A Clustering-based Coordinated Spectrum Sensing in Wideband Large-scale Cognitive Radio Networks

Behzad Shahrasbi and Nazanin Rahnavard
School of Electrical and Computer Engineering, Oklahoma State University, Stillwater, OK 74078
Emails: {behzad.shahrasbi, nazanin.rahnavard}@okstate.edu

***Abstract*—Efficient spectrum sensing is one of the key features that allows the implementation of fully agile cognitive radio networks. In this paper, we present an efficient coordinated spectrum sensing algorithm for wideband large-scale cognitive radio networks. Our approach is based on clustering secondary users according to their spectrum sensing results and performing the spectrum sensing tasks collaboratively within each cluster. In addition, the clusters can collaborate with each other to achieve an optimal distributed spectrum sensing across the network. We set up a cognitive radio framework and evaluate our proposed algorithm using numerical simulations. We show that the proposed algorithm increases the successful channel sensing rate at a reasonable computational cost.**

I. Introduction

Cognitive radio (CR) is a promising solution to alleviate today's spectrum deficiency caused by an increased demand for the wireless technologies [1]. Therefore, CR was proposed to mitigate the under-utilization of the spectrum and to make spectrum allocation more efficient [1, 2]. According to the CR paradigm, in addition to the existing licensed users (a.k.a. primary users (PUs)) of the spectrum, a new type of users called unlicensed users or secondary users (SUs) is defined. These users are allowed to access the spectrum given that they do not interfere with the licensed users. The under-utilized spectrum bands that can be used by the SUs are called spectrum holes [2].

The ideal CR can efficiently detect and utilize spectrum holes. *Spectrum sensing*, which is responsible for finding the spectrum holes, is one of the key tasks in cognitive radio networks (CRNs). Finding more spectrum holes means more opportunities for SUs to transmit their own data. Due to the unknown activities of PUs, SUs should periodically sense the entire wideband spectrum during the very short time available for sensing [3]. To tackle this problem, *coordinated spectrum sensing* (CSS) was proposed [4–6]. In CSS, spectrum is divided into several narrow subbands or channels, and a central unit (a.k.a. the base station (BS)) assigns each SU a unique channel to sense in each sensing time slot (the time is divided into frames and each frame consists of sensing and access slots). In other words, different SUs will be assigned to sense different channels. Accordingly, several channels will be sensed by the SUs at one sensing time slot, and the sensing results (whether the sensed channel is *busy* or *empty*) will be sent to the BS. It has been shown that CSS significantly increases sensing efficiency [4–6]. In [6], we took a graph-theoretic approach to solve the CSS problem in a centralized way by finding a one-to-one matching between SUs and PU channels (as vertices of a bipartite graph) at each sensing time.

This material is based upon work supported by the National Science Foundation under Grant No. ECCS-1056065 and CCF-0915994.

In large-scale cognitive radio networks (CRNs), such as wireless regional area networks (WRANs), a channel may be busy in one location while it is empty in another location within the network. Such conflicting channel sensing results hinder the CSS capability to determine the spatial spectrum holes and thus degrade the performance of CSS. Similar to the case of ad-hoc networks, CRNs can be scaled down by clustering the SUs [7]. Clustering the SUs enhances the performance of spectrum sensing in CRNs. This is because the SUs that are far apart most likely have uncorrelated sensing results, and they will be assigned to different clusters. Therefore, they sense the PUs' activities from two distant locations, which helps to determine the locally available spectrum holes. In CRNs, clustering has been used to increase the sensing reliability [7–10] by assigning the SUs within a cluster to sense the same channel. In addition, clustering is employed to reduce the network management traffic [11, 12]. Nevertheless, no study has focused on the clustering issues in CSS.

Our contributions in this paper can be summarized as follows. First, we introduce a novel *metric* for clustering SUs in a distributed CRN. This metric is based on the similarities in the SUs' previous channel sensing results. Using this metric in conjunction with a low-complexity clustering algorithm based on *highly connected subgraphs* (HCS) enables the BS to efficiently form the clusters without the need to know the location of the SUs. Second, within each cluster, we propose to employ a graph-theory-inspired CSS procedure based on our previous work [6]. Our proposed scalable algorithm significantly increases the rate of successful channel sensing.

