Distributed Cognitive Radio Network Management via Algorithms in Probabilistic Graphical Models

Yingbin Liang, Lifeng Lai, and John Halloran

***Abstract*—In this paper, cognitive radio wireless networks are investigated, in which a number of primary users (PUs) transmit in orthogonal frequency bands, and a number of secondary users (SUs) monitor the transmission status of the PUs and search for transmission opportunities in these frequency bands by collaborative detection. A network management problem is formulated to find the configuration of SUs (assignment of SUs) to detect PUs so that the best overall network performance is achieved. Two performance metrics are considered, both of which characterize the probability of errors for detecting transmission status of all PUs. For both metrics, a graphical representation of the problem is provided, which facilitates to connect the problems under study to the sum-product inference problem studied in probabilistic graphical models. Based on the elimination algorithm that solves the sum-product problem, a message passing algorithm is proposed to solve the problem under study in a computationally efficient manner and in a distributed fashion. The complexity of the algorithm is shown to be significantly lower than that of the exhaustive search approach. Moreover, a clique-tree algorithm is applied to efficiently compute the impacts of each SU's choice on the overall system performance. Finally, simulation results are provided to demonstrate the considerable performance enhancement achieved by implementing an optimal assignment of SUs.**

***Index Terms*—Cognitive radio, collaborative detection, distributed algorithm, message passing algorithm, probabilistic graphical model.**

I. Introduction

THE COGNITIVE radio technology has recently received considerable attention due to its ability to substantially improve the spectral efficiency of wireless communication systems [2], [3]. The basic idea is to allow unlicensed users, referred to as secondary users (SUs), to reuse the spectrum when licensed users, referred to as primary users (PUs), are not using it. The ability to efficiently and reliably detect the presence of PUs' transmission is one of the fundamental building blocks of the cognitive radio technology [4]. The idea of collaborative detection was proposed and studied recently in, e.g., [5]–[16] and in [17] for a review, which suggests that multiple SUs can collaborate by combining their observations to enhance the detection reliability. Previous work on this topic mainly focuses on design of collaborative detection schemes for a single primary user. In this paper, we study collaborative detection from a network perspective. In particular, we study a network management problem in cognitive radio networks, which addresses the best assignment of SUs for detecting transmission status of multiple PUs so that the overall network performance is optimized.

The cognitive radio network that we consider consists of multiple PUs, each transmitting at a different frequency band (that may include a set of subbands). Each PU has a certain detection range around it such that the SUs within this range may reliably detect whether this PU is transmitting or not. There are also multiple SUs in the network, and each SU may fall into the detection ranges of multiple PUs. Due to constraints on the hardware design and implementation complexity, each SU is assumed to tune only to one PU's frequency band at a time for detection. This is also due to the fact that one PU's frequency band in general contains multiple subbands, and SUs need to scan over these subbands. Hence, the SUs that fall into the detection ranges of multiple PUs need to select one PU for detection. SUs that select the same PU can collaborate to improve the detection performance.

Two types of detection errors are of practical significance. The first type of error occurs when the PU is in transmission but the SUs wrongly determine that the PU is not in transmission. In this case, the subsequent transmissions of the SUs will cause interference to the PU. The probability of this type of error is referred to as the *probability of interference*, and it must be guaranteed to be sufficiently small for each PU as requested by interweave cognitive radio networks. The second type of error occurs when the PU is not in transmission but the SUs determine that the PU is in transmission. As a result, the transmission opportunity is wasted. The probability of this type of error is referred to as the *probability of missed opportunity*. To utilize the spectrum in an efficient manner, the system design needs to minimize the probability of missed opportunity subject to the constraint that the probability of interference is less than a given threshold.

Manuscript received 1 December 2009; revised 5 June 2010. The material in this paper has been presented in part at the 47th Annual Allerton Conference on Communication, Control, and Computing, Monticello, IL, USA, Sept. 2009 [1]. The work of Y. Liang and J. Halloran was supported by the National Science Foundation CAREER Award under Grant CCF-10-26565.

Y. Liang is with Department of Electrical Engineering and Computer Science, Syracuse University, Syracuse, NY 13244, USA (e-mail: yliang06@syr.edu).

L. Lai is with the Department of Systems Engineering, University of Arkansas at Little Rock, Little Rock, AR 72204, USA (e-mail: lxlai@ualr.edu).

J. Halloran is with Department of Electrical Engineering, University of Washington, Seattle, WA 98195, USA (e-mail: halloj3@u.washington.edu.

Y. Liang and J. Halloran were both with Department of Electrical Engineering, University of Hawaii, Honolulu, HI 96822, when this work was conducted.

Digital Object Identifier 10.1109/JSAC.2011.110207.

In this paper, we consider two overall system performance metrics (objective functions): the sum of the probabilities of missed opportunities over all PUs and the maximum among the probabilities of missed opportunities over all PUs. It is clear that the probability of missed opportunity for each PU depends on which SUs are collaboratively detecting this PU, and how well these SUs receive signals from the PU if the PU transmits. Hence, the assignment of SUs to these PUs leads to tradeoffs among the individual probabilities of missed opportunity of PUs. The assignment of SUs that minimizes the sum of all individual probabilities (the first performance matric), which we refer as the *min-sum* problem in the sequel, thus achieves the best tradeoff among these probabilities and yields the best exploitation of PUs' white space (transmission opportunity) from the overall system point of view. On the other hand, the assignment that minimizes the maximum among the probabilities of all PUs, which we refer as the *min-max* problem, guarantees that SUs equally exploit the transmission opportunities arising in different parts of the network.

Although the search for the best assignment can be carried out at a centralized controller, such an approach is not desirable in practice, since it is in general too costly to centralize the network information for the controller to perform the search for the optimal assignment, in particular for large and dynamic networks. The network information needed for the centralized search includes the number of PUs and SUs in the network, the location of each node, the transmission power of each PU etc. Furthermore, many networks do not even have a central controller. In this paper, we are interested in designing distributed algorithms that allow the SUs to find the optimal assignment using only their local information. As will be clear in the sequel, the only information required for each SU in our algorithm is the information of its neighbors. In this way, the minimization can be efficiently performed in a distributed manner at each SU locally.

We obtain our solution by first identifying a close link between the problem under consideration and the sum-product inference problem studied in the context of probabilistic graphical models (see, e.g., [18], [19]). We show that the assignment problems (both the min-sum and min-max problems) possess the same algebraic structure of *commutative semiring* as the sum-product problem. Based on the celebrating elimination algorithm [18], [19] derived for solving the sum-product problem, we propose a message passing algorithm, which efficiently solves our problem in a distributed fashion that does not need centralized global information of the network. By analyzing the complexity of the algorithm, and comparing it with that of the exhaustive search approach that does not exploit the problem structure, we show that the proposed algorithms significantly reduce the computational complexity. We also provide graphical representations of our problems and interpretations of our solutions.

We finally study the problem of computation of beliefs, which captures the impact of the choice (assignment) of each SU on the overall system objective functions for both the min-sum and min-max problems. Such a problem can be solved efficiently by a message passing algorithm originally designed for the sum-product problem in probabilistic graphical models for tree graphs. However, our problem in general is not represented by tree graphs. In this case, we apply a more general clique-tree (junction-tree) algorithm [18], [19] that is applicable for general graphs. We first propose a distributed scheme to construct a clique tree based on the graphical representation of our problem, and then apply the clique-tree algorithm that runs a message passing algorithm over the clique tree.

