Local Channel Assignments in Cognitive Radio Networks

Jie Wu,* Ying Dai,* and Yanchao Zhao†*

*Department of Computer and Information Sciences, Temple University, Philadelphia, PA 19122

†State Key Lab of Novel Software, Department of Computer Science and Technology, Nanjing Univesity, P. R. China

Email: {jiewu, ying.dai}@temple.edu, yczhao@dislab.nju.edu.cn

***Abstract*—*Cognitive radio networks* (CRNs) promise to enable the next generation of communication networks. The *channel assignment* (CA) problem is one of the most important issues in CRNs. In this paper, our goal is to design highly efficient and localized protocols for CA. In addition, we want to maximize node connectivity after CA, which is important for packet delivery. To this end, we design two basic algorithms and an advanced algorithm framework. Within this framework, we can change the edge priority in CA to meet different requirements. Simulation results show that the proposed framework is fast (two rounds of communication among nodes, regardless of network size) and outperforms an existing method.**

***Index Terms*—channel assignment, cognitive radio networks, list edge coloring, localized algorithms.**

I. Introduction

Cognitive radio networks (CRNs) are the key technology that enables next generation communication networks [1]. One of the most challenging problems in CRNs is the *channel assignment* (CA) problem. The CA problem is well-studied in traditional wireless networks. Considering assigning frequency channels, the objective is to satisfy the interference constraints and maximize the number of nodes with channels assigned. In its most general form, the CA problem is equivalent to the generalized graph-coloring problem, which is a well-known NP-hard problem [2].

The fundamental difference between CRNs and traditional wireless networks is that the available channels are dynamic and their availabilities vary over time. In CRNs, the CA problem has been studied from different perspectives. Some of them aim to maximize the spectrum utilization [3] subject to interference constraints. Some other works study the cross-layer optimization, including using power control [4], [5] and considering both network and link layers [6]. Our focus here is CA at the link layer only.

Our work differs from previous work in three aspects: First, in view of the high dynamics of channel availability, keeping connectivity would be significant in maintaining performance. Second, the network cannot afford to run time-consuming protocols to allocate channels in a dynamic environment. Last, we want to enhance the network performance by maximizing the assigned conflict-free links. To this end, we design a fast convergent localized protocol that assigns conflict-free channels to maximize connectivity in multihop CRNs. The main contributions can be summarized as follows:

- We propose two basic localized algorithms and an advanced localized algorithm to solve the CA problem. Specifically, we propose a method to partition the given network into "stars" (resemble 2-level trees) where a localized match between links and channels is feasible.
- A comparative simulation is conducted. We compare our localized algorithms to other existing methods in terms of assigned link rate, delivery rate, and coordinating rounds. Simulation results show that our advanced algorithm outperforms an existing method.

The rest of paper is organized as follows: In Section II, we introduce the related works. Then, the preliminaries are presented in Section III. We propose two basic localized algorithms in Section IV. In Section V, we give an advanced localized algorithm. Section VI discusses simulation methods and results. Finally, we conclude this paper in Section VII.

II. Related Works

Optimal conflict-free CA satisfying a global optimal objective is often NP-hard [7]. Based on a simplified interference model, this problem can be described as a vertex-coloring or edge-coloring problem. The generalized form of our problem could be reduced to a list-edge-coloring problem [8], which assigns every edge to a color from a prescribed list.

Centralized approximations in CRNs formulate the problem as a mixed integer programming problem [5]. However, centralized CA approaches suffer from the poor scalability due to the difficulty in capture consistent global information in a dynamic environment.

Many distributed approximations are proposed. Several distributed algorithms using $O(\Delta)$ (Δ is the maximum node degree) colors have been proposed in literatures [9]–[11]. In [12], Wang and Liu considered an iterative distributed solution based on the orientation of each link. An end node with a larger number of channels points to the other one with a smaller number of available channels. The CA starts with nodes that are local minimum (i.e. the minimum number of channel choices) and applies this process iteratively.

The localized solutions observe local interference patterns and access spectrum based on a set of rules [13] to maximize some system utilities [3]. Different from these works, our work aims at increasing network performance by maximizing connectivity in multihop CRNs based on only local information without using any iterative process.

TABLE I
LIST OF NOTATIONS

Notation	Meaning
G	a graph (V,E)
V	set of nodes
E/E'	set of links (without channels assigned)
uv	$uv \in E$, link connecting node u and node v
Δ	maximal node degree in G
N_u/N_{uv}	set of adjacent nodes of u (adjacent links of uv)
C/c	set of total available channels ($c \in C$)
C_u/C'_u	set of available channels (unused channels) on u
C_{uv}/A_{uv}	set of admissible (assigned) channels on uv
a_{uv}	$a_{uv} \in C_u$, channel assigned to uv by u
$ID(u)$	ID of node u
d_u/d_{uv}	effective degree of node u (link uv)
$p_{uv}(c)$	conflict probability of c on uv
$w_{uv}(c)$	channel weight of c over uv
Pr_{uv}	2-tuple link uv priority
S/s_u	set of stars (links in a star associated with node u)