The rest of this paper is organized as follows. In Section II, we introduce the PU-SU coexistence in the spatially distributed CRNs and we define the spectrum sensing efficiency factor for CRNs. In Section III, we propose the clustering algorithm for spectrum sensing using the highly connected subgraph selection method from graph theory literature. Section IV contains our proposed coordinated spectrum sensing for spatially distributed CRNs. In Section V, we provide the simulation results and the performance evaluation of the proposed algorithm. Finally, Section VI concludes the paper.

II. Wideband Spectrum Sensing in CRNs

Establishing the coexistence of PUs and SUs is the most important and challenging aspect of CRNs. Since PUs have a higher priority to access the available channels, SUs should constantly monitor PU channels to find the transmission opportunities for themselves while avoiding interference with the incumbent PUs. In this paper, we consider n SUs and m PU channels where $n \geq m$.

A. PUs' activities model

Similar to [13], we model each PU channel's state as an independent two-state Markov chain alternating between the states *busy* (B) and *empty* (E) (Figure 1). Let α_i and β_i be the probabilities that the i^{th} channel switches its state from B to E and from E to B, respectively, for all $i \in \{1, \ldots, m\}$.

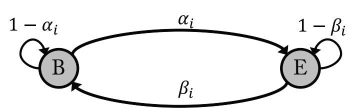

Fig. 1. Model of primary user's channel occupancy

B. SU-PU coexistence scenarios

From the network topology point of view, coexistence can be modeled by the following two scenarios:
(i) Long-range PU activity (Non-distributed): In this scenario, the transmission range of PUs is far beyond the transmission range of SUs. Therefore, all of the existing SUs experience similar PU's activities [14]. We refer to this scenario as the *non-distributed* cognitive radio.
(ii) Short-range PU activity (Distributed): In this scenario, the transmission range of PUs is comparable to that of SUs (e.g, in IEEE 802.22 when the PU network consists of wireless microphones [15]). Therefore, the SUs that are far apart will have different channel sensing results [16]. We refer to this scenario as the *distributed* cognitive radio. An example of a distributed cognitive radio can be seen in Figure 2. Each SU can only sense PUs' activities on the channels indicated by gray color. The PU channels indicated by the white color are always sensed empty and can be accessed by the corresponding SU at any time without causing harmful interference to the PUs.

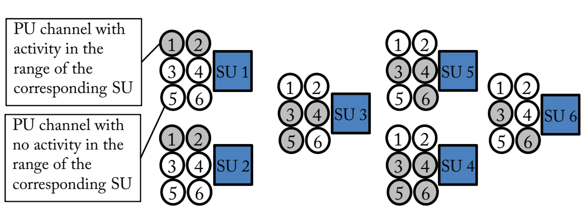

Fig. 2. An example of a distributed CRN with $n = 6$ SUs (squares) operating in a spectrum containing $m = 6$ PU channels. Each SU can only sense PU's activity on the channels shown by gray circles. The PU channels shown by white circles can be accessed by the corresponding SU at any time without causing harmful interference to the PUs.

In the case of non-distributed CRNs, due to the long range of PUs' transmitters, all SUs are expected to experience almost similar PU's activities. Hence, a CSS algorithm similar to [6] is optimal. Yet, in the case of distributed CRNs, sensing algorithms that assign a unique PU channel to each SU are not optimal anymore. In the distributed CRN scenario as presented in Figure 2, each SU is only affected by some of the PUs (indicated with gray-colored channels) and remains out of the transmission range of other PUs (indicated with white-colored channels). Therefore, two distant SUs are more likely to be in the transmission range of completely different sets of PUs. This location-dependent channel state information makes the one-to-one CSS algorithms (e.g. [6]) sub-optimal due to their inability to assign distant SUs to sense the same channel. For instance two SUs that are outside of each other's transmission range can both be assigned to sense an empty channel and access it to increase the throughput (*frequency reuse*). Yet, such scenarios are not accounted for in the one-to-one CSS.

As an example in Figure 2, SU nodes 2 and 5 may only interfere with PU channels $\{1, 2\}$ and $\{3, 4, 6\}$, respectively. Since the intersection of these subsets is empty, SU nodes 2 and 5 can access the same PU channel at the same time. These opportunities will be lost if a one-to-one sensing approach is employed for distributed CRNs.