We note that the network spectrum management problem for cognitive radio networks has attracted considerable attention and has been addressed from a number of perspectives recently. For example, an auction based spectrum management scheme is proposed in [20] for cognitive radio networks. A spectrum management scheme that exploits spectrum usage patterns to improve the overall spectrum efficiency is proposed in [21]. The reader can refer to [22] for an overview of recent progress in this area. Compared with the existing work, our paper develops an approach based on probabilistic graphical models to improve the spectrum efficiency via optimizing the assignment of secondary users for collaborative sensing. This perspective incorporates the physical layer design of collaborative sensing into the network management issue and demonstrates that network management based on physical layer design improves the spectrum efficiency substantially.

The rest of the paper is organized as follows. In Section II, we describe the network model and performance metrics. In Section III, we present our solution to the min-sum and min-max problems. In Section IV, we study the complexity of the proposed algorithms. In Section V, we study the problem of computation of beliefs by applying the clique-tree algorithm. In Section VI, we present simulation results to illustrate the proposed algorithms. Finally, in Section VII, we provide a few concluding remarks.

II. System Model and Problem Statement

A. Network Model

We consider a cognitive radio network with K PUs communicating over K orthogonal frequency bands (each frequency band may contain a number of subbands). There are also J SUs in the network, which are collaboratively monitoring the transmission status of the K PUs. We index the PUs and SUs by PU $1, \ldots,$ PU K, and SU $1, \ldots,$ SU J, respectively. Fig. 1 illustrates an example network with four PUs (indicated by circles) and nine SUs (indicated by squares).

An SU, say SU j, may choose to monitor a PU, say PU k, if SU j is within the detection range of PU k, i.e., $d_{jk} < D_k$, in which d_{jk} denotes the distance from SU j to PU k, and D_k denotes the detection range of PU k. However, as mentioned above, one SU can choose only one PU for detection at a time although it may be in the detection ranges of multiple PUs. For each SU, say SU j, we use s_j to denote the index of the PU that this SU chooses to detect.

For each PU, say PU k, we use $\mathcal{D}_k$ to denote the set that includes all indices of SUs that are in the detection range of PU k and hence may possibly choose PU k for detection, that is

$$\mathcal{D}_k = \{j \in \{1, \ldots, J\} : d_{jk} \leq D_k\}. \quad (1)$$

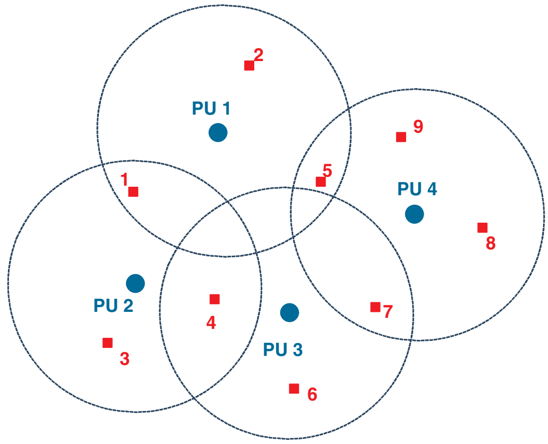


Fig. 1. An example cognitive network

The actual set of SUs that choose to detect PU k is a subset of $\mathcal{D}_k$.

We adopt the detection model used in [13]. We assume that each SU uses an energy detector to detect whether a PU is in transmission or not. Suppose that SU j chooses to detect PU k. We assume that signal power of PU k is at the same level as thermal noise power at the energy detector of SU j if SU j is at the distance η_k away from PU k. Hence, in general $\eta_k > D_k$, i.e., SUs within the detection range of PU k receives stronger signal power from PU k than thermal noise power. As in [13], we also assume that thermal noise at the energy detector is approximately Gaussian distributed in the unit of dB. We further normalize the mean of noise power to be 0dB. Hence if PU k is not in transmission, the output Y_{jk} at the energy detector of SU j contains noise only, and hence $Y_{jk} \sim \mathcal{N}(0, \sigma^2)$, where σ^2 is the variance of the noise power. On the other hand, if PU k is in transmission, the output at the energy detector is dominated by the signal power of PU k which is much stronger than the noise power if SU j is in the detection range. Hence, $Y_{jk} \sim \mathcal{N}(\mu_{jk}, \sigma^2)$, where the mean μ_{jk} is the received signal power minus the mean of noise power (that is at the same level as signal power at SU j if SU j is at the distance η_k away from PU k) due to normalization, and is given by

$$\mu_{jk} = -10b \log \frac{d_{jk}}{\eta_k}, \tag{2}$$

where b is the path loss exponent. We note that in general, the variances under the two conditions (PU's transmission is on or off) may be different. We here assume the same variance for simplicity. The algorithms presented in this paper are applicable for more general signal and detection models.

As mentioned above, SUs that choose to detect the same PU can pool their observations together to make a decision regarding to the transmission status of the PU. For simplicity, we assume that the SUs share their output signals with neighboring nodes. However, our algorithms are also applicable to more general scenarios, in which SUs share quantized outputs or local hard decisions with their neighboring nodes to reduce the communication cost.

We use $\underline{Y}_k$ to denote the vector with elements Y_{jk} being the observations of SU j that has $s_j = k$, i.e., SU j chooses to detect PU k. We also let $\underline{\mu}_k = \mathsf{E}[\underline{Y}_k]$, which includes μ_{jk} as elements. Hence, the set of SUs that choose to detect PU k are collaborating with each other to distinguish between the following two hypotheses:

$$\begin{aligned} \mathcal{H}_0^k: &\quad \underline{Y}_k \sim \mathcal{N}(\underline{0}, \sigma^2 \mathbf{I}), \\ \mathcal{H}_1^k: &\quad \underline{Y}_k \sim \mathcal{N}(\underline{\mu}_k, \sigma^2 \mathbf{I}), \end{aligned} \tag{3}$$

in which $\mathbf{I}$ is the identity matrix.

It is clear that the decision rule can be characterized by a threshold θ_k, and is given by [23]

$$\begin{aligned} \text{if } \underline{\mu}_k^T \underline{Y}_k > \theta_k, &\quad \text{determines } \mathcal{H}_1^k, \\ \text{if } \underline{\mu}_k^T \underline{Y}_k \le \theta_k, &\quad \text{determines } \mathcal{H}_0^k, \end{aligned} \tag{4}$$

where $\underline{\mu}_k^T$ denotes the transpose of the vector $\underline{\mu}_k$. We note that although $\underline{Y}_k$ contains only Y_{jk} with $s_j = k$, it is a function of all $s_j \in \mathcal{D}_k$, because all these s_j may affect how $\underline{Y}_k$ is comprised of.

We use $P_{I,k}$ to denote the probability of interference to PU k, i.e., $P_{I,k} = \Pr(\mathcal{H}_0^k | \mathcal{H}_1^k)$. In this case, PU k is in transmission but the SUs wrongly determine that the PU is not in transmission. Hence, the subsequent transmissions of the SUs will cause interference to the PU. We also use $P_{MO,k}$ to denote the probability of missed opportunity to use PU k's frequency band, i.e., $P_{MO,k} = \Pr(\mathcal{H}_1^k | \mathcal{H}_0^k)$. In this case, PU k is not in transmission but the SUs determine that PU k is in transmission. Hence, the transmission opportunity in PU k's frequency band is wasted.

