equilibrium can be achieved after finite steps of best response dynamic, i.e., each player choose strategy to maximizes/minimizes its utility/cost with respect to the other players' strategies. Congestion game has been used to model certain problems in internet-centric applications or cloud computing, where self-interested clients compete for the centralized resources and meanwhile interact with each other. For example, server selection is involved in distributed computing platforms [32], or users downloading files from cloud, etc. In the following we will introduce an *server matching* [33] problem to illustrate congestion game's application in communication systems.

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 can be represented by a 4-tuple $\Gamma = (\mathcal{P}, \mathcal{R}, \sum_{i,i\in\mathcal{P}}, f)$, and the elements in Γ are explained below,

- $\mathcal{P}$, the set of players of the game, which are the debatable nodes after phase I of ROSS.
- $\mathcal{R} = \cup S_i, i \in \mathcal{P}$, 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.
- Strategy space $\sum_i, i \in \mathcal{P}$ is the set of claiming clusters S_i. As debatable node i is supposed to choose one claiming cluster in our problem, only one piece of resource is allocated for i.
- The utility (cost) function f as to resource $C \in \mathcal{R}$ is $\Delta|K_C^i|$, which represents the decrement of CCCs in cluster C when debatable node i' joins in it. As to cluster $C \in S_i$, the decrement of CCCs caused by the 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.

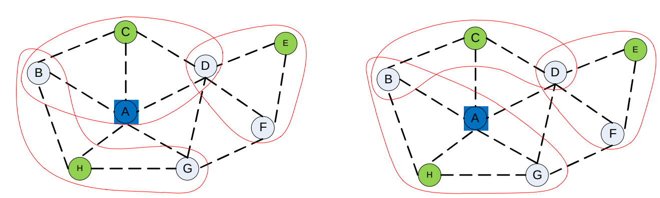
  The utility function is not purely decided by the number of players (debatable nodes) as that in a canonical congestion game, because in this game the channel availability on debatable nodes is different. Given two same sized groups of debatable nodes, when the nodes are not completely the same (neither 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, as described by Algorithm 3.

As to singleton congestion game, there exists pure equilibria which can be reached with best response update, and the upper bound of number of steps before convergence is $n^2 * m$ [34], 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 upper bound of the number of steps can be expressed as $O(N^3)$. In fact, the actual number of steps is much smaller than N^3 as both n and m are considerably smaller than N. 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 4.1, 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.

4.2.3 Distributed Fast Algorithm (DFA)

We propose a faster version of ROSS, which is ROSS-DFA, which is especially suitable for CRN where channel availability change dynamically and re-clustering is necessary. In DFA, debatable nodes regard their claiming clusters includes all their debatable nodes, thus the membership of claiming clusters is static and debatable nodes can make decisions simultaneously without considering the change of membership of their claiming clusters. To run ROSS-DFA, debatable node executes only one loop in Algorithm 3.

Now we apply the two versions of ROSS to the toy network in Figure 4, which has finished cluster head selection. In the network, node A's claiming clusters are cluster $C(C), C(H) \in S_A$, their members are $\{A, B, C, D\}$ and $\{A, B, H, G\}$ respectively. The two possible strategies of node A'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

5 Centralized Clustering Scheme

The centralized clustering scheme aims to form clusters with desired sizes, meanwhile the total number of common control channels of all clusters is maximized. In the following, we refer

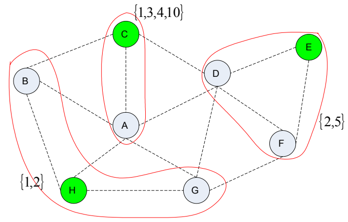


Fig. 8. Final formation of clusters, CCCs for each cluster is shown.