C. Sensing-Access trade-off

The SUs are assumed to be synchronized and operate in time on a frame-by-frame structure as in [13, 17]. The frame structure of a CRN, as is shown in Figure 3, includes a sensing time T_S and a transmission time T_X that add up to the total frame time T. During T_S all SUs cease their transmission, perform spectrum sensing, and report the sensing results on a dedicated common control channel to the BS. As depicted in Figure 3, the sensing time T_S is comprised of two parts, namely *channel sensing time* (T_C) and *sensing and access overhead time* (T_O). During channel sensing time T_C, each SU senses a PU channel. During sensing and access overhead time T_O, SUs report the sensing results to the BS and the BS assigns each SU a channel to sense in the next frame. In addition, the BS informs SUs that whether or not they can access the channels they have sensed empty. In standard IEEE 802.22, T has been set equal to $200ms$ [13, 18]. As in [19], the sensing efficiency η can be defined as the ratio of the transmission time over the total frame time, i.e.,

$$\eta = \frac{T_X}{T} = 1 - \frac{T_C + T_O}{T}. \tag{1}$$

For a fixed T, η can be increased by reducing the sensing time T_S, which is equivalent to more transmission time for SUs.

We employ energy detectors in SUs for the spectrum sensing purposes. Energy detector relaxes the necessity of prior information about PU's signal at SUs.

III. Our Proposed Clustering Technique for Spectrum Sensing in Distributed Cognitive Radios

As we have discussed in Section II, in the distributed CRNs, one-to-one sensing assignment approaches, such as [6] are

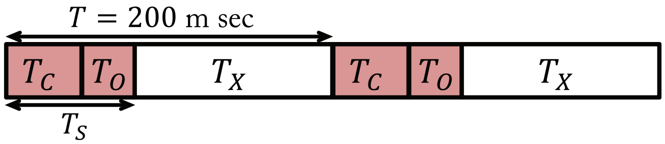

Fig. 3. The frame structure of an SU's operation in a CRN depicting two consecutive time frames. During the sensing time all SUs cease their transmissions.

not optimal. Therefore, we propose to group SU nodes into several non-overlapping clusters such that each cluster will be treated as a non-distributed CRN. In general, the goal of every clustering process is to group the entities into the most *homogenous* groups that are maximally *separated* from each other [20].

To describe any clustering process, three elements should be defined, namely, the distance measure, the clustering algorithm, and the clustering evaluation indicators. In the following, we defined each of the above elements for our proposed method on clustering SUs.

1) Sensing-based distance measure: In large-scale CRNs, the Euclidean distance between the SUs can be considered as a metric to perform clustering. This metric may seem to perform the best, yet in many applications, obtaining the location information about the SUs might not be readily possible. Therefore, we propose to use the *sensing results of SUs* as a *clustering metric*. The BS stores each node's collective history of previous sensing experiences as the node's *belief vector*. The belief vector for SU j at time t is a probability mass function (pmf) $\underline{x}_j(t) \triangleq [x_{j,1}(t), \dots, x_{j,m}(t)]^T$, where $x_{j,i}(t)$ denotes the probability that SU j would have chosen channel i for sensing in sensing frame t if it was up to itself to choose and $\sum_{i=1}^{m} x_{j,i}(t) = 1$. The belief vector represents the accumulative history of SU's observations prior to time t. The belief vector is initialized as $\underline{x}_j(0) = [\frac{1}{m}, \frac{1}{m}, \dots, \frac{1}{m}]^T$ and is updated by every sensing attempt of SU j for all $j \in \{1, \dots, n\}$ following the learning algorithm in [21].

We define the distance between two SUs based on the distance between their belief vectors.More specifically, we define the distance $\overline{D}_{KL}$ between any two SUs as the Kullback-Leibler (KL) divergence between beliefs of those SUs. In other words, the distance is measured by the divergence in the beliefs of SUs j and SU l are defined as follows

$$\overline{D}_{KL} \triangleq D_{KL}(\underline{x}_j(t) \| \underline{x}_l(t)) + D_{KL}(\underline{x}_l(t) \| \underline{x}_j(t)), \qquad (2)$$

where $D_{KL}(\underline{x}_j(t) \| \underline{x}_l(t)) \triangleq \sum_{i=1}^{m} x_{j,i}(t) \log \frac{x_{j,i}(t)}{x_{l,i}(t)}$. If two SUs experience exactly the same set of observations on PUs' channels, they will have the same beliefs on PU's channels and the KL distance between them will be zero. Similarly, SUs with different PU's channel sensing experiences will have diverged beliefs and consequently greater distances.