We require that the probability of interference to each PU must be less than a given value γ, i.e.,

$$P_{I,k} \le \gamma, \quad \text{for } k = 1, \ldots, K.$$

It is easy to see that for each PU k, $P_{MO,k}$ is minimized when $P_{I,k} = \gamma$. Hence the detection threshold θ_k should be set such that $P_{I,k} = \gamma$. We first derive

$$P_{I,k} = 1 - \Pr(\underline{\mu}_k^T \underline{Y}_k > \theta_k | \mathcal{H}_1^k) = 1 - Q\left(\frac{\theta_k - \underline{\mu}_k^T \underline{\mu}_k}{\sigma \sqrt{\underline{\mu}_k^T \underline{\mu}_k}} \right), \tag{5}$$

in which $Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^\infty \exp(-u^2/2) \mathrm{d}u$.

Letting $P_{I,k} = \gamma$, we obtain

$$\theta_k = \sigma \sqrt{\underline{\mu}_k^T \underline{\mu}_k} Q^{-1}(1 - \gamma) + \underline{\mu}_k^T \underline{\mu}_k. \tag{6}$$

We hence obtain

$$P_{MO,k} = \Pr(\underline{\mu}_k^T \underline{Y}_k > \theta_k | \mathcal{H}_0^k) = Q\left(\frac{\theta_k}{\sigma \sqrt{\underline{\mu}_k^T \underline{\mu}_k}} \right) \tag{7}$$

$$= Q\left(\frac{\sigma \sqrt{\underline{\mu}_k^T \underline{\mu}_k} Q^{-1}(1 - \gamma) + \underline{\mu}_k^T \underline{\mu}_k}{\sigma \sqrt{\underline{\mu}_k^T \underline{\mu}_k}} \right). \tag{8}$$

Since $\underline{\mu}_k$ can be viewed as a function of $\{s_j, j \in \mathcal{D}_k\}$, we can express the above $P_{MO,k}$ as $P_{MO,k}(s_j, j \in \mathcal{D}_k)$.

B. Objective Functions

In this paper, we study system design under two overall system objective functions. The first one is to minimize the sum of the probabilities of missed opportunity over all possible assignments of SUs, i.e., our goal is to solve the following optimization problem

$$\min_{\{s_1,\ldots,s_J\}} \sum_{k=1}^{K} P_{MO,k}(s_j, j \in \mathcal{D}_k) \tag{9}$$

where $P_{MO,k}(s_j, j \in \mathcal{D}_k)$ is given in (8).

It is clear that the above optimization problem is in the form of *min-sum*. The solution to this problem may result in the case that some PUs have small probabilities of missed opportunity and some have large probabilities of missed opportunity. However, in many cases, we may need to guarantee the probability of missed opportunity to be small for all PUs so that the transmission opportunities over the entire network can be equally exploited. In these scenarios, it is desirable to consider the objective function of the maximum among probabilities of missed opportunity of K PUs, and minimize this function over all possible assignments of SUs. To be specific, the problem is in the form of *min-max* and is given by

$$\min_{\{s_1,\ldots,s_J\}} \max_{1\leq k\leq K} P_{MO,k}(s_j, j \in \mathcal{D}_k) \tag{10}$$

where $P_{MO,k}(s_j, j \in \mathcal{D}_k)$ is given in (8).

In the following sections, we will study the min-sum problem in more detail. Once the min-sum problem is solved, the min-max problem can be solved in a similar manner.

C. Graphical Representation of the Problem

In this section, we provide a graphical representation of the problems given in (9) and (10). This representation will facilitate the application of the elimination algorithm in Section III and the clique-tree algorithm in Section V, and the analysis of complexity of the algorithms in Section IV.

We can represent the problems given in (9) and (10) by an undirected graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the set that includes all SUs as nodes, and $\mathcal{E}$ is the set that includes all edges in the graph $\mathcal{G}$, as detailed in the sequel. The graph is induced by the set of functions $P_{MO,k}(s_j, j \in \mathcal{D}_k)$ for $k = 1, \ldots, K$ in that the nodes of the graph represent all variables involved in these functions, and there is an edge between two nodes if their corresponding variables appear in one common $P_{MO,k}(s_j, j \in \mathcal{D}_k)$ for some k. In this case, the corresponding SUs are in PU k's detection range. For example, if we consider the network given in Fig. 1, it is clear from the figure that SUs 1, 2, 5 are in the detection range of PU 1, and hence there are edges between any two of them. The graphical representation of the network in Fig. 1 is given in Fig. 2. It is clear that each function $P_{MO,k}(s_i, i \in \mathcal{D}_k)$ corresponds to one clique (a subgraph with one edge between every pair of nodes). We further note that in general the graph contains circles and is not a tree graph.

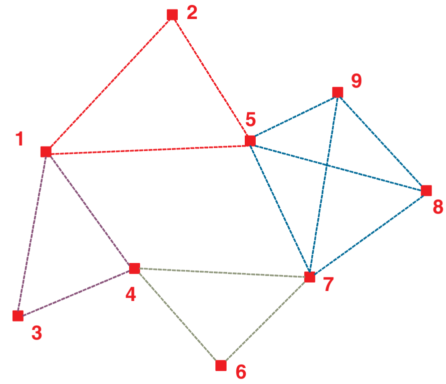


Fig. 2. Graphical representation of the example cognitive network in Fig. 1

III. Distributed Message Passing Algorithm

In this section, we present efficient distributed algorithms for the SU assignment problems (9) and (10). We first provide the solution for the min-sum problem in detail and then describe how to solve the min-max problem using the results obtained in the min-sum problem.

A. The Min-Sum Problem

To solve the minimization problem in (9), we first note that this problem is not decomposable if the corresponding graph is connected, which is mostly encountered in practice. Hence, independently optimizing each $P_{MO,k}$ functions or a subset of these functions is not possible, because in general these functions are interconnected by common variables. In this case, one possible approach is to compute the objective function for all possible assignments of SUs, i.e., all possible configurations of the values of $(s_1, \ldots, s_J)$, and then find the minimal value for the objective function and the corresponding assignment of SUs. Such an approach has several drawbacks. First, the network information, i.e., all information needed to compute $P_{MO,k}$ for $k = 1, \ldots, K$, needs to be collected at one center to perform the minimization. This causes a lot of network traffic, in particular, for large networks. Furthermore, wireless transmission of such information may not be reliable. Second, such an approach is not computationally efficient, because the complexity of computing the objective function for all possible configurations is large, which will be clear in Section IV. Thus, an efficient distributed algorithm is needed.

In order to reduce the complexity and design distributed algorithms, we exploit the structure of the objective function given in (9). It is clear that each $P_{MO,k}$ is a function of only those $s_i \in \mathcal{D}_k$, i.e., the assignments of local SUs, and such a property is reflected in sparsity of graph structure. Furthermore, the min-sum pair possesses the same algebraic structure (i.e., a commutative semiring) as the sum-product problem in the inference problem (see, e.g., [18], [19]). Thus, the elimination algorithm that solves the sum-product problem can be applied to design a message passing algorithm to solve our min-sum problem. The idea of the algorithm is as follows. The commutative semiring structure allows us to rewrite the objective function in (9) as

$$\min_{\{s_1,\ldots,s_J\}} \sum_{k=1}^{K} P_{MO,k} \tag{11}$$
$$= \min_{\{s_2,\ldots,s_J\}} \left\{ \sum_{k:1\notin D_k} P_{MO,k} + \min_{s_1} \sum_{k:1\in D_k} P_{MO,k} \right\}.$$

From this equation, we can first compute the minimization over s_1 (and hence eliminate s_1) by picking the best configuration for SU 1 for all other configurations of the s_js that are involved in $\min_{s_1} \sum_{k:1\in D_k} P_{MO,k}$. In this way, minimization is performed for configurations of variables only when necessary. This quantity is passed as a message from SU 1 to the next node, and the same procedure is followed to minimize over other variables one by one. A more detailed description of the elimination algorithm for solving the sum-product problem can be found in [18, Chapter 9].