$$
Q = \begin{array}{c} \\ 1 \\ 2 \\ \vdots \\ i \\ \vdots \\ \vdots \\ G \end{array}
\begin{array}{c}
\begin{array}{cccccccc} 1 & 2 & 3 & \cdots & j & \cdots & N-1 & N \end{array} \\
\begin{pmatrix}
k_{C_1} & k_{C_1} & 0 & \cdots & \cdots & \cdots & 0 & 0 \\
k_{C_2} & 0 & k_{C_2} & \cdots & \cdots & \cdots & 0 & 0 \\
\vdots & & & \vdots & & \vdots & & \\
0 & k_{C_i} & 0 & \cdots & \cdots & \cdots & k_{C_i} & 0 \\
\vdots & & & \vdots & \cdots & \cdots & & \vdots \\
\vdots & 0 & 0 & \cdots & \cdots & \cdots & k_{C_i'} & 0 \\
k_{C_G} & \cdots & & \vdots & \cdots & \cdots & & \vdots
\end{pmatrix}
\end{array}
$$

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

this problem as *centralized clustering*, 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 the size of cluster C and k is a 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 that cluster), $\mathcal{S}$ satisfies the following properties: $\bigcup_{1\le i\le|\mathcal{S}|} C_i = \mathcal{N}$ and $K_{C_i} \neq \emptyset$ for any i which satisfies $1 \le i \le |\mathcal{S}|$.

Following condition distinguishes the centralized clustering problem discussed in this paper. The number of common control channels is $|K_C|$ if $|K_C| > 1$, and is zero when $|K_C| = 1$. We use f to denoted this new number of CCCs. The question of this problem is to find a subcollection $\mathcal{S}' \subseteq \mathcal{S}$, so that $\bigcup_{C_j\in\mathcal{S}'} C_j = \mathcal{N}$, 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.

5.1 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.

Theorem 5.1: *CRN clustering problem is NP-hard, when the maximum size of clusters $k \geqslant 3$.*

The proof is in Appendix 19.

5.2 Centralized Optimization

As there is no efficient algorithm to solve clustering problem in CRN, we propose a centralized optimization where the objective function and the constraints are heuristic, then we adopt binary linear programming to solve the problem.

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 3.1. 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 constant $G \times N$ 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, which can be seen as the node ID. 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. 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 $j \notin C_i$. In other words, each non-zero element q_{ij} denotes the number of CCCs of the cluster i where node j resides.

We build a $G \times N$ binary variable 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 in the clustering solution, $x_{ij} = 0$ means this partition is not adopted. 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. We notice that the first part is the metric adopted by the scheme SOC [22]. The second part is the *punishment* for choosing the clusters whose sizes are not δ. 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 does the mechanism of the punishment in the objective work. 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 defined as follows,

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

where $\alpha_i > 0$ 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.

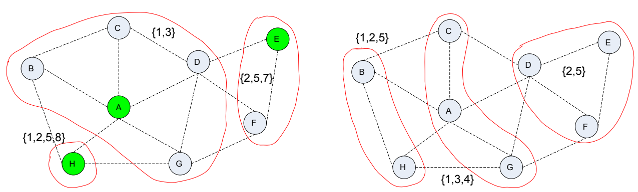
This is a integer linear optimization problem, which is solved by function *bintprog* provided in MATLAB. Note that the proposed centralized solution is heuristic. There are two reasons for pursuing the heuristic scheme, first, the problem of centralized clustering is NP hard, and there is no efficient 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.

5.2.1 Example of the Centralized Optimization

We look into how it the centralized scheme perform in 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 3.1 and the sizes of clusters are 1, 2 or 3. $\mathcal{G} = \{\{A\}, \{B\}, \ldots, \{B, C\}, \{B, A\}, \{B, H\}, \cdots, \{B, A, C\}, \{B, H, C\}, \{A, D, C\}, \cdots\}$, $G = |\mathcal{G}| = 38$. α_1 and α_1 are set as 0.2 and 0.8. The clustering result of centralized programming is shown in Figure 10(b).