In our numerical simulations, we also considered the ℓ_2-norm of the difference of the pmfs as

$$D_{\ell_2} = \|\underline{x}_j(t) - \underline{x}_l(t)\|_2 = \Big(\sum_{i=1}^{m} (x_{j,i} - x_{l,i})^2 \Big)^{\frac{1}{2}}, \qquad (3)$$

for comparison. As we will see later in the simulations $\overline{D}_{KL}$ is a better distance measure compared to D_{ℓ_2}.

2) The Clustering algorithm: Various clustering algorithms have been proposed in the literature for different applications. In our case, we are interested in a clustering algorithm with the following properties:

(i) Hard partitioning: the sets of SUs in each cluster are non-overlapping.

(ii) Low complexity: In our proposed approach, the BS is required to frequently perform clustering based on the dynamics of PUs' activities. Therefore, the clustering algorithm should have low computational complexity to facilitate real-time clustering.

(iii) No prior knowledge of PUs' activities model at SUs: We assume SUs are not aware of the PUs' activities statistics a priori (for that we set $\underline{x}_j(0) = [\frac{1}{m}, \frac{1}{m}, \dots, \frac{1}{m}]^T$ for all $j \in \{1, \dots, n\}$).

To meet the above requirements, we employ the HCS clustering algorithm [22] to cluster the SUs. HCS is a connection-based and low-complexity clustering algorithm, in which the SUs are represented as vertices (nodes) of a graph and the edges are determined based on the SU's sensing history. Specifically, there exists an edge between two nodes in the graph if the divergence measure $\overline{D}_{KL}$ between the corresponding SUs is smaller than a threshold τ_d. A larger τ_d results in a graph with more edges and vice versa. We then apply HCS [22] to find the highly connected subgraphs and cluster nodes (see Figure 4).

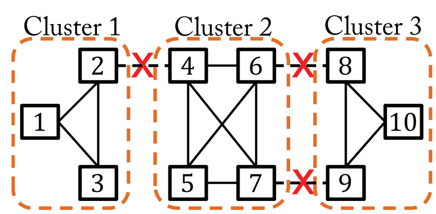

Fig. 4. Highly connected subgraph (HCS) clustering. Existence of an edge represents smaller than τ_d divergence. The dashed edges with a cross on them will be removed to form the highly connected subgraphs.

The HCS clustering algorithm determines the clusters by finding the minimum number of highly connected subgraphs in the underlying graph of SUs in the network (Figure 4). Let *connectivity* of a graph G, $k(G)$, be the minimum number of edges that their removal makes the graph disconnected. Therefore, $k(G)$ is equal to the cardinality of the minimum cut set of the graph G. A connected subgraph G_T with q vertices is called an HCS if $k(G_T) > q/2$ [22]. The clustering algorithm determines the HCSs of G. It first, checks if G is an HCS itself. If true, the algorithm is terminated, otherwise by finding the minimum cut set of G and removing the associated edges, it forms two subgraphs. Each subgraph will be checked for being an HCS. This process is continued until all existing subgraphs are HCSs [22]. For example the graph in Figure 4 is composed of three HCSs. First by removing the edge between nodes 2 and 4 two disconnected subgraphs is formed ($\{1, 2, 3\}$

and $\{4,5,6,7,8,9,10\}$). Subgraph $\{1,2,3\}$ is an HCS. In subgraph $\{4,5,6,7,8,9,10\}$ by removing two edges between the nodes 6 and 8 and the nodes 7 and 9, two HCSs $\{4,5,6,7\}$ and $\{8,9,10\}$ are formed and the algorithm is terminated. This algorithm has a low computational complexity compared to other algorithms (such as k-means) that minimize the distance measure between the nodes within each cluster. This is because HCS performs a connection-based clustering [23].

It should be noted that the parameter τ_d determines the number of clusters. A larger τ_d results in a graph with more edges and results in less number of clusters.

