# A Factor Graph Based Dynamic Spectrum Allocation Approach for Cognitive Network

Shu Chen, Yan Huang
Department of Computer Science & Engineering
Universities of North Texas
Denton, Texas 76201
Email: shuchen@unt.edu, huangyan@unt.edu

Kamesh Namuduri
Department of Electrical Engineering
Universities of North Texas
Denton, Texas 76201
Email: kamesh.namuduri@unt.edu

***Abstract*—Cognitive radios share the whitespace in the absence of licensed users. For spectrum sharing, avoiding the interference among multiple secondary users is a fundamental problem (known as dynamic spectrum allocation problem). This paper proposes a factor graph based approach to solve the dynamic spectrum allocation problem in an efficient manner. With the proposed DWA-Tree (Distributed Wave Algorithm), this problem can be solved with $2n$ number of messages for tree structured graphs. This paper also proposes another novel algorithm called DWA-Cycle to solve the problem for general graphs with cycles. Simulation results show that both DWA-Tree and DWA-Cycle can improve the global link quality consistently better than naive local optimization approaches.**

## I. Introduction

### A. Cognitive Network

CN (Cognitive Network) is a network with a cognitive process that perceives current network conditions, acts on those conditions, and learns from the consequences of its actions following end-to-end goals [1] that include:

- sensing environment,
- learning and cooperating,
- self-Organize and software-defined reconfigurability,
- flexibility and agility.

Mitola [2] generalized the idea of cognitive radios that have the ability to sense the external environment and simultaneously adjust their states (such as channel, or other transmission parameters). For many years, FCC licensing has yielded false scarcity of spectrum. The reformed of FCCs policy gives unlicensed user more flexibility to share the whitespace spectrum with existing licensed users. This change creates a new challenge for dynamic spectrum sharing among secondary users.

### B. Factor Graph and Belief Propagation

A factor graph is a bipartite graph representing the factorization of a function which enables efficient computation of marginal distributions. Given a factorization:

$$g(X_1, X_2, ..., X_n) = \prod_{j=1}^{m} F_j(S_j), \qquad (1)$$

in which $S_j \subseteq X_1, X_2, ..., X_n$, the corresponding factor graph $G = (X, F, E)$ consists of variable vertices $X = X_1, X_2, ..., X_n$ , factor vertices $F = F_1, F_2, ..., F_n$ , and edges $E$.

The edges depend on the factorization as follows: there is an undirected edge between factor vertex $F_j$ and variable vertex $X_k$ when $X_k \in S_j$. Figure 1 illustrates the factor graph corresponding to the following function:

$$g(X_1, X_2, X_3) = F_1(X_1)F_2(X_1, X_2)F_3(X_1, X_2)F_4(X_2, X_3). \qquad (2)$$

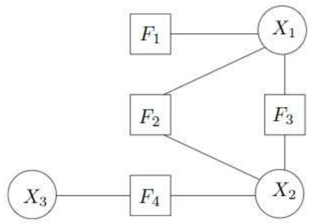


Fig. 1. A Factor Graph Corresponding to 2

The factor graph also belongs to Bayesian network and satisfies the local Markov property that each variable is conditionally independent with each other. If $X = X_1, X_2, ..., X_n$ is a set of discrete random variables with a joint mass function $p$ for a given DAG (directed acyclic graph) $G' = (X, E)$, the marginal distribution $P$ of a single variable $X_i$ is simply the summation of $p$ over all other variables as:

$$P(X_i) = \sum p(X_1, X_2, ..., X_{i-1}, X_{i+1}, ..., X_n) \qquad (3)$$

This quickly becomes computationally prohibitive since for 100 binary variables needs to sum over $2^{99} \approx 6.338 \times 10^{29}$ possible values.

By exploiting the factor graph, belief propagation allows marginals to be computed much more efficient.