In the following, we propose a *distributed* message passing algorithm adapted from the elimination algorithm for solving the sum-product problem. The algorithm contains two phases: in-phase and out-phase processes. The in-phase process performs minimization over all variables by passing messages from one SU to another. Each message is a data structure created by minimizing over the corresponding SU assignment variable s_i. However, none of the intermediate SU except the last node (i.e., the root) can figure out its own best assignment, because such an assignment may still depend on the assignments of remaining SUs. Only the last SU at the end of the in-phase process can determine its best assignment and the corresponding optimal objective function. The out-phase process finds the optimal assignments for all SUs. It starts from the root, and messages are transmitted from one SU to another in an inverse ordering with respect to the in-phase process. Each message contains the optimal assignments of the SUs that have already been obtained in the out-phase process. The process ends at the SU that starts the in-phase process. We describe the algorithm more formally as follows.

A distributed message passing algorithm

1. In-Phase (optimizing the objective function)
   - One SU, say SU 1, generates a function list consisting of $P_{MO,k}$ with $1 \in \mathcal{D}_k$, and minimizes the sum of the functions in the function list over s_1 for all configurations of values of other variables involved in these functions, i.e., all variables that connects to SU 1 in graphical representation.
   - SU 1 records the minimized objective function (as a function of other variables that connects to SU 1) as a new function $Q_{MO,1}$ and the corresponding optimizing s_1.
   - SU 1 informs all its neighboring nodes that it has finished the process, and has all neighboring nodes connect to each other. SU 1 then passes the new function $Q_{MO,1}$ to one of its neighbor.
   - The chosen SU adds $Q_{MO,1}$ to its function list, removes $P_{MO,k}$ containing variables of previous nodes from its function list, and then repeats the above steps.
   - The in-phase process ends when a chosen SU, say SU i, removes all $P_{MO,k}$ from its function list, and receives a function with only one variable (i.e., its own s_i). This SU is the last node, and the minimal value of the received function is hence the optimal objective function.
2. Out-Phase (finding the optimal assignment)
   - The last node SU i informs the node who previously passed a function to it in the in-phase process about its minimizing s_i^*. Then this node is able to obtain its own minimizing s_j^* based on its record in the in-phase process.
   - Repeat the above step until the first node SU 1 is reached. Now every node knows its optimizing assignment s_j^*.

Although networks in general may have loops, the above algorithm yields the exact global optimal solution. In the end, each SU knows only its own and its neighbors' optimal assignment.

Theorem 1: The above message passing algorithm yields the global optimal objective function and the corresponding optimal SU assignment within a finite number of steps (more specifically, the twice of the number of SUs in the network).

Proof: At each step of the proposed message passing algorithm, minimization over a variable is performed exactly over the sum of all $P_{MO,k}$ functions that involve this variable, which can be seen from (11). Hence, no approximation is made. This is similar to the elimination algorithm for the sum-product problem [18]. ■

We note that this algorithm is distributed in that the minimization is performed locally and the message passing ordering is chosen locally as well. This is different from the regular elimination algorithm for the sum-product algorithm that initially specifies an elimination ordering and constructs an active list based on the global information of the network. To carry out our algorithm, each SU needs to know only the information about its neighbors and does not need to know the global network information.

We also note that exchanging of local information in the algorithm between SUs can be implemented via a common control channel (CCC), which is predefined for all SUs to exchange system control information in dynamic spectrum access. The reader can refer to [24]–[26] for detailed information.

B. Graphical Interpretation of the Algorithm

The in-phase process of the proposed algorithm can be viewed as an elimination algorithm, i.e., an SU is eliminated as the message is passed from this SU to the next node. In terms of graphical representation, the initial elimination step involves the cliques in the graph, which include the first eliminating node. During the elimination of each node, a new function that includes the variables corresponding to the neighbors of the eliminating node as its arguments is created. Correspondingly, edges are added to all pairs of neighbors of the eliminating node, and a new clique is thus formed. Such a process (as implemented in the third step in the in-phase process of the algorithm) is called *triangulation* (see [18]), and it guarantees the following property.

Lemma 1: For any connected graph, the in-phase process of the given message passing algorithm ends when all nodes are visited.

Proof: The above lemma follows because the triangulation process keeps the network to be connected after elimination of each node, and hence there is always a node to be eliminated before the last node is reached. ■

C. The Min-Max Problem

To solve the min-max problem, we again exploit the structure of the objective function (10). Similar to the min-sum problem, we observe that the min-max problem also possesses the same structure as that of the min-sum problem. Hence, the distributed message passing algorithm proposed in Section III-A can be applied to solve the min-max problem with the operation "max" replacing the operation "sum" in the algorithm.

Remark 1: Although the min-max and min-sum problems can be solved using the algorithms similar to each other, the optimal assignments of SUs for the two problems are different and have different implications. The assignment of the SUs in the min-max problem needs to balance the probabilities of missed opportunity over all PUs, and hence treats all PUs in a fair manner, whereas the min-sum problem focuses on only the sum of probabilities, and may not treat all individual PUs equally.

IV. Complexity

In this section, we compare the complexity of the proposed algorithms with that of the exhaustive search approach. The complexity of the algorithm proposed in Section III-A for the min-sum problem includes the computation of the functions $P_{MO,k}$ for $1 \leq k \leq K$ and the minimization carried out at each SU. In the following theorem, we characterize the complexity of both types of computation.

Theorem 2: For a given message passing ordering, the number of computations (including addition, multiplication, and comparison) of the message passing algorithm implemented for the min-sum problem is given by $O(K|\mathcal{D}|2^{|\mathcal{D}|}) + O((K+J)v^B)$. Here $|\mathcal{D}| = \max_{1\leq k\leq K} |\mathcal{D}_k|$, v denotes the largest number of values that s_i can take over $1 \leq i \leq J$, and B denotes the largest number of variables involved in either $P_{MO,k}$ for $1 \leq k \leq K$ or $Q_{MO,j}$ for $1 \leq j \leq J$. If the structure of the objective function is not exploited (i.e., using the exhaustive search), the total number of computations is given by $O(2|\mathcal{D}|v^J) + O((K+1)v^J)$.

Proof: We first consider the computation of the function values of $P_{MO,k}$ for $k = 1, \ldots, K$. Each $P_{MO,k}$ is computed once for each configuration of values of variables. For each configuration, only s_j with $j = k$ contributes to the computation dimension, and hence the maximum dimension is at most $|\mathcal{D}_k|$. The total number of addition and multiplication involved is at most $2|\mathcal{D}_k|$. There are $2^{|\mathcal{D}_k|}$ configurations. Hence the total number of computations for K functions $P_{MO,k}$ is less than $O(2K|\mathcal{D}_k|2^{|\mathcal{D}_k|})$.