3) Clustering evaluation methods: There are many ways to assess the performance of a clustering method. In this work, we evaluate the clustering performance based on the overall efficiency of the channel sensing process. The overall performance of the sensing algorithm can be measured in terms of the *channel sensing success rate* defined as,

$$R_s = \frac{\text{Average number of successful sensing attempts per time frame}}{N}. \tag{4}$$

A sensing is considered to be a *successful attempt* if an SU finds its assigned and sensed channel empty and if there is no other SU in its transmission range that has been assigned the same channel to sense. Otherwise, the sensing attempt is considered to be a *failed attempt*.

In Figure 5, we provide an example of a spectrum sensing assignment. In this example $n = 6$ and $m = 3$ and we have 3 clusters each containing two SUs (see Figure 5(a)). Figure 5(b) represents the channel assignments on a bipartite graph between the SUs (square nodes) and the PU channels (circular nodes) as well as the state of the PU channels (B and E stand for the busy and the empty states, respectively). Using the assignment represented by the edges of bipartite graph in Figure 5(b), the SUs $\{1,2,3,5,6\}$ will sense an empty channel. Yet, SU 1 from cluster 1 and SU 3 from cluster 2 are in the transmission range of each other. Therefore, we do not consider those channel sensing as success (due to the possible interference between SUs 1 and 3). Hence, channel sensing success rate becomes $R_s = \frac{3}{6} = 0.5$.

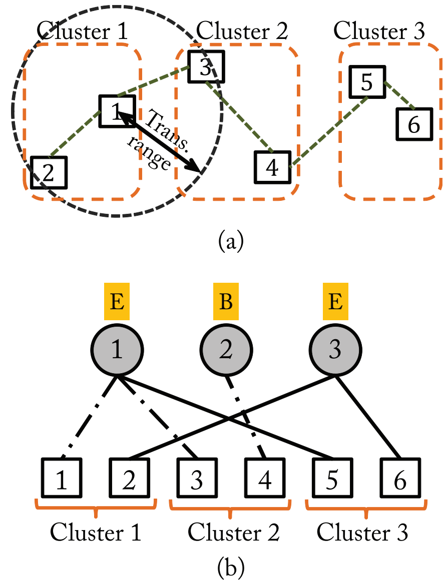

Fig. 5. Channel sensing allocation for a network with $n = 6$ (SUs are represented by squares) and $m = 3$ (channels are represented by circles). Subfigure (a) represents the location of SUs. There is a dashed line between two SUs if they are in the transmission range of each other. Subfigure (b) represents the channel sensing assignments and the state of each PU channel on a bipartite graph. In addition the edges with dash-dotted line and solid line represent the failed and the successful sensing attempts, respectively.

Algorithm 1: The proposed clustering-based spectrum sensing in distributed CRNs at time frame t.

1: SUs sense the allocated channels.
2: The BS receives the sensing results (B or E) corresponding to the sensed channels from SUs.
3: The BS updates the belief vectors $\underline{x}_j(t)$ for all $j \in \{1,\ldots,n\}$ using learning algorithm *LA1* in [21].
4: The BS partitions SUs into k clusters using the HCS clustering algorithm.
5: The BS performs the one-to-one matching algorithm proposed in [6] to allocate channels the members of each cluster.
6: The BS transmits the channel access permissions and the ID of the channel that each SU has to sense at frame $t+1$.

IV. Clustering-Based Coordinated Spectrum Sensing in Distributed CRNs

In this section, we introduce our clustering-based algorithm for coordinated spectrum sensing. At the beginning of each frame, the BS, after receiving the sensing results from the SUs (whether the sensed channel is busy or empty), performs the following steps. SUs are partitioned into several clusters using the HCS clustering algorithm. For every cluster, the BS determines the unique channels to be sensed by the members of the cluster, performing a one-to-one matching algorithm [6] between the members of that cluster and the channels. Algorithm 1 represents the pseudo code of the steps taken at the beginning of each frame in our proposed algorithm. The time required to perform Algorithm 1 at each frame is equal to T_S, in which Step 1 takes T_C seconds and all other steps together take T_O seconds.

It should be noted that in the distributed CRN scenarios, the proposed algorithm can achieve a higher sensing success rate compared to non-cooperative [21] and non-clustered [6] strategies. In non-cooperative scenario each SU greedily tries to sense the PU channel with maximum probability of being empty. Therefore, it is very likely that two SUs at transmission range of each other, sense the same PU channel due to lack of coordination. On the other hand, in coordinated but non-clustered scenario the opportunities of frequency reuse will be lost.