### C. Motivation

First, the distances and power levels vary among nodes which makes the interference levels differ from one transmitter to another for any receiver. Second, local optimization does not work for global optimization. Even if the interference of any individual receiver is minimized, the global interference is not. For these reasons, naive approaches will not be effective in minimizing the global

978-1-61284-254-7/11/$26.00 ©2011 IEEE 

interference. If we assume that any transmitter has a probability distribution on channel usage (e.g. using some scheduling algorithms), the spectrum allocation for a single user is based on the marginal distribution and the summation computation. A belief propagation method could solve the DSA (dynamic spectrum allocation) problem efficiently for a given network, and improve the global link quality.

### D. Contributions

This paper has the following contributions. First, it presents a factor graph based model, abstracting the dynamic spectrum allocation problem to a belief propagation problem in the graph. Second, it proposes novel Distributed Wave algorithm (DWA) to solve the problem with $2n$ messages for tree structures and $2(n + c)$ messages for general graphs with cycles, where $n$ is the number of nodes and $c$ is the number of cycles. Simulation results show that the proposed algorithms can consistently achieve higher global link quality than the local optimization approach with or without cycles.

### E. Organization

The rest of the paper is organized as follows: the dynamic spectrum allocation problem is described in Section III with the related works in Section II. Section IV provides our graphical model. Section V and VI present DWA algorithms with simulation results. Section VII concludes the paper and outlines future research directions.

## II. Related Work

Recently, there has been a significant amount of research in the area of cognitive network and radios. The idea of the cognitive network is first introduced by Mitola in [3]. A good discussion on cognitive radios is given in ([4]). An information-theoretic approach on cognitive radios is presented in [5]. These are some of the articles that provide a fundamental background for our research.

An introduction of factor graphs and sum-product algorithm can be found in [6]. And an adaptive approach for factor graph inference propagation problem is presented in [7] using a cluster method. Some other approaches including [8] and [9] also focus on providing an adaptive Bayesian inference framework to maintain marginal distributions. Our work extends the idea of sum-product algorithm and provides a framework to solve the dynamic spectrum allocation problem which emerges with the sensing ability of cognitive radios to spectrum sharing.

Dynamic channel assignment or link scheduling problem in multi-radio wireless mesh networks are discussed in [10] and [11]. However, their research focus on how to solve the link scheduling problem with multi-radio (multi-channel) to improve throughput or other measurements of the network. Instead, our approach focuses on avoiding the interference among multiple secondary users.

The closest research to our work is [12]. It makes use of Nash etiquette to solve the channel allocation problem and assumes users (players) to have full knowledge of other users' (players') actions. The purpose is to find a convergence point to maximize the set of utility functions. Our approach gives a framework when users do not have knowledge of the whole network to fit a more practical case, and uses a graphical model instead of regret minimization learning.

To our knowledge, our work is the first factor graph-based approach to solve the dynamic spectrum allocation problem.

## III. Problem Statement

### A. Problem Definition

The problem setting is as follows:

- Since one bidirectional communication can be divided to two one way communications between two different pairs of transmitters and receivers. We only consider one way communication.
- Each node in the abstracted graph is assumed to have a transmitter and receiver.
- We assume all nodes are cooperatively to achieve a fair usage of the shared channel.
- We assume the nodes' density is acceptable. For a high density scenario, no approach can achieve a good performance. And the problem is trivial for a low density scenario.
- We assume the network is reliable (no missing messages).
- We assume the noise is equally distributed to simplify the problem.

Based on these settings (constraints), the dynamic spectrum allocation problem can be formalized as:

**DSA Problem 1.** *Given $n$ number of nodes, $m$ number of available frequency channels $C_1, C_2, ..., C_m$, find an efficient way to allocate the communication channels for all pairs of nodes to minimize the interference ($\sum_{i=1}^{n}\sum_{j=1}^{n} I_{i,j}$ where $i, j \in n$ and $i \neq j$) among them.*

The problem is trivial if we have enough channels ($n >> m$).

### B. System Measurement

Network performance can be measured using different methods such as throughput and delay. A cognitive network can be considered as a collection of interfering pairs of nodes. From this point of view, we consider the minimum interference or the best global communication quality, as our main performance metric.