For a given message passing ordering, $Q_{MO,j}$ is the new function yielded by minimizing the sum of all $P_{MO,k}$ that involve s_j over s_j. Hence, entries of $Q_{MO,j}$ is less than v^B. Each of the functions $P_{MO,k}$ and $Q_{MO,j}$ is added once for each entry of $Q_{MO,j}$. Hence the total number of additions is less than $(K+J)v^B$. For each $Q_{MO,j}$, the number of comparisons is less than v^B, i.e., v times for each configuration of other variables.

If the structure of the objective function is not exploited, the computation of K functions $P_{MO,k}$ is $O(2|\mathcal{D}|v^J)$, and the computation of minimization is $O((K+1)v^J)$. ■

Remark 2: The definitions of $|\mathcal{D}|$ and B imply that $|\mathcal{D}| \leq B$. Hence, the complexity of the optimization dominates that of the computation of the objective functions. These two are comparable only for certain specific graphs, in which a message passing ordering does not introduce new edges in the triangulation process discussed in Section III-B.

Remark 3: For many networks, B is much smaller than J. Hence, the complexity of the message passing algorithm is substantially lower than that of the exhaustive search that does not exploit the structure of the objective function.

Remark 4: The complexity of the algorithm to solve the min-max problem is the same as the complexity of the algorithm that solves the min-sum problem, except that the counts for addition becomes the counts for comparison.

We further note that the complexity of the algorithm depends on the size of the functions that each message passing step involves, which depends on the in-phase message passing ordering. However, finding the best message passing ordering that achieves the lowest complexity is NP-hard [18]. Several approaches have been proposed to construct orderings that perform well in reducing complexity [18]. In particular, it has been shown in [18, Section 9.4] that for *chordal graphs* (i.e., graphs with every minimal loop having length three), the *Max-Cardinality* algorithm can be used to yield the best message passing ordering, which does not result in new edges during the process. Here, we provide a decentralized implementation of the Max-Cardinality algorithm.

A decentralized Max-Cardinality algorithm

- One SU, say SU 1, creates a table by recording its neighbors and, for each neighbor, records one as the marking value. Chooses any of its neighbors, and passes the table to this node.
- Being chosen as the next node, say SU i, creates a deleting list and adds SU 1 to this list, updates the table by adding its own neighbors, and records one as marking values for new nodes and adds one to the marking values for existing nodes. SU i chooses the node in the table with the largest marking value as next node.
- If the chosen node is SU i's neighbor, SU i passes the table and the deleting list to this node. If the chosen node is not SU i's neighbor, SU i passes the table and the deleting list to its previous node. Continue this process until the chosen node is found.
- The chosen node does the same step as SU i.
- A node identifies itself as the last node if it has only itself in the table and all its neighboring nodes in the deleting list. This node reverses the deleting list and declares the reversed deleting list as the message passing ordering.

We also note that the above algorithm is guaranteed to terminate with all nodes being deleted for connected graphs.

V. Computation of Beliefs

From the message passing algorithm given in Section III-A, it is clear that the last node (say, SU i) in the in-phase process computes the overall system performance (optimized over other SUs' assignment already) as a function of s_i, i.e., a function $f_i(s_i)$, chooses the value of s_i that minimizes $f_i(s_i)$, and obtains the optimal overall system performance. The function $f_i(s_i)$ is usually referred to as the *belief* (marginal distribution) in inference problems [18]. In the scenario considered in this paper, it characterizes the impact of choice (assignment) of SU i on the overall system performance. In general, the information of beliefs is important for all SUs in the system. For instance, given flexibility, some SU may have a preferred PU to detect, and can choose this PU if this choice does not dramatically sacrifice the overall system performance. If more than one SUs wish to deviate from their optimal choices, a game-theoretic framework can be adapted to study the interactions between the SUs. In this section, we study how to efficiently obtain the information of beliefs for all SUs in the network.

One possible approach is to implement the in-phase process of the message passing algorithm proposed in Section III-A for each SU with this SU being the last node. The total number of messages computed during this process for a tree graph is the number of SUs multiplied by the number of edges in the graph. Such an approach is not computationally efficient, because same messages are computed multiple times for different SUs. Furthermore, it is not easy to determine the message passing ordering such that the algorithm ends at a specific node via a decentralized algorithm. For graphs that are trees, an efficient belief propagation algorithm [18, Chapter 10] used for inference problems can be implemented to collect the exact belief information for each SU. The key idea is to exploit the fact that the message passed from one node to another in order to compute the belief for node A does not change when computing the belief for another node B as long as the message is passed along the same direction as the previous one. Therefore, to obtain the belief for all nodes in the system, messages need to be passed along each edge only twice (one for each direction). Hence, the total number of messages computed is only twice of the total number of edges. Thus, such an algorithm is considerably more efficient than the approach suggested above.

The above algorithm is exact for tree graphs, but not for graphs with circles. Since the graph in our problems has circles in general, a more general clique-tree algorithm [18] needs to be applied. In the clique-tree algorithm, a clique-tree is first constructed based on the original graph, and then messages are passed between cliques. The clique tree satisfies the following properties: (1) family preserving, i.e., each factor ($P_{MO,k}$) must have all argument variables in one of the cliques; (2) running intersection property, i.e., if a variable X is in two cliques, it must be in all cliques that connects these two cliques. In general, clique trees can be constructed by the elimination algorithm. Details can be found in [18, Chapter 10]. Fig. 3 depicts a clique tree constructed for the example cognitive network in Fig. 1, which satisfies the family preserving and running intersection properties.

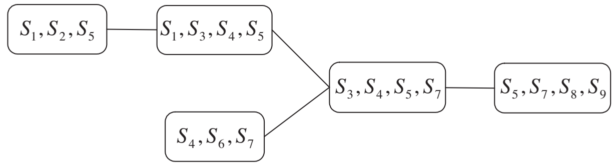


Fig. 3. A clique tree of the example cognitive network in Fig. 1

Based on the clique tree constructed, the clique-tree algorithm specifies the messages passed between the cliques (nodes) in order to compute the beliefs for all cliques and furthermore for each variable (each SU) in the cliques. In the following, we describe an implementation of the clique-tree algorithm to obtain beliefs for the SUs in the cognitive radio network based on the min-sum objective function. This algorithm can be modified for the min-max objective function with "sum" being replaced by "max".

A clique-tree algorithm to compute beliefs for the min-sum problem

1. Forming a clique tree using the message passing algorithm given in Section III-A:
   - One SU, say SU 1, forms the first clique C_1 that includes all its neighboring nodes. Claims a set $I_1 = \{k : 1 \in \mathcal{D}_k\}$, i.e., the indices of $P_{MO,k}$ with s_1 as one of its arguments. Defines the *initial potential function*
   $$\Psi_1 = \sum_{k \in I_1} P_{MO,k}.$$
   In the original graph (not the clique tree), connects all pairs of its neighboring nodes. Eliminates SU 1. SU 1 picks the next node. Informs the next node I_1.
   - Suppose the first $i-1$ steps eliminate nodes $1, \ldots, i-1$.
   - Suppose SU $i-1$ picks SU i. SU i forms a clique C_i in clique tree that includes all its neighboring nodes in the original graph. Claims a set $I_i = \{k \notin \cup_{a=1}^{i-1} I_a : i \in \mathcal{D}_k\}$. Defines
   $$\Psi_i = \sum_{k \in I_i} P_{MO,k}.$$
   In the original graph, connects all pairs of its neighboring nodes. Eliminates SU i. SU i picks the next node. Informs the next node $\cup_{a=1}^{i} I_a$.
   - Repeat the above step until the last node is reached.
2. Asynchronous clique tree algorithm
   - Initiate the algorithm at any clique, then each clique transmits a message to its neighbor after it has received messages from all other neighbors.
   - The message $\delta_{i \to j}$ from the clique C_i to C_j is given by
   $$\delta_{i \to j} = \min_{x_{C_i - C_i \cap C_j}} \left[\Psi_i + \sum_{j' \neq j : j' \in N_i} \delta_{j' \to i} \right]$$
   where N_i includes the indices of the neighboring cliques of C_i.
3. Computation of beliefs

- At each node, say SU i, its belief function is computed by

$$f_i(s_i) = \min_{x_{C_i - \{i\}}} \left[\Psi_i + \sum_{j' \in N_i} \delta_{j' \to i} \right]$$

Remark 5: Once an SU makes its choice based on its belief $f_i(s_i)$, this node takes the clique to which it belongs as the root and passes out its choice to the leaves so that all nodes can figure out their assignments of PUs in order to minimize the objective function given the root node has taken the current choice.