V. Simulation Results

To perform the numerical experiments, we assume $n = 20$ SUs and $m = 20$ PUs, all with transmission range equal to 1 distance unit, are uniformly at random distributed in an area with size $A = 100$ (distance unit)^2. In addition, we assume all PU channels have the similar parameters (i.e., $\alpha_i = \alpha$ and $\beta_i = \beta$ for all $i \in \{1, \ldots, m\}$). We set $\alpha = \beta = 0.01$. The results of this simulation is shown in Figure 6, which represents the channel *sensing success rate* (R_s) versus the number of clusters for different clustering scenarios. As we have mentioned, using HCS algorithm, unlike k-means, the BS does not require to decide on the number of clusters prior to the clustering. However, this algorithm can be enforced to group SUs into a pre-defined number of clusters. Hence, for sake of comparison with k-means we have forced HCS algorithm to form pre-determined number of clusters. We numerically obtained the optimal threshold for each given number of clusters in case of HCS clustering (e.g., when we have $k = 5$ clusters, $\tau_d = 0.16$). Channel sensing success rate is defined in (4). As we can see for the cluster size of $k = 5$, we obtain nearly 20% more sensing success rate using the proposed HCS clustering algorithm with KL divergence measure compared to the non-clustered scenario.

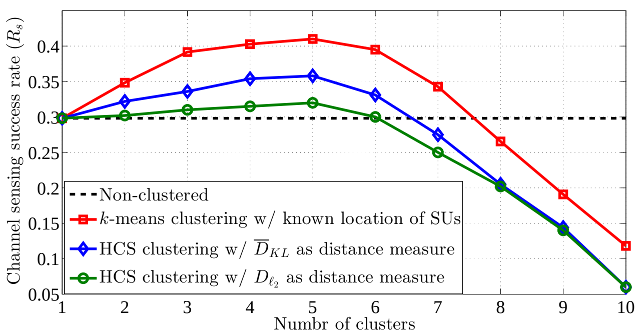

Fig. 6. Channel sensing success rate using different clustering approaches versus the number of clusters

In Figure 6, we have compared different distance measures for clustering as well. Each solid line represents a different measure for performing the clustering. In the k-means clustering, we assumed that the BS knows the location information of all SUs. As we can see in Figure 6, the performance of our proposed algorithm (HCS w/ KL divergence) is slightly degraded compared to the k-means clustering. However, the latter requires the knowledge of the locations of all SUs, which may not be feasible. We have also depicted the results of HCS clustering using D_{ℓ_2} between SUs beliefs as a distance measure. As we can see D_{ℓ_2} does not perform as well as $\overline{D}_{KL}$.

In Figure 7, we have depicted channel sensing success rate R_s as a function of distance measure threshold τ_d. The simulation parameters are the same as in Figure 6. Each point represents the average of 1000 random implementations of SUs and PUs. The righthand-side axis of Figure 7 represents the average number of clusters that HCS clustering algorithm finds. For smaller τ_d values, the graph of SUs have only a few number of edges and HCS algorithm will form many clusters each containing only a few nodes. Hence, the coordination among SU node will be very limited and R_s will be small. On the other hand, by increasing the threshold τ_d, the graph of SUs approaches to a complete graph. Consequently, the

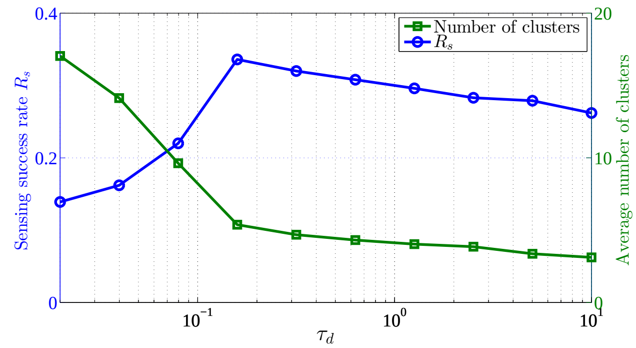

Fig. 7. The channel sensing success rate and the average number of clusters versus distance measure threshold τ_d

HCS algorithm forms fewer clusters causing to exploit fewer frequency reuse opportunities. Therefore, there is an optimal value of τ_d such that the benefits of coordination and frequency reuse are being exploited at the same time.