The SIR (signal-to-interference-ratio) at receiver $j$ ($i$ is corresponding transmitter for $j$) can be expressed as:

$$SIR_{ij} = \frac{R_i G_{ij}}{\sum_{k=1,k\neq i}^{n} R_k G_{kj} I(k,i)}, \tag{4}$$

where $R_i$ is the power level of $i$, and $G_{ij}$ is the channel gain. $I(k, i)$ is the interference from the transmitter $k$ to $i$ described by:

$$I(k,i) = \begin{cases} 0 \ (when\ C_k \neq C_i) \\ 1 \ (when\ C_k = C_i), \end{cases} \tag{5}$$

in which $C_i$ and $C_k$ are the channel selections for transmitter $i$ and $k$ respectively. The global communication quality is the summation of all pairs. We use the average SIR, $\frac{\sum_i^n \sum_{j \neq i}^n SIR_{ij}}{n}$, as our metric to measure the transmission link quality of the network.

## IV. Preference Distribution and Graph Generation

### A. Graph Generation

A factor graph has two types of nodes: factor nodes and variable nodes. It can be generated based on the interference relationship among nodes. A factor node exists if two or more nodes have interference relationship. To simplify the graph, factor nodes can be abbreviated to and represented as a single edge between two nodes. We assume each node $j$ has knowledge of its power level $R_i$ ($i$ is the corresponding transmitter) and the channel gain $G_{kj}$ between its neighboring nodes $k$ and itself. Since the channel gain is different for transmitter $k$ to receiver $j$ and transmitter $j$ to receiver $k$, we use $W(k, j)$ as the weight of factor edges from $k$ to $j$ as $W(k, j) = R_k G_{kj}$.Two asymmetric edges are created between two interfering nodes with different weight. We consider edges with weight lower than $\frac{R_{average} G_{average}}{\phi}$ ($R_{average}, G_{average}$ are average power level and channel gain for all nodes, and $\phi$ is an experimental factor between 10 to 1000), as trivial edges and remove them during the graph generation. Figure 2 shows a graph in which nodes $1, 2$ and 3 interfere with each other, node 4 interferes with nodes $3, 5$, and node 5 only interferes with node 4.

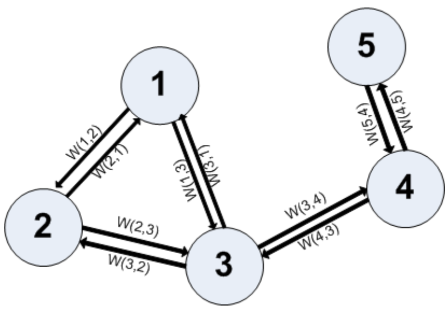


Fig. 2. Graph Generation Example

### B. Preference Distribution Generation

Each node has its own preference probability distribution on each possible channel. It is based on the following general environmental information:

- current usage,
- channel noise level (determined by sensing channels),
- channel occupation rate (licensed users may occupy specific channel), and
- usage preference (adjusted based on previous transmission).

The preference probability distribution is a vector of all $m$ frequency channels with a summation of 1. Consider for example, a graph with 5 nodes and 3 channels using the same topology as in Figure 2. Assume that node 1 first initializes its preference vector as shown in Table I. The power level and channel gain settings are in Table II.

| Channel | Preference Value |
|---|---|
| 1 | 0.5 |
| 2 | 0.3 |
| 3 | 0.2 |

TABLE I
CHANNEL DISTRIBUTION

| Node | Power Level | Channel Gain |
|---|---|---|
| 1 | 2 | 0.8 |
| 2 | 1 | 0.5 |
| 3 | 4 | 0.5 |

TABLE II
NODE PARAMETERS

## V. Distributed Wave Algorithm

In this section, we describe our distributed wave algorithm (DWA) to solve the dynamic spectrum allocation problem for cognitive networks. The DWA algorithm has two versions: (1) DWA-Tree, which solves the problem in tree structures; and (2) DWA-Cycle, which extends DWA-Tree to support general graphs with cycles. Both of them are based on the generated abbreviated factor graph.