The formed clusters are $\{\{D, E, F\}, \{A, C, G\}, \{H, G\}\}$, the numbers of CCCs are 2, 3, 3. The resulted clustering solutions from ROSS-DGA/DFA and SOC are shown in Figure 8 and Figure 10(a) respectively.As to the average number of CCCs, the results of ROSS, centralized and SOC are 2.66, 2.66, and 3 respectively. Note there is one singleton cluster C_H generated by SOC, which is not preferred. When the singleton cluster $\{E\}$ is excluded, the average number of common channels of SOC drops to 2.5.



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

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

6 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 features. The latter is referred as ROSS-x-DGA/ROSS-x-DFA, where x is the desired cluster 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. In particular, 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 in a 2-dimensional Euclidean space. The number of licensed channels is 10, each PU is operating on each channel with probability of 50%. All primary and CR users are assumed to be static during the process of clustering.

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. The simulation of this part is conducted in a small network, as there is no polynomial time solution available to solve the centralized problem. In the second part, we investigate the performance of the proposed distributed schemes thoroughly in the networks with different scales and densities.

6.1 Centralized Schemes vs. Decentralized Schemes

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 4.8. 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. 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.

6.1.1 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 1. 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.

6.1.2 Survival Rate of Clusters with Increasing Primary Users

With the number of PUs 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 PUs, extra 100 PUs are added sequentially to the network. Figure 12 shows the number of unclustered CRs with the increasing number of PUs.We draw three conclusions corresponding to three comparisons shown in this figure,

- Centralized scheme with cluster size of 2 generates the most robust clusters, and SOC results in the most vulnerable clusters. When the desired cluster size is 3, the centralized scheme performs similarly with the variants of ROSS. 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 PUs' activities become intense. The comparison on cluster sizes will be given in details in 6.1.3.
- Greedy algorithm improves survival rate. ROSS-DGA improves the survival rate of ROSS-DFA, so does ROSS-x-DGA against ROSS-x-DFA. This complies with the observation in Figure 11. As the debatable CRs greedily update their affiliation with claiming clusters, and the metric for updating is the maximum increase of CCCs of the demanding clusters, the average number of CCCs in non-singleton clusters is improved.
- ROSS with size control is better than the other two distributed schemes. Conducting size control improves both ROSS-DGA and ROSS-DFA's performance when the number of new PUs is greater than 50. The size control decreases the clusters size and makes the clusters more robust when against PUs' activity.

6.1.3 Cluster Size Control

Figure 13 depicts the empirical cumulative distribution of the CRs residing in certain sized clusters in 50 runs. The centralized schemes are able to form clusters which satisfy the requirement on cluster sizes strictly. When the desired size is 2, each generated cluster has two members, whereas when the desired size is 3, about 15% CRs are formed into 2 node clusters. Both of ROSS-DFG and ROSS-DFA with size control feature also obtain clusters with homogeneous sizes. The sizes of clusters generated by ROSS-DGA and ROSS-DFA are disperse, but appear better than SOC, i.e., the 50% percentiles for ROSS-DGA, ROSS-DFA and SOC is 4.5, 5, and 5.5, and the 90% percentiles for the three schemes are 8, 8, and 9. Note ROSS-DGA and ROSS-DFA with size control feature generate 10%-20% singleton clusters, which is due to the cluster pruning discussed in section 4.1.3, whereas, without size control, only 3% nodes are in singleton clusters. When applying SOC, 10% of nodes are in singleton clusters.

6.1.4 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 [36], 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 Synonymous with *the number of updates* discussed in Section 4.

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 4.1 and Section 4.2.2 respectively.