In Table I, we derived the sensing efficiency factor η for the same parameters as in Figure 6. We find the spectrum sensing efficiency factor for three non-cooperative [21], non-clustered [6], and our proposed algorithm. In all of the three algorithms the sensing success rate R_s is set greater than or equal to 0.35. Therefore, as we can see from the table in cases of non-cooperative and non-clustered spectrum sensing, SUs have to perform more than one channel sensing at each sensing frame to comply with the sensing success rate criteria (i.e. $R_s \geq 0.35$). The results on η are based on assuming the required time to sense one channel is $5ms$ and $T = 200ms$ [13, 18]. In Table II, we have considered employing HCS and

TABLE I
SENSING EFFICIENCY FACTOR, η OF DIFFERENT ALGORITHMS. OUR PROPOSED ALGORITHM (CLUSTERING W/ HCS) HAS THE HIGHEST SENSING EFFICIENCY.

Algorithm	Non-cooperative	Non-clustered	Clustering w/ HCS
Channel sensing per frame	4	2	1
η	0.9	0.95	0.975

k-means with the KL divergence measure and the separation and the homogeneity of these clustering techniques are compared. Homogeneity is the measure of closeness between the members of one cluster and separation measures distance of members of one clusters with the other clusters members on average (both are defined in [23]). A better algorithm has greater homogeneity and smaller separation. The simulation parameters are $n = 50$, $m = 20$, and $\alpha = \beta = 0.01$. In Table II, the number of clusters for k-means should be determined a priori. Therefore, we have picked the number of clusters that results in the better performance. As we can see HCS performs slightly better than k-means in terms of average homogeneity and performs slightly worse than k-means in terms of average separation. In addition as we can see, the computational complexity of the HCS algorithm is greatly

TABLE II

SIMULATION RESULTS FOR DIFFERENT CLUSTERING ALGORITHMS AND MEASURES USING KULLBACK-LEIBLER DIVERGENCE MEASURE AS THE DISTANCE MEASURE.

Algorithm	Number of clusters	Average homogeneity	Average separation	Run time
k-means	11	1.37	1.31	$O(n^2 \log n)$
HCS	12	1.39	1.40	$O(n \log n)$

smaller than that of k-means algorithm.

VI. CONCLUDING REMARKS

In this paper, we have considered the problem of coordinated spectrum sensing in the distributed cognitive radio networks. The efficiency of coordinated (one-to-one) spectrum sensing in *non-distributed* CRNs and the benefits of frequency reuse for *distributed* CRNs made us to seek a balanced solution that uses the benefits of both schemes by clustering the SUs. SUs within one cluster can leverage the one-to-one spectrum sensing while SUs in different clusters can benefit from the frequency reuse. Clearly, the underlying clustering method will be very important in the efficiency of our scheme. We introduced a novel metric for clustering SU nodes that is based on the similarities in the previous channels sensing results of SUs and the belief that each SU has about the status of each channel. Using this metric in conjunction with a low-complexity clustering algorithm based on highly connected subgraphs enables a base station to efficiently form clusters without the need to know the location of the SUs. We have shown through extensive simulations that the proposed algorithm can considerably increase channel sensing success rate for secondary users at a reasonable computational cost.

REFERENCES

[1] I. F. Akyildiz, W.-Y. Lee, M. C. Vuran, and S. Mohanty, "Next generation/dynamic spectrum access/cognitive radio wireless networks: A survey," *Computer Networks*, vol. 50, no. 13, pp. 2127 – 2159, 2006.

[2] S. Haykin, "Cognitive radio: brain-empowered wireless communications," *Selected Areas in Communications, IEEE Journal on*, vol. 23, pp. 201–220, Feb. 2005.

[3] L. Lu, X. Zhou, U. Onunkwo, and G. Y. Li, "Ten years of research in spectrum sensing and sharing in cognitive radio," *EURASIP Journal on Wireless Communications and Networking*, vol. 2012, no. 1, pp. 1–16, 2012.

[4] S. Xie, Y. Liu, Y. Zhang, and R. Yu, "A parallel cooperative spectrum sensing in cognitive radio networks," *Vehicular Technology, IEEE Transactions on*, vol. 59, no. 8, pp. 4079–4092, 2010.

[5] Z. Wang, Z. Feng, and P. Zhang, "An iterative hungarian algorithm based coordinated spectrum sensing strategy," *Communications Letters, IEEE*, vol. 15, pp. 49 –51, January 2011.