### A. DWA-Tree

DWA-Tree is a modified version of general sum-product algorithm. At first, it generates the maximum interference spanning tree (MIST) based on the factor tree graph and edge weights. Since the weights is corresponding to the two directions are different, the larger one will be selected as the edge weight for computation, which is based on the assumption that all nodes are cooperative and the aim is to improve the global link quality instead of individual link quality. During the MIST computation, we pick the node with minimum distances to all leaf nodes as the root node, by which all leaf nodes can reach the root node with minimum hops. The height and parent relationship are marked, and low weight edges are removed. Then, with the given MIST, DWA-Tree starts the two steps process described below:

- Forward Step: Forward step starts from all leaf nodes. A leaf node forwards its preference distribution to its parent. Any node receives a forward message will calculate the summation of all children combining with its own preference distribution as $P_{j'} = \frac{\sum_{m \in Child_j} (P_m) \times W_{m,j}}{COUNT(Child_j)} \times (1 - P_j)$, and forwards the result $P_{j'}$ to its parent unless it is the root node. The root node decides its spectrum allocation by selecting the channel with the lowest probability of $P_{root'}$, and sets the channel allocation vector $C_{root} = MINof(P_{root'})$ where $MINof$ sets the lowest probability in vector $P_{j'}$. The forward step ends when all forward messages reach the root node.
- Backward Step: Backward step starts from the root node after it selects the highest probability and sets it to $C_{root}$. It sends $C_{root}$ back to all its children nodes. Any node that receives a backward message calculates the summation based on the parent's allocation as $P_{j'} = \frac{\sum_{m \in Child_j, Parent_j} (P_{m'}) \times W_{m,j}}{COUNT(Child_j)+1} \times (1 - P_j)$, chooses the channel with the minimum probability, sets $C_j = MINof(P_{j'})$, and sends $C_j$ with backward

messages to its children unless it is a leaf node. Leaf nodes use a similar method to allocate their channels. The backward step ends when backward messages reach all leaf nodes.

The detailed DWA-Tree is described in Algorithm 1.

---

**Algorithm 1** Distributed Wave Algorithm For Tree

---

**Require:** 1) Given a channel set $C = C_1, C_2 ..., C_m$
2) Given a factor graph $G$ with $n$ vertices as number of nodes (pairs of transmitter and receiver) and directed weighted edges
3) Initializes $W_{i,j} = MAX(W(i,j), W(j,i))$ as the Weight of edge $E_{i,j}$ and nodes' preference probabilities $P_j (j \in n)$

Building MIST $G'$ of $G$ with the Root node $Root$ and leaf nodes $Leaf$
**On** $Leaf$ **side:**
Let $P_{j'} = P_j$
Send $P_{j'}$ to $Parent_j$ with *FORWARD* message
**if** Receive *BACKWARD* messages from Parent nodes **then**
  Let $P_{j'} = (C_{Parent'_j}) \times (1 - P_j)$, $C_j = MINof(P_{j'})$
**end if**
**On** $Root$ **side:**
**if** Receive *FORWARD* messages from all child nodes **then**
  Let $P_{j'} = \frac{\sum_{m \in Child_j} (P_{m'}) \times W_{m,j}}{COUNT(Child_j)+1} \times (1 - P_j)$, $C_j = MINof(P_{j'})$
  Send $C_j$ to $Child_j$ with *BACKWARD* message
**end if**
**On** $intermedia$ **side:**
**if** Receive *FORWARD* messages from all Child nodes **then**
  Let $P_{j'} = \frac{\sum_{m \in Child_j} (P_m) \times W_{m,j}}{COUNT(Child_j)} \times (1 - P_j)$
  Send $P_{j'}$ to $Parent_j$ with *FORWARD* message
**end if**
**if** Receive *BACKWARD* messages from Parent nodes **then**
  Let $P_{j'} = \frac{\sum_{m \in Child_j, Parent_j} (P_{m'}) \times W_{m,j}}{COUNT(Child_j)+1} \times (1 - P_j)$, $C_j = MINof(P_{j'})$
  Send $C_j$ to $Child_j$ with *BACKWARD* message
**end if**