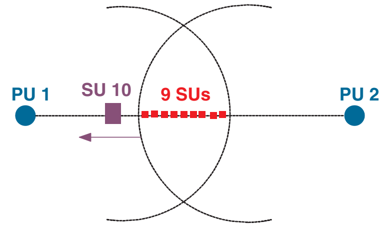
We note that part 1 of the above algorithm is a distributed implementation of the clique tree construction algorithm. In our implementation, the message passing algorithm involved and the information transmitted do not depend on the global system information. We also note that such an implementation constructs only a specific type of trees, which is the line graph. In general, with centralized information, the trees constructed may have multiple branches, Fig. 3 being one example.

VI. Numerical Results

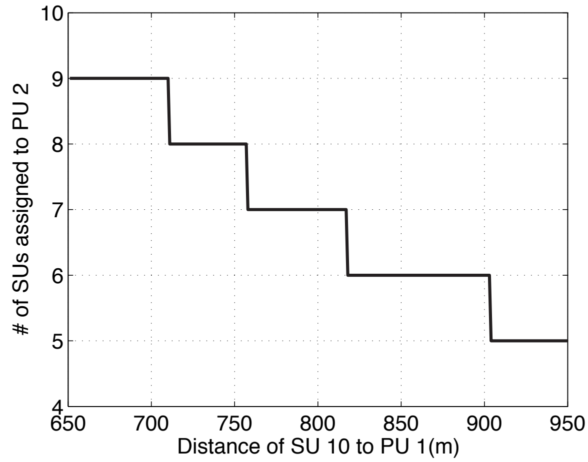
In this section, we implement algorithms proposed in this paper for some example cognitive radio networks and illustrate our results with numerical simulations.

The first scenario we consider is shown in Fig. 4 (a). Two PUs (PUs 1 and 2) are located 1950 meters away from each other, each with detection radius 1000 meters. Hence, the two PUs' detection ranges overlap with each other. We also assume that there are 9 SUs uniformly distributed in this overlapping area. SU 10, however, is located only in the detection range of PU 1. The channel model parameters are chosen as $\eta_1 = \eta_2 = 1150$ meters and $b = 3$. The probability of interference is required to be less than $\gamma = 0.0005$. From Fig. 4 (b), it is clear that as the distance of SU 10 to PU 1 decreases (i.e., as SU 10 moves toward PU 1), the number of SUs assigned to PU 2 increases. This is because SU 10 monitors PU 1 better as it moves closer to PU 1, and hence more SUs switch to detect PU 2 to minimize the total probability of the missed opportunity. Fig. 4 (c) plots how the total and individual probabilities of missed opportunities change with the location of SU 10. As SU 10 moves closer to PU 1, the probability of missed opportunity for PU 1 decreases in general. It can be also seen that a jump occurs whenever an SU switches to detect PU 2, which causes the probability of missed opportunity for PU 1 to increase. The probability of missed opportunity for PU 2 decreases as SU 10 moves toward PU 1, and a decreasing jump occurs whenever an SU switches to detect PU 2. The total probability of missed opportunity decreases as SU 10 moves toward PU 1, because SU 10 serves PU 1 better and hence helps improve the overall system performance.

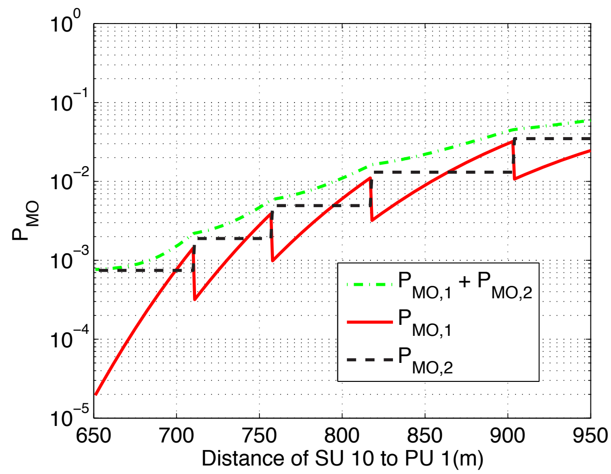
For scenario 1 also shown in Fig. 5 (a), we further consider the min-max problem. In Fig. 5 (b), it can be seen that as the distance of SU 10 to PU 1 decreases (i.e., if we view the plot from the right to the left), a larger number of SUs switch to serve PU 2 similar to how they behave for the min-sum problem. However, each SU switches at a larger distance



(a): Network configuration



(b): # of SUs assigned to PU 2 vs the position of SU 10

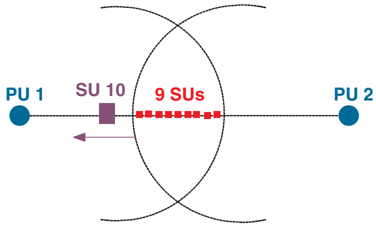


(c): The probabilities of the missed opportunity

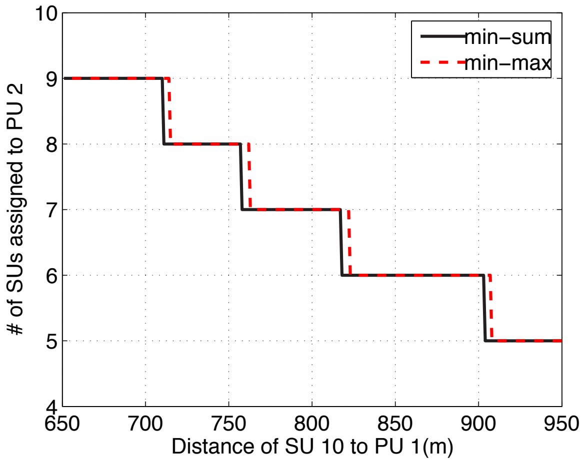
Fig. 4. Scenario 1 with two PUs and ten SUs for the min-sum problem

(the dash line) for the min-max problem than that (the solid line) for the min-sum problem. This is to prevent the gap between $P_{MO,1}$ and $P_{MO,2}$ (as shown in Fig. 5 (c)) from further increasing. This demonstrates that the solution to the min-max problem tends to balance $P_{MO,1}$ and $P_{MO,2}$ for the two PUs.