III. PRELIMINARIES

A. System Model & Problem Formulation

We consider a CRN as a graph $G = (V, E)$, where $V = \{u, v, w, ...\}$ is the node set and E is the link set. $uv \in E$ if nodes u and v can communicate with each other. $N(u)$ represents the neighbor set of node u. $C_u \subseteq C$ denotes the set of available channels on u, where C is the set of total available channels in the network. C_u is the subset of C due to primary users at different locations. The notations used in this paper are listed in Table I. Two adjacent nodes can communicate only when they both tune to the same channel. A link exists when two adjacent nodes select the same channel for this link. Two links are adjacent if they share one end node. Conflicts exist if two adjacent links are assigned the same channel. We also assume that a common control channel exists for nodes to exchange information. The goal of CA is to maximize the number of links existing without conflicts.

We make the following assumptions used in the paper: (1) The communication range equals the interference range to make our algorithms and analysis more concise and clearer. Our model can be extended to more sophisticated ones, as shown in our simulation; (2) Each link is only assigned with a single channel. With multiple channels on a single link, our algorithm can be extended by including a bandwidth metric.

B. Example Topology

We use the topology of Fig. 1 to illustrate our algorithms. In Fig. 1, $C = \{1, 2, 3, 4\}$, $V = \{a, b, c, d, e, f, g\}$ and E is the edge set. (The link connections and labels will be described later.) Suppose there is a certain number of primary users in the network. In Fig. 1, the available channel set on node a is $\{1, 2\}$. This is because node a is within the interference range of primary users occupying channels 3 and 4.

IV. BASIC LOCAL ALGORITHMS FOR CA

Two basic algorithms are proposed: one is the node-based algorithm without coordinations between adjacent nodes, and another is the link-based algorithm with coordinations.

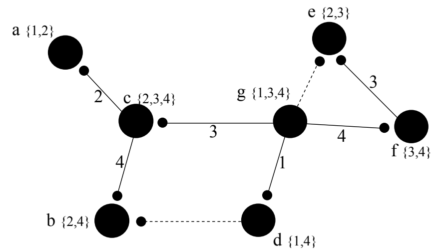


Fig. 1. The example topology.

Algorithm 1 Node-based selection

1: /* *Initial allocation phase* */
2: **for** $\forall v \in V$ **do**
3: **for** $\forall u$ such that $vu \in E'$ **do**
4: **if** $|A_{vu}| = 0$ and $|C'_v| > 0$ **then**
5: randomly pick c from C'_v
6: $A_{vu} \leftarrow c$
7: /* *Conflict resolution phase* */
8: **for** $\forall uv \in E'$ **do**
9: **if** $a_{uv} = a_{vu}$ **then**
10: $A_{uv} \leftarrow \{a_{uv}\}$, $E' \leftarrow E' - \{uv\}$
11: $C'_u \leftarrow C'_u - \{a_{uv}\}$, $C'_v \leftarrow C'_v - \{a_{uv}\}$
12: **if** $\exists mn$ s.t. $A_{mn} = 0$ and ($|C'_m| > 0$ or $|C'_n| > 0$) **then**
13: go to step 1

A. Node-based Algorithm

The node-based algorithm, which uses only local channel information at each node to select channels for adjacent links, is given in Algorithm 1. Considering the topology in Fig. 1, each node will select different channels for its own adjacent links. For example, node c would select different channels for its adjacent links. So does node g. If nodes c and g select the same channel 3 for link cg, then link cg is assigned with a matched channel. Then, both link cg and channel 3 on it will be removed from the two end nodes. The process will repeat if any node has both the unused channel and unmatched link.

The algorithm above randomly assigns channels for each link based on the information of each node. There is no coordination between two end nodes of one link. Obviously, its efficiency is low. The probability of selecting the same channel at both end nodes of a link is small, which results in many rounds needed to complete the algorithm.

B. Link-based Algorithm

Due to the low efficiency of the node-based algorithm, we now present an algorithm which has coordination between nodes and reduces the number of rounds needed.

Definition 1: *The admissible channels for link uv is defined as $C_u \cap C_v$ denoted as C_{uv}.*

To simplify our discussion, we exchange the role of nodes and links. In this case, uv corresponds to a node. uv's neighbors are either uw or vw. After this exchange, adjacent links become adjacent nodes. Then, we focus on channel

Algorithm 2 Link-based selection

1: /* *Initial allocation phase* */
2: **for** $\forall vu \in E'$ **do**
3: randomly select c from C_{vu}
4: $A_{vu} \leftarrow c$
5: /* *Conflict resolution phase* */
6: **for** $\forall uv \in E'$ **do**
7: **if** uv and any link in N_{uv} have conflicts **then**
8: remove the channel from the link with the lowest priority
9: **if** $A_{uv} > 0$ **then**
10: $E' \leftarrow E' - \{uv\}$, $C_{uv} \leftarrow C_{uv} - A_{uv}$
11: **if** $\exists mn$ s.t. $A_{mn} = 0$ and $C_{mn} > 0$ **then**
12: go to step 1

selections for nodes instead of links. So the nodes in the algorithm description below are actually the original links.