---

*B. DWA-Cycle*

In a more general case, if cycles exist, DWA-Tree will fail due to the lost edge information while building the MIST. To solve this problem, we provide our DWA-Cycle algorithm. Similar to the DWA-Tree, DWA-Cycle also starts from the generated factor graph and creates the MIST. Instead of simply removing low weight edges during MIST generation, it marks all edges with the weight larger than the threshold. Another difference is that the root node is picked based on the peer relationship, and only peer edges can be removed during the generation of MIST. We will describe the peer relationship in V-C.

The algorithm also has two steps:

- Forward Step: The Forward Step is similar to that of DWA-Tree, while the MIST is built with mark and recover feature.
- Backward Step: Backward also starts from the root node. The root node sends the channel allocation $C_{root}$ back to all its children nodes. When a node receives a backward message, it first checks whether it has edges in remove set $Remove_{i,j}$, and recovers all removed edges with peer relationship. It calculates the channel distribution and allocates the channel setting based on the parent's allocation, peer recovered edges' preference distribution, and the summation of all children nodes as $C_j = MINof(P_{j'}) = MINof(\frac{\sum_{m \in Child_j, Parent_j, Peer_j} (P_{m'}) \times W_{m,j}}{COUNT(Child_j)+1} \times (1 - P_j))$. The allocation follows root to leaf order using the Sub-MIST built on nodes with peer relationship to prevent them from ignorantly allocating the same channel. Then it sends backward messages to its children unless the node itself is a leaf node. Leaf nodes use a similar method to allocate their channels. The backward step ends when backward messages reach all leaf nodes.

The detailed algorithm is described in Algorithm 2.

---

**Algorithm 2** Distributed Wave Algorithm For Cycle

---

**Require:** 1) Similar as DWA-Tree
Building MIST $G'$ of $G$ with $k$ levels by moving $E_{i,j}$ with lowest $W(i,j)$ from $G$ to $Remove_{i,j}$
**On** $Leaf$ **side and** $Root$ **side:**
Same as DWA-Tree
**On** $intermedia$ **side:**
**if** Receive *FORWARD* messages from all Child nodes **then**
  Let $P_{j'} = \frac{\sum_{m \in Child_j} (P_m) \times W_{m,j}}{COUNT(Child_j)} \times (1 - P_j)$
  Send $P_{j'}$ to $Parent_j$ with *FORWARD* message
**end if**
**if** Receive *BACKWARD* messages from Parent nodes **then**
  **if** $E_{i,j} \in Remove_{i,j}$ where $i \in Peer_j$ **then**
    Recover $Remove_{i,j}$ for $G'$ {Peer relationship lists all vertices with the same height as $j$}
  **end if**
  **if** No $Sub - MIST_j$ **then**
    Building $Sub - MIST_j$
  **end if**
  Let $C_j = MINof(P_{j'}) = MINof(\frac{\sum_{m \in Child_j, Parent_j, Peer_j} (P_{m'}) \times W_{m,j}}{COUNT(Child_j)+1} \times (1 - P_j))$ {Following root to leaf order in $Sub - MIST_j$}
  Send $C_j$ to $Child_j$ with *BACKWARD* message
**end if**

---

*C. Peer Relationship*

We introduce the peer relationship to make the DWA-Cycle to be an exact algorithm instead of an approximation algorithm for general graphs. We assume nodes have peer relationship if they stay in the same height in the MIST. After the root is determined, all its neighboring nodes should have same height as 1, and so on. Since a node only has one parent, it can be proved that if a cycle exists there should be one peer edge inside. Our method takes advantage of this result, and breaks all cycles by removing all peer edges. Another advantage is that any peer edge only affects the allocation of peer nodes instead of their parent, so the Forward step of DWA guarantees the correctness of the allocation for the *root* node. An example is shown in Figure 3, in which we have a complete graph with 3 nodes, after we select node 1 as the root node, nodes 2 and 3 have to be the children of node 1 with height 1, so they have a peer relationship. And the edge $E(2,3)$ should be removed to break the cycle $Cycle(1,2,3)$. If there exist more cycles, we can use similar approaches to build the MIST.