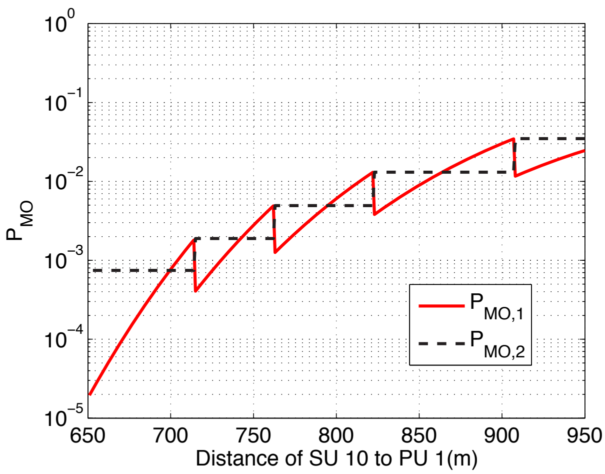
Scenario 2 we consider is a lattice network with a number of square cells (see Fig. 6 (a) and Fig. 7 (a)). There is one PU located at the center of each square cell with the side-length



(a): Network configuration

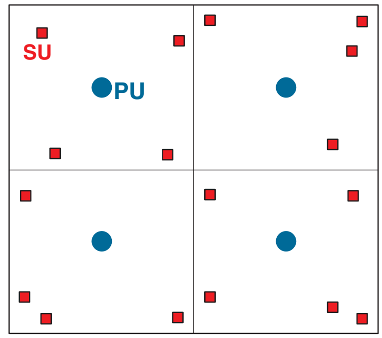


(b): # of SUs assigned to PU 2 vs the position of SU 10

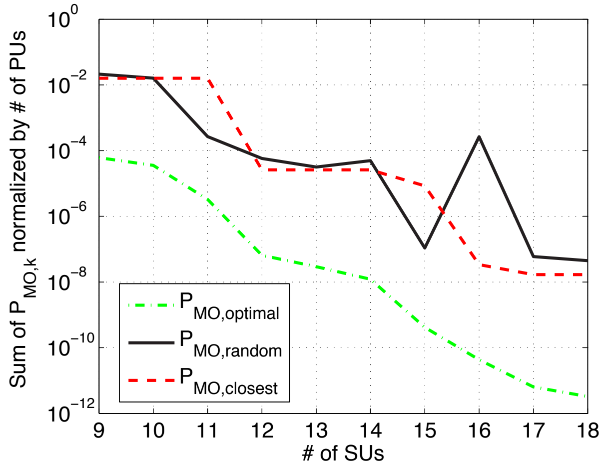


(c): The probabilities of the missed opportunity

Fig. 5. Scenario 1 with two PUs and ten SUs for the min-max problem



(a): Network configuration

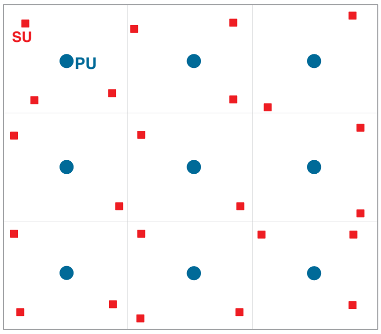


(b): Comparison of sum P_{MO} for three SU assignments

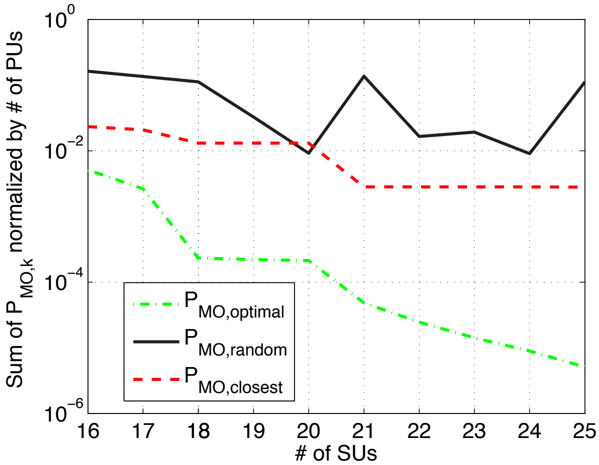
Fig. 6. Scenario 2 with four PUs

being 1200 meters. Each PU has a detection radius of 1000 meters. The SUs are randomly located around the vertices of the cells. Hence, all SUs located around the vertices of a cell are in the detection range of the PU in this cell. The channel model parameters are chosen to be $\eta_k = 1200$ meters for all k and $b = 3$. The probability of interference is required to be less than $\gamma = 0.0005$. For the lattice network, we consider the min-sum problem. Fig. 6 (a) shows a lattice network with four cells and four PUs in the center of the cells. Fig. 6 (b) plots the sum of probabilities of missed opportunity for three different SU assignment schemes, namely, 1) the optimal assignment obtained using the message passing algorithm developed in this paper, 2) the closest selection scheme in which each SU chooses the closest PU to detect, and 3) the random selection scheme in which each SU randomly chooses one PU to detect. It is clear from the figure that the optimal assignment scheme has a much smaller sum of probabilities of missed opportunity than that of the other two assignment schemes, and hence exploits the spectrum considerably more efficiently. It can also been seen that the sum of the probabilities of the other two assignment schemes are close to each other, with the closest selection scheme having a slightly better performance. The advantage of the optimal assignment scheme is also demonstrated in Fig. 7 (b) for the network with nine cells (shown in Fig. 7 (a)). From this figure, we can also see that the advantage of the closest selection scheme over the random selection scheme appears more clearly as the network size becomes larger.

For lattice networks, we also compare the program running time of the message passing algorithm with that of the exhaustive search. For the lattice network with 9 PUs and 16 SUs, the message passing algorithm takes 0.2496s, which is ten times faster than the exhaustive approach that takes



(a): Network configuration



(b): Comparison of sum P_{MO} for three SU assignments

Fig. 7. Scenario 2 with nine PUs

3.1356s. Our simulation also demonstrates that the advantage of the message passing algorithm is more significant as the network size becomes large.

VII. Conclusions

We have formulated and studied a network management problem, i.e., the assignment of SUs to PUs for collaborative detection that results in the best exploitation of available spectrum in cognitive radio networks. We have proposed a distributed message passing algorithm to derive the optimal configuration of SUs by exploiting the connection between the problem under study and the inference problem studied via probabilistic graphical models. We have shown that the complexity of the algorithm is significantly smaller than that of the exhaustive search approach. We have also applied the clique-tree algorithm to efficiently compute the beliefs (impacts of each SU's choice of PU on the overall system performance) for all SUs. Our simulation results have demonstrated the substantial gains obtained by optimizing over the assignments of SUs for cognitive radio networks.

We note that many other important issues can be addressed by applying the algorithms proposed in this paper. For example, the mobility of SUs may be incorporated. The case when multiple PUs simultaneously use the same frequency band by applying schemes such as CDMA or FDMA is also interesting. In this case, the solutions to the detection problem and the user assignment problem may depend on the specific schemes that the PUs are using.

References

[1] Y. Liang, L. Lai, and J. Halloran, "Distributed algorithm for collaborative detection in cognitive radio networks," in *Proc. Allerton Conf. on Communication, Control, and Computing*, Monticello, IL, Sept. 2009.

[2] J. Mitola, "Cognitive radio: Making software radios more personal," *IEEE Pers. Commun.*, vol. 6, pp. 13–18, Aug. 1999.

[3] S. Haykin, "Cognitive radio: Brain-empowered wireless communications," *IEEE J. Sel. Areas Commun.*, vol. 23, pp. 201–220, Feb. 2005.