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

As to the centralized scheme, except for the calculation in the centralized control device, it also involves two phases of control message transmission. The first phase is information aggregation, in which every CR node's channel availability and neighbourhood is sent to the centralized controller. The second phase is broadcasting, where the final clustering solution is disseminated over the network to every CR node. We adopt the algorithm proposed in [37] to broadcast and gather information as the algorithm is simple and self-stabilizing. This scheme needs building a backbone structure to support the communication, we use our generated cluster heads as the backbone and the debatable nodes as the gateway nodes between the backbone nodes. As the backbone is built once and can support transmission for

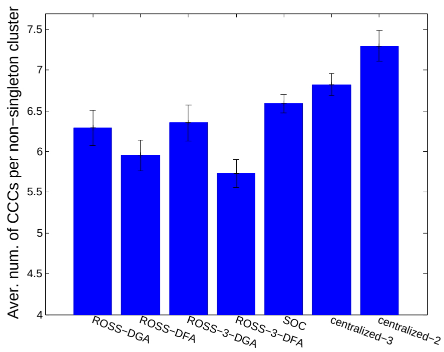


Fig. 11. Number of common channels for non-singleton clusters

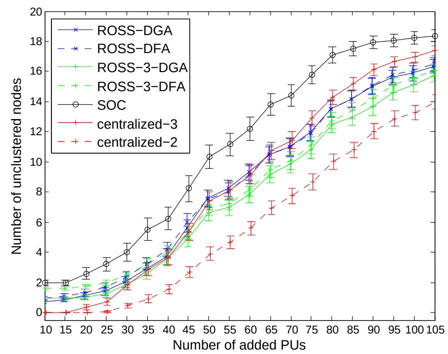


Fig. 12. Number of CRs which are not included in any clusters

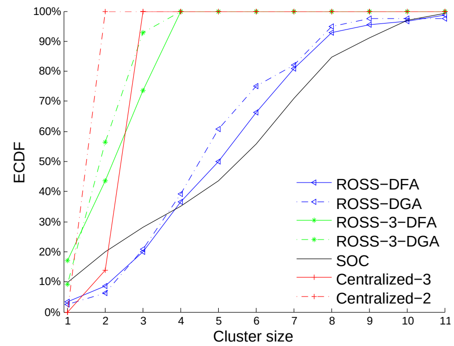


Fig. 13. Cumulative distribution of CRs residing in clusters with different sizes

Fig. 14. Comparison between the distributed and centralized clustering schemes in a small network (N = 20)

multiple times, the messages involved in the clustering process are not included. Every CR node can be informed if the cluster heads broadcast the message sent from the controller or other cluster heads, then the number of transmission is $h + m$. As to information gathering, we assume that every cluster member sends the spectrum availability and its ID to its cluster head, which further forwards or the message to the controller. The number of transmission for information gathering is N. The number of transmission is depicted in Figure 15. Note that the curve for the centralized scheme is from theoretical analysis, which is $h+m+N$ as discussed beforehand. Note the overhead involved to construct the backbone (clusters) is not included.

The message complexity, quantitative analysis of the number of messages, and the size of control messages are shown in Table 2. Some notations are written as follows, h: number of cluster heads, m: number of debatable nodes, d: number of demanding clusters, $D(s)$: the maximum distance between centralized controller and CR users.

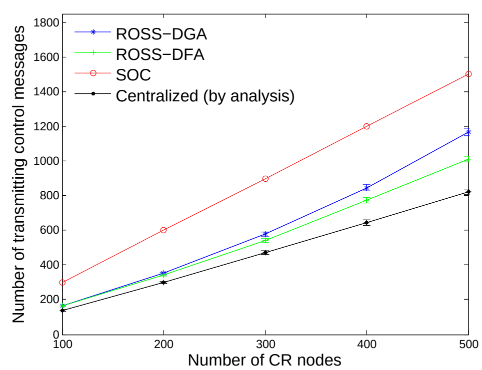


Fig. 15. Number of control messages, note the curves for ROSS-DGA and ROSS-DFA are the upper bounds of the number of messages, the curve of centralized scheme reflects an ideal situation.

6.2 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. We list some parameters of the simulation in the Table 3.