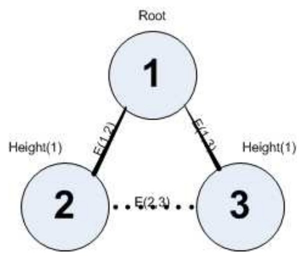


Fig. 3. Peer Relationship

### D. Example

Using the graph shown in Figure 4 with 5 nodes and 3 available channels where node 3 is the root node with all others as its children nodes, we show how Forward and Backward steps work towards spectrum allocation. The preference distributions are $[1/3,1/3,1/3]$, $[1/3,1/3,1/3]$, $[1/3,1/3,1/3]$, $[1/2,1/4,1/4]$, $[1/2,1/4,1/4]$ corresponding to nodes $1,2,3,4,5$. To simplify the calculation, we assume all edges have the same weight $W=1$.

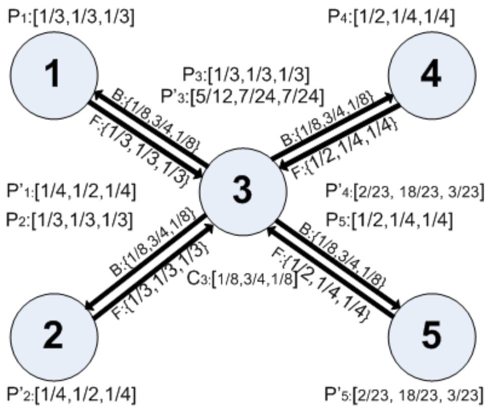


Fig. 4. DWA Example with 5 Nodes & 3 Channels

The algorithm starts with all leaf nodes $1,2,4,5$ sending forward messages with their preference distributions to the parent node 3. After collecting all forward messages, node 3 calculates its distribution $P_{3'} = \frac{\sum_{m \in Child_3}(1-P_m)\times W_{m,j}}{COUNT(Child_3)} \times (1-P_3)$ which equals to $[5/12,7/24,7/24]$ after scaling to 1. Backward step starts from node 3 selecting channel 2 ($C_3 = MINof(P_{3'})$) as its allocation. Node 3 sends the result back with a backward message to its children nodes $1,2,4,5$. Since each node only has interference relationship with its parent node 3, it can get its probability distribution as $P_{j'} = \frac{\sum_{m \in Child_j, Parent_j, Peer_j}(P_{m'})\times W_{m,j}}{COUNT(Child_j)+1} \times (1-P_j)$ which is $[1/4,1/2,1/4]$ for nodes $1,2$ and $[2/23,18/23,3/23]$ for nodes $4,5$. The channel allocations are $C1,C1,C2,C1,C1$ corresponding to nodes $1,2,3,4,5$ which make sense because nodes $4,5$ pick better channel $C1$, and node 3 avoids $C1$.

### E. Complexity Analysis

The time complexity of the DWA-Tree is between $O(2Tlog(n))$ and $O(2Tn)$ depending on the tree topology where $n$ is the number of nodes and $T$ is the message transmission time.The total number of messages transmitted is $2n$ since all nodes send one message to parents in the Forward step, and receive one from parents in the Backward step. For each node, it computes at most the summation of all neighboring nodes once in the Forward step. So the computation complexity for each node is $\frac{\lambda}{n}$ where $\lambda$ is the average cardinality of the graph. For DWA-Cycle, the time complexity is the same while the message complexity is $2(n+c)$ where $c$ is the number of removed edges, because each removed node requires 2 messages. We assume the MIST and all Sub-MISTs are computed only once, and didn't include their cost in our analysis.