[4] Y-C. Liang and Y. Zeng and E. Peh and A. Hoang,"Sensing-Throughput Tradeoff for Cognitive Radio Networks", *IEEE Trans. Wireless Commun.*, vol. 7, no. 4, pp. 1326-1337, Apr. 2008.

[5] A. Ghasemi and E. S. Sousa, "Collaborative spectrum sensing in cognitive radio networks," in *Proc. IEEE Intl. Symposium on New Frontiers in Dynamic Spectrum Access Networks*, Baltimore, MD, Nov. 2005, pp. 131–136.

[6] G. Ganesan and Y. G. Li, "Cooperative spectrum sensing in cognitive radio, part I: Two user networks," *IEEE Trans. Wireless Commun.*, vol. 6, no. 6, pp. 2204–2213, June 2007.

[7] ——, "Cooperative spectrum sensing in cognitive radio, part II: Multiuser networks," *IEEE Trans. Wireless Commun.*, 2007, submitted.

[8] G. Ganesan, Y. G. Li, B. Bing, and S. Li, "Spatiotemporal sensing in cognitive radio networks," *IEEE J. Sel. Areas Commun.*, vol. 28, no. 1, pp. 5–12, Jan. 2008.

[9] S. M. Mishra, A. Sahai, and R. W. Brodersen, "Cooperative sensing among cognitive radios," in *Proc. IEEE Intl. Conf. on Communication*, vol. 4, Istanbul, Turkey, June 2006, pp. 1658–1663.

[10] Z. Quan, S. Cui, and A. H. Sayed, "An optimal strategy for cooperative spectrum sensing in cognitive radio networks," in *Proc. IEEE Global Telecommunications Conf.*, Washington DC, Nov. 2007, pp. 2947–2951.

[11] ——, "Optimal linear cooperation for spectrum sensing in cognitive radio networks," *IEEE J. Special Topics Signal Process.*, vol. 2, no. 1, pp. 28 – 40, Feb. 2008.

[12] E. Vistotsky, S. Kuffner, and R. Peterson, "On collaborative detection of TV transmissions in support of dynamic spectrum sharing," in *Proc. IEEE Intl. Symposium on New Frontiers in Dynamic Spectrum Access Networks*, Baltimore, MD, Nov. 2005, pp. 338–345.

[13] J. Unnikrishnan and V. V. Veeravalli, "Cooperative sensing for primary detection in cognitive radio," *IEEE J. Special Topics Signal Process.*, vol. 2, no. 1, pp. 18 – 27, Feb. 2008.

[14] J. Hillenbrand, T. Weiss, and F. Jondral, "Calculation of detection and false alarm probabilities in spectrum pooling systems," *IEEE Commun. Lett.*, vol. 9, pp. 349–351, Apr. 2005.

[15] W. Zhang and K. B. Letaief, "Cooperative spectrum sensing with transmit and relay relay diversity in cognitive radio networks," *IEEE Trans. Wireless Commun.*, vol. 7, pp. 4761–4766, Dec. 2008.

[16] W. Zhang, R. K. Mallik, and K. B. Letaief, "Cooperative spectrum sensing optimization in cognitive radio networks," in *Proc. IEEE Intl. Conf. on Communication*, Beijing, China, May 2008, pp. 3411–3415.

[17] K. B. Letaief and W. Zhang, "Cooperative communications for cognitive radio networks," *Proc. IEEE*, vol. 54, no. 6, pp. 2493–2507, June 2008.

[18] D. Koller and N. Friedman, *Probabilitic Graphical Models: Principles and Techniques*. Cambridge, MA: The MIT Press, 2009.

[19] M. J. Wainwright and M. I. Jordan, "Graphical Models, Exponential Families, and Variational Inference," *Foundations and Trends in Machine Learning*, vol. 1, nos 1-2, pp. 1–305, Now Publishers Inc., Hanover, MA, USA, 2008.

[20] H.-B. Chang and K.-C. Chen, "Auction-Based Spectrum Management of Cognitive Radio Networks," *IEEE Trans. Veh. Technol.*, vol. 59, no. 4, pp. 1923-1935, May 2010.

[21] C. Lee and W. Lee, "Exploiting Spectrum Usage Patterns for Efficient Spectrum Management in Cognitive Radio Networks" in *Proc. IEEE International Conference on Advanced Information Networking and Applications*, Perth, Australia, Apr. 2010.

[22] I. F. Akyildiz and W.-Y. Lee and K. R. Chowdhury, "Spectrum Management in Cognitive Radio Ad Hoc Networks," *IEEE Network*, vol. 23, no. 4, pp. 6-12, June/July 2009.

[23] H. V. Poor, *An Introduction to Signal Detection and Estimation*. New York: Springer-Verlag, 1994.

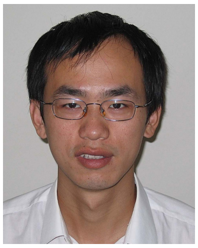
[24] V. Brik, E. Rozner, S. Banerjee and P. Bahl, "DSAP: A protocol for coordinated spectrum access," in *Proc. IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks (DySPAN)*, Baltimore, Maryland, USA, Nov. 2005, pp. 611-614.

[25] L. Ma, X. Han and C.-C. Shen, "Dynamic open spectrum sharing MAC protocol for wireless ad hoc networks," in *Proc. IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks (DySPAN)*, Baltimore, Maryland, USA, Nov. 2005, pp. 203-213.

[26] S. Sankaranarayanan, P. Papadimitratos, A. Mishra and S. Hershey "A bandwidth sharing approach to improve licensed spectrum utilization," in *Proc. IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks (DySPAN)*, Baltimore, Maryland, USA, Nov. 2005, pp. 279-288.

Yingbin Liang (S'01–M'05) received the Ph.D. degree in Electrical Engineering from the University of Illinois at Urbana-Champaign in 2005. In 2005-2007, she was working as a postdoctoral research associate at Princeton University. In 2008-2009, she was an assistant professor at the Department of Electrical Engineering at the University of Hawaii. Since December 2009, she has been an assistant professor at the Department of Electrical Engineering and Computer Science at the Syracuse University. Dr. Liang's research interests include communications, wireless networks, and information theory.

Dr. Liang was a Vodafone Fellow at the University of Illinois at Urbana-Champaign during 2003-2005, and received the Vodafone-U.S. Foundation Fellows Initiative Research Merit Award in 2005. She also received the M. E. Van Valkenburg Graduate Research Award from the ECE department, University of Illinois at Urbana-Champaign, in 2005. In 2009, she received the National Science Foundation CAREER Award, and the State of Hawaii Governor Innovation Award.



Lifeng Lai (M'07) received the B.E. and M. E. degrees from Zhejiang University, Hangzhou, China in 2001 and 2004 respectively, and the PhD degree from the The Ohio State University at Columbus, OH, in 2007. He was a postdoctoral research associate at Princeton University from 2007 to 2009. He is now an assistant professor at University of Arkansas, Little Rock. Dr. Lai's research interests include wireless communications, information security, and information theory.

Dr. Lai was a Distinguished University Fellow of the Ohio State University from 2004 to 2007. He co-authored a paper that received the Best Paper Award from IEEE Global Communications Conference, 2008.

John Halloran recieved the B.S. degrees in Mathematics and Electrical Engineering from Seattle University in 2008, and the M.S. degree in Electrical Engineering from the University of Hawaii at Manoa in 2010. He is currently pursuing the Ph.D. degree in Electrical Engineering at the University of Washington. His research interests include wireless communications, information theory, and data compression.