TABLE 3

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

6.2.1 Number of CCCs per Non-singleton Clusters

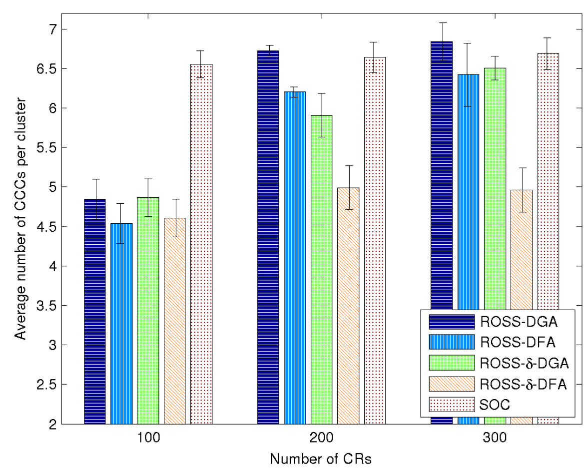


Fig. 16. Number of common channels of non-singleton clusters.

Figure 16 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 in terms of the average number of CCCs per non-singleton clusters when network is sparse. this is also observed in the evaluation in Section 6.1.1 where N = 20. When the network becomes denser, even this metric favours SOC as discussed in the beginning of Section 6, ROSS-DGA achieves even more CCCs than SOC, and ROSS-DFA and ROSS-x-DGA increase the number of CCCs visibly.

TABLE 2
Singalling overhead.

Scheme	Message Complexity	Quantitative number of messages	Content of message
ROSS-DGA, ROSS-x-DGA	$O(N^3)$ (worst case)	$h + 2 * m^2 d$ (upper bound)	Phase I: notification from cluster head (1 byte), new individual connectivity degree (1 byte); Phase II: update of debatable nodes' affiliation (1 byte), claiming clusters' new membership ($\|C_i\|$ bytes)
ROSS-DFA, ROSS-x-DFA	$O(N)$ (worst case)	$h + 2m$ (upper bound)	
SOC	$O(N)$	$3 * N$	C_i ($\|C_i\|$ bytes), K_i (P bytes), $i \in \mathcal{N}$
Centralized	$O(N)$	$h + m + N$ (upper bound) [37]	$\{C\}$ ($\|C_i\| * N$ bytes)

6.2.2 Survival Rate of Clusters with Increasing Primary Users

In this part of simulation, we investigate the robustness of clusters by increasing the PUs working on certain channels.

Figure 17 illustrates the increasing trend of singleton clusters with the increase of PUs. SOC generates around 10 more singleton clusters than the variants of ROSS, which accounts for 10% of the total CR nodes. We only show the average values of the variants of ROSS as their confidence intervals overlap. Figure 18 depicts a denser CRN where $N = 300$. SOC noticeably causes more singleton clusters than ROSS variants, except that ROSS-20-DFA results in more singleton clusters when PUs are few. The reason is ROSS-20-DFA conducts cluster membership clarification for only once, which causes large number of singleton clusters. ROSS-20-DGA increases the size of smaller clusters through debatable nodes' repeated updates thus drastically decreases the number of singleton clusters.

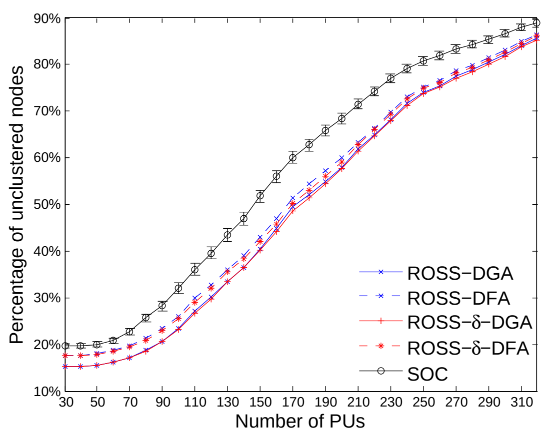


Fig. 17. Percentage of CRs which are not included in any clusters with the increasing number of primary users, $N = 100$

From the Figure 17 and 18, we can conclude that the greedy versions of ROSS are slightly more robust than their counterpart variants of ROSS, and the clusters obtained from variants of ROSS are clearly more robust than SOC.