## VI. Simulation Results

We have implemented our algorithms using the Matlab-BGL (Boost Graph Library) package, simulated DWA-Tree, and DWA-Cycle in different scenarios, and compared our results with the general local optimization algorithm (LO). LO always picks the least frequent channel for each pair of transmitter and receiver by sensing local channel usage. We use the average SIR 6 as our performance metric.

$$Y_{AverageSIR} = \frac{\sum_i^n \sum_{j\neq i}^n SIR_{ij}}{n} \tag{6}$$

### A. $200m \times 200m$ Square Simulation

We first simulate our algorithms in a $200m \times 200m$ square area with random deployed nodes (each node as a pair of transmitter and receiver). The channel gain is given as $G_(ij) = \kappa \times D_(ij)$ where $\kappa$ is a random generated factor, and $D_(ij)$ is the euclidean distance between node $i$ and node $j$. The preference distribution matrix of each node is also random generated in this scenario. We set up this area with 10 and 20 available channels. The result for DWA-Tree is shown in Figure 5. In the simulation, we start from 0 node and end up with 300 nodes. The interferences are generally the same with fewer neighboring nodes during the initiation. Then it shows a remarkable difference when the number of nodes is between 100 and 250. The average SIR in DWA-Tree is about 30% percent lower than LO with

250 nodes for both 10 and 20 availible channels. When the number of nodes is more than 250, the area starts getting crowded, and the interference level significantly increases consequently. It also shows a reasonable decrease when the number of availible channels increases from 10 to 20. Similar results are shown in Figure 6 for DWA-Cycle.

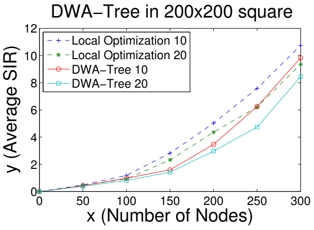


Fig. 5. DWA-Tree (200x200)

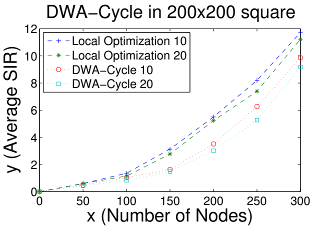


Fig. 6. DWA-Cycle (200x200)

### B. Scalability

The square area becomes too crowded when the number of nodes increases, and it ends up with an uninteresting scenario in which each node becomes a neighbor to all others. For that reason, we choose another method to test the scalability of our algorithms. We generate the adjacency matrix with random weights, so that two nodes are neighbors when the corresponding weight in adjacency matrix is positive. We began testing with 500 nodes and increased the number of nodes up to 2500 nodes. Although the results fluctuated, DWA-Tree and DWA-Cycle consistently showed the advantage comparing with LO for both 10 channels and 20 channels as shown in Figure 7 and 8.

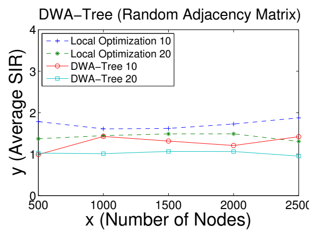


Fig. 7. DWA-Tree (RAM)

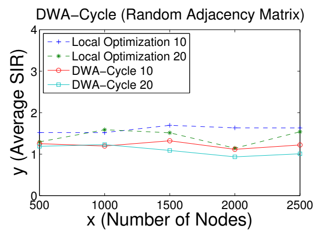


Fig. 8. DWA-Cycle (RAM)

### C. Impact of Cycles

To evaluate the impact of the number of cycles, we analyze the result with increasing number of cycles and show the results in Figure 9 and 10 with 10 and 20 channels.