[6] B. Shahrasbi and N. Rahnavard, "Cooperative-Parallel spectrum sensing in cognitive radio networks using bipartite matching," in *The 2011 Military Communications Conference (MILCOM 2011)*, (Baltimore, Maryland, USA), Nov. 2011.

[7] Y. Liu, S. Xie, R. Yu, Y. Zhang, and C. Yuen, "An efficient mac protocol with selective grouping and cooperative sensing in cognitive radio networks," 2013.

[8] C. Sun, W. Zhang, and K. Ben, "Cluster-based cooperative spectrum sensing in cognitive radio systems," in *Communications, 2007. ICC '07. IEEE International Conference on*, pp. 2511 –2515, june 2007.

[9] J. Unnikrishnan and V. V. Veeravalli, "Cooperative sensing for primary detection in cognitive radio," *Selected Topics in Signal Processing, IEEE Journal of*, vol. 2, no. 1, pp. 18–27, 2008.

[10] J. Wei and X. Zhang, "Energy-efficient distributed spectrum sensing for wireless cognitive radio networks," in *INFOCOM IEEE Conference on Computer Communications Workshops , 2010*, pp. 1 –6, march 2010.

[11] Y. Sun, H. Hu, F. Liu, H. Yi, and X. Wang, "Selection of sensing nodes in cognitive radio system based on correlation of sensing information," in *Wireless Communications, Networking and Mobile Computing, 2008. WiCOM '08. 4th International Conference on*, pp. 1 –6, oct. 2008.

[12] M. Bradonjic and L. Lazos, "Graph-based criteria for spectrum-aware clustering in cognitive radio networks," *Ad Hoc Networks*, 2011.

[13] H. Kim and K. Shin, "Efficient discovery of spectrum opportunities with MAC-layer sensing in cognitive radio networks," *IEEE Transactions on Mobile Computing*, pp. 533–545, 2007.

[14] R. Chen, J.-M. Park, and K. Bian, "Robust distributed spectrum sensing in cognitive radio networks," in *INFOCOM 2008. The 27th Conference on Computer Communications. IEEE*, pp. 1876–1884, April.

[15] C. Stevenson, G. Chouinard, Z. Lei, W. Hu, S. Shellhammer, and W. Caldwell, "IEEE 802.22: The first cognitive radio wireless regional area network standard," *Communications Magazine, IEEE*, vol. 47, no. 1, pp. 130–138, 2009.

[16] S. Jafar and S. Srinivasa, "Capacity limits of cognitive radio with distributed and dynamic spectral activity," *Selected Areas in Communications, IEEE Journal on*, vol. 25, no. 3, pp. 529–537, April.

[17] E. Peh, Y.-C. Liang, Y. L. Guan, and Y. Zeng, "Cooperative spectrum sensing in cognitive radio networks with weighted decision fusion schemes," *Wireless Communications, IEEE Transactions on*, vol. 9, no. 12, pp. 3838–3847, 2010.

[18] C. Cordeiro, K. Challapali, D. Birru, and N. Sai Shankar, "IEEE 802.22: the first worldwide wireless standard based on cognitive radios," in *New Frontiers in Dynamic Spectrum Access Networks, 2005. DySPAN 2005. 2005 First IEEE International Symposium on*, pp. 328 –337, 2005.

[19] W.-Y. Lee and I. Akyildiz, "Optimal spectrum sensing framework for cognitive radio networks," *Wireless Communications, IEEE Transactions on*, vol. 7, no. 10, pp. 3845 –3857, 2008.

[20] R. Sharan and R. Shamir, "Click: A clustering algorithm with applications to gene expression analysis," in *ISMB'00*, pp. 307–316, AAAI Press, Menlo Park, CA.

[21] C. Tekin, S. Hong, and W. Stark, "Enhancing cognitive radio dynamic spectrum sensing through adaptive learning," in *Military Communications Conference, 2009. MILCOM 2009. IEEE*, pp. 1 –7, 2009.

[22] E. Hartuv and R. Shamir, "A clustering algorithm based on graph connectivity," *Information Processing Letters*, vol. 76, pp. 175–181, 2000.

[23] R. Xu and I. Wunsch, D., "Survey of clustering algorithms," *Neural Networks, IEEE Transactions on*, vol. 16, pp. 645 –678, may 2005.