6.2.3 Cluster Size Control

The number of formed clusters is shown in Fig. 19. When the network scales up, the number of formed clusters by ROSS increases by smaller margin. This result coincides with the analysis in Section 4.1.3, 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,

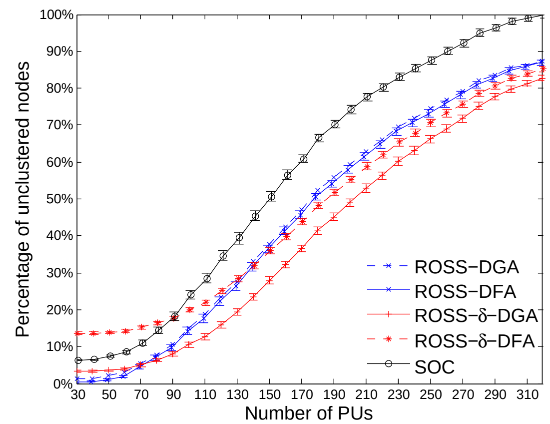


Fig. 18. Percentage of CRs which are not included in any clusters with the increasing number of primary users, $N = 300$

we depict the cluster sizes with cumulative distribution. In this group of evaluation, the number of PRs is 30.

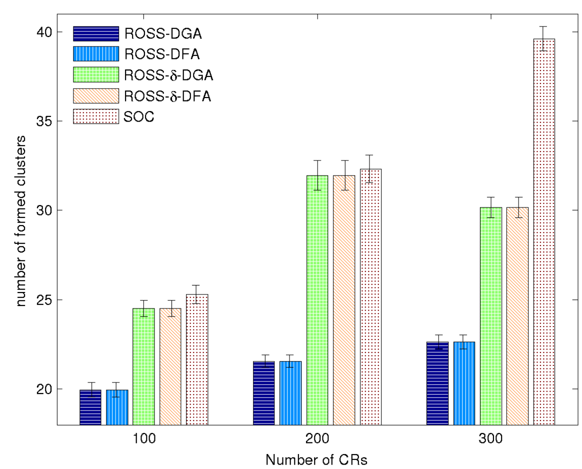


Fig. 19. The number of formed clusters.

Figures 20 21 22 illustrate the empirical cumulative distribution of CR nodes which reside in clusters with certain sizes in CRNs with different densities. When the variants of ROSS with size control feature are applied, the sizes of the most generated clusters are below δ, and most of them are around the 50% percentile. The sizes of clusters generated by ROSS-DGA and ROSS-DFA span a wider range than that with feature control feature. We find that average number of neighbours is roughly equal with the 95% percentile of the ROSS-DGA curve. As to SOC, the 95% percentiles are 36, 30, and 40. Overviewing the three Figures, we can see ROSS-DGA and ROSS-DFA show similar behaviour on cluster sizes. The clusters generated from SOC demonstrate strong divergence on cluster sizes.

7 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 control overhead.

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

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

if $\nexists j \in Nb_i \setminus CHs$, *such that* $d_i \le d_j$ **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* $g_i \le g_j$ **then**
  return 1;
end
if $\exists j \in \tau_1$, *such that* $g_i < g_j$ **then**
  return 0;
else
  if $\nexists j \in \tau_1$, *such that* $g_j == g_i$ **then**
    $\tau_2 \leftarrow j$
  end
end
if ID_i *is smaller than any* ID_j, $j \in \tau_2 \setminus i$ **then**
  return 1;
end
return 0;

Proof of Theorem 4.1

Proof. Note the formed cluster can be a singleton cluster, i.e., cluster size is 1. To simplify the discussion, 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 of this kind. As node α is not contained in any cluster, there must be at least one node $\beta \in \mathrm{Nb}_\alpha$, 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 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 24.