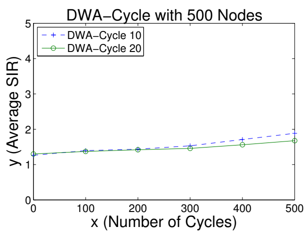


Fig. 9. DWA-Cycle (500 Nodes)

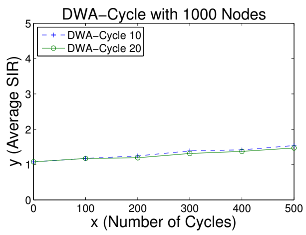


Fig. 10. DWA-Cycle (1000 Nodes)

We observe that the performance increases consistently but slightly with increasing number of cycles. The ratio goes down when the number of nodes increases (or the number of available channels increases). It is relatively consistent when the number of cycles is reasonable (100 to 300 for 500 nodes; 100 to 400 for 1000 nodes). In general, the proposed algorithms work well with reasonable number of nodes and channels, and have remarkable advantage comparing with local optimization methods.

## VII. Conclusion and Future Work

In this paper, we first provided an introduction to cognitive networks and factor graphs. Then, we described the general problem of distributed spectrum allocation, and proposed a factor graph based model. We show that with the proposed DWA algorithms, the global link quality is consistently improved in a variety of scenarios. In future, we will evaluate our algorithms in the presence of malicious nodes or noncooperative nodes.

## Acknowledgment

This work is partially supported by the National Science Foundation under Grant No. IIS-1017926 and CNS-0709285.

## References

[1] R. Thomas, L. DaSilva, and A. MacKenzie, “Cognitive networks,” in *New Frontiers in Dynamic Spectrum Access Networks, 2005. DySPAN 2005. 2005 First IEEE International Symposium on*, nov. 2005, pp. 352 –360.

[2] J. Mitola and G. Q. Maguire, “Cognitive radio: making software radios more personal,” *Personal Communications, IEEE*, vol. 6, no. 4, pp. 13–18, 1999. [Online]. Available: http://dx.doi.org/10.1109/98.788210

[3] J. M. Iii, “An integrated agent architecture for software defined radio,” 2000.

[4] A. Sahai, S. M. Mishra, R. T, and K. A. Woyach, “Cognitive radios for spectrum sharing,” *IEEE Signal Processing Mag*, pp. 140–145, 2009.

[5] C. Wang, S. Tang, X. yang Li, and C. Jiang, “Capacity scaling laws of cognitive networks,” 2009.

[6] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, “Factor graphs and the sum-product algorithm,” *IEEE Transactions on Information Theory*, vol. 47, pp. 498–519, 2001. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.54.1570

[7] U. Acar, A. Ihler, R. Mettu, and O. Sumer, “Adaptive inference on general graphical models,” in *Proceedings of the Proceedings of the Twenty-Fourth Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI-08)*. Corvallis, Oregon: AUAI Press, 2008, pp. 1–8.

[8] U. Acar, A. T. Ihler, R. R. Mettu, and Ö. Sümer, “Adaptive bayesian inference,” in *Advances in Neural Information Processing Systems 20*, J. Platt, D. Koller, Y. Singer, and S. Roweis, Eds. Cambridge, MA: MIT Press, 2008, pp. 1441–1448.

[9] M. Johansson, “Approximate bayesian inference by adaptive quantization of the hypothesis space,” in *in Proc. 25th Int. Workshop on Bayesian Inference and Maximum Entropy Methods in Science and Engineering*, 2005, pp. 7–12.

[10] P. Bahl, R. Chandra, and J. Dunagan, “Ssch: slotted seeded channel hopping for capacity improvement in ieee 802.11 ad-hoc wireless networks,” in *MobiCom ’04: Proceedings of the 10th annual international conference on Mobile computing and networking*. New York, NY, USA: ACM, 2004, pp. 216–230.

[11] P. Kyasanur, “Routing and interface assignment in multi-channel multi-interface wireless networks,” in *In IEEE WCNC*, 2005, pp. 2051–2056.

[12] N. Nie and C. Comaniciu, “Adaptive channel allocation spectrum etiquette for cognitive radio networks,” *Mob. Netw. Appl.*, vol. 11, no. 6, pp. 779–797, 2006.