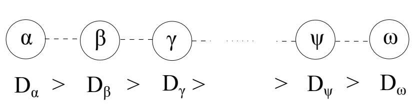


Fig. 24. The node series discussed in the proof for Theorem 4.1, the deduction begins from node α

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

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

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

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

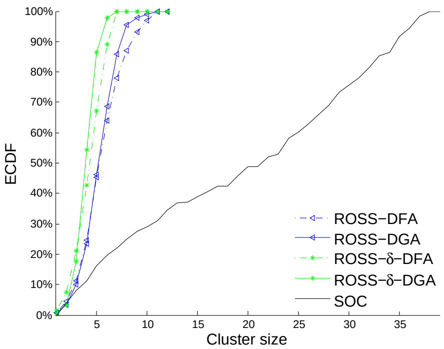


Fig. 20. 100 CRs, 30 PRs in network

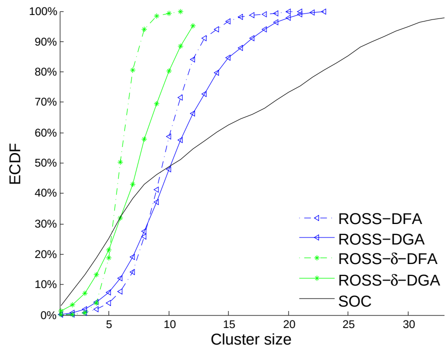


Fig. 21. 200 CRs, 30 PRs in network

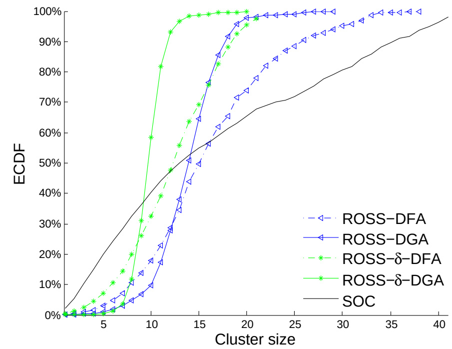


Fig. 22. 300 CRs, 30 PRs in network

Fig. 23. Cumulative distribution of CRs residing in clusters with different sizes

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

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

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

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 5.1

Proof. 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 $k \geqslant 3$. [38]

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

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

We continue to use the notation adopted in the problem definition in Section 5.2. Let set $\mathcal{G}$ contain N natural numbers which are from 1 to N. $\mathcal{Q}$ is a collection of sets $\{s_1, s_2, \cdots s_m\}$, each set is composed with certain elements in $\mathcal{G}$. Assume $\mathcal{S} \subseteq \mathcal{Q}$ is one instance of weighted k-set packing, and the sets in $\mathcal{S}$ are disjoint. ω indicates the weight for each set s, $\omega : \mathcal{S} \to \mathbb{Z}^+$.

The polynomial algorithm σ consists of three transformations.

- Given a collection $\mathcal{Q}$, on basis of which we construct a CRN. We prepare N CR nodes who are labelled from 1 to N. We put the CR users on a 2 dimension space, and deem a pair of them can communicate if they appear in the same set in $\mathcal{Q}$. We regard each set in $\mathcal{Q}$ is a cluster, whose number of CCCs equals to the weight of that set.
  Assuming two sets in $\mathcal{Q}$ are $s_1 = \{1, 2\}$ and $s_2 = \{1, 2, 3\}$, then their weights are 3 and 5 respectively. We find it is impossible to map the sets into clusters in the same time, because the number of CCCs of the cluster which bases on s_1 should be no less than the cluster which bases on s_2, as the latter has one extra node compared with the former cluster. But as to any instance of the solution to the weighted k-set packing problem, this contradic-

tion doesn't happen because the instance $\mathcal{S}$ contains only disjoint sets, thus at most only one set of s_1 and s_2 appears in $\mathcal{S}$.

- In the second step, we transform the instance $\mathcal{S}$ to $\mathcal{S}'$ by adding dummy elements into each set in $\mathcal{S}$. For each set $s_i \in \mathcal{S}$, the elements in s_i are duplicated, for instance, given $s_i = \{1, 4, 6\}$, the dummy set s_i' is $\{1, 1, 4, 4, 6, 6\}$. The purpose of this transformation is to eliminate the set in $\mathcal{S}$, which has single element. The weight of set is unchanged after this transformation, i.e., $\omega(s_i) = \omega(s_i')$. This transformation requires $\sum_{s_i \in \mathcal{S}} |s_i|$ steps.
- In this step, we transform the instance $\mathcal{S}'$ to a clustering solution for CRN. We prepare a second pool of CR nodes which are identical with the CR nodes prepared in step 1, i.e., identical IDs and channel availabilities on them, we call these CR nodes as dummy nodes. We locate these CR nodes besides the CR nodes with the sane IDs in the CRN built in step 1, and there is connection between the CR node and its dummy node (the one CR node and its dummy node can be seen as two transceivers at one node). Because of the dummy nodes, the clustering solution which corresponds to $\mathcal{S}'$ doesn't have singleton cluster. This transformation requires $2 \cdot \sum_{s_i \in \mathcal{S}} |s_i|$ steps. 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 CCCs in these single node clusters is 0. These singleton clusters and the clusters in $\mathcal{S}$ constitute a clustering solution, and finding the singleton clusters requires at most N steps.

An example is shown in Table 4.

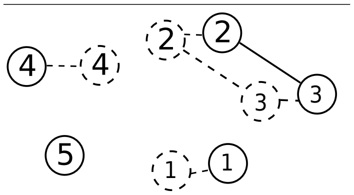
$\mathcal{N}$	$\{1, 2, 3, 4, 5\}$
$\mathcal{Q}$	$\{(1), (1, 5), (1, 2, 4), (2, 3), (4)\}$
Instance for Weighted k-set packing	$\{(1), (2, 3), (4)\}$
Instance with dummy elements	$\{(1, 1), (2, 2, 3, 3), (4, 4)\}$
Instance for clustering solution (dashed circles are dummy nodes)	

TABLE 4

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.

As to an instance $\mathcal{S}$ for weighted k-set packing, the sum weights is identical to the sum of CCCs in the CRN mapped from $\mathcal{S}'$, even $\mathcal{S}$ contains set which only has one element. Thus, when the instance $\mathcal{S}$ is one solution and its sum weights is greater than λ, in the CRN which is mapped from $\mathcal{S}'$, the summed number of CCCs of the clusters is greater than λ. When there is no solution out of set $\mathcal{G}$ for weighted k-set packing, the summed number of CCCs of the clusters in the mapped CRN is also smaller than λ.

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

Acknowledgment

The authors would like to thank xxxx

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

PLACE PHOTO HERE

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

PLACE PHOTO HERE

James Gross Biography text here. Biography text here. Biography text here. Biography text here. Biography text here. Biography text here. Biography text here. Biography text here. Biography text here. 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 *Proc. of the First International Workshop on Technology and Policy for Accessing Spectrum*, ser. ACM TAPAS ’06.

[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 *Proc. of IFIP Annual Mediterranean Ad Hoc Networking Workshop 2011*, 2011.

[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*.

[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*.

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

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

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

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

[33] A. Kothari, S. Suri, C. D. TÃşth, and Y. Zhou, “Congestion games, load balancing, and price of anarchy,” in *Proc. of CAAN’04*.

[34] 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.

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

[36] 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.

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

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