Unlike the node-based solution, the link-based solution will result in conflicts among adjacent links (new nodes). Local solutions vary depending on how (1) admissible channels are selected and (2) conflicts among adjacent nodes are resolved. These methods can be based on either a random choice or a predefined priority. The simple approach in Algorithm 2 is to have a random admissible channel selection from C_{uv} and conflict resolution based on node id: $ID(uv) = ID(u) + ID(v)$. That is, node uv with the highest $ID(uv)$ will win.

Algorithm 2 reduces the number of rounds needed by CA compared to Algorithm 1 since there is a coordination between two end nodes during channel selection. However, Algorithm 2 does not take priorities of different links into consideration, which would still result in a relatively low efficiency. In the next section, we will present an advanced algorithm which considers the priorities of links and applies maximal matching.

V. Advanced Local Algorithm for CA

Different from the previous two basic algorithms, the node-link-based algorithm makes improvements on the both sides of initial assignment and conflict resolution.

A. Basic Definitions

For the initialization, we first propose a notion of "star".

Definition 2: *A star is a special 2-level tree with one node and a set of adjacent links associated with that node.*

In each star, each link is "handled" by the end node, called a *host*. The node with a higher ID is the host. This process is called a partition based on node ID. In this way, each link is associated with one node that has a larger ID. This partition will form a forest of "stars". Then, in each "star", it is possible to perform a good initial assignment through maximal matching processing by assigning channels to links that minimizes channel conflict probability. This will maximize channel weight, which is defined in Definition 3.

Suppose a link uv selects a channel $c \in C$, the *conflict probability* with its neighbors is depicted as follows:

Algorithm 3 Node-link-based selection

1: /* *Initial allocation phase* */
2: $S \leftarrow$ partitions of G according to ID
3: **for** $\forall s_v \in S$ **do**
4: **for** $\forall vu \in s_v$, $\forall c \in C_{vu}$ **do**
5: calculate $w_{vu}(c)$
6: calculate maximal matching between channels and adjacent links by the Hungarian's algorithm
7: **for** $\forall vu \in s_v$ **do**
8: update A_{vu}
9: /* *Conflict resolution phase* */
10: **for** $\forall uv \in E'$ **do**
11: **if** uv and any link in N_{uv} have conflicts **then**
12: remove the channel from the link with a lower effective degree
13: **for** $\forall s_v \in S$ **do**
14: **for** $\forall vu \in s_v$ **do**
15: **if** $A_{vu} > 0$ **then**
16: $s_v \leftarrow s_v - \{vu\}$, $E' \leftarrow E' - \{vu\}$, $C_{vu} \leftarrow C_{vu} - A_{vu}$
17: **if** $\forall s_i$ satisfies $|C'_v| > 0$ and $A_{vu} = 0$ for $\forall vu \in s_v$ **then**
18: go to step 3

TABLE II
ADMISSIBLE CHANNEL SET ON EACH LINK

ca	cb	gc	gd	ge	gf	db	fe
2	2, 4	3, 4	1, 4	3	3, 4	4	3

TABLE III
WEIGHT OF EVERY CHANNEL ON EACH LINK

$w_{ca}(2)$	$w_{cb}(2)$	$w_{cb}(4)$	$w_{gc}(3)$	$w_{gc}(4)$	$w_{gd}(1)$
$\frac{5}{6}$	$\frac{3}{4}$	$\frac{5}{8}$	$\frac{3}{4}$	$\frac{3}{4}$	1
$w_{gd}(4)$	$w_{ge}(3)$	$w_{gf}(3)$	$w_{gf}(4)$	$w_{db}(4)$	$w_{fe}(3)$
$\frac{3}{5}$	$\frac{3}{5}$	$\frac{1}{2}$	$\frac{4}{5}$	$\frac{2}{3}$	$\frac{1}{2}$

$$p_{uv}(c) = \sum_{w \in N_v} \frac{1}{|C_{uw}|} E_{uw}(c) + \sum_{w \in N_u} \frac{1}{|C_{vw}|} E_{vw}(c) \quad (1)$$

where $E_{uw}(c)$ is a step function with a value of 1 when $c \in C_{uw}$ and 0 otherwise. Suppose $d_{uv} = d_u + d_v - 1$, where $d_u = |N_u|$. Based on this, we define the channel weight:

Definition 3: *The weight of channel c over link uv is:*

$$w_{uv}(c) = \frac{d_{uv} - p_{uv}(c)}{d_{uv}}.$$

For the conflict resolution part, we propose a local and greedy solution by considering various priorities, which is related to the importance of each node in resolving conflicts. Let d_u be the *effective node degree* of node u, defined as node degree subtracting neighbors with channels assigned. The importance of a node is then defined as its effective node degree. This strategy will establish more connections quickly, since the node with more links has a higher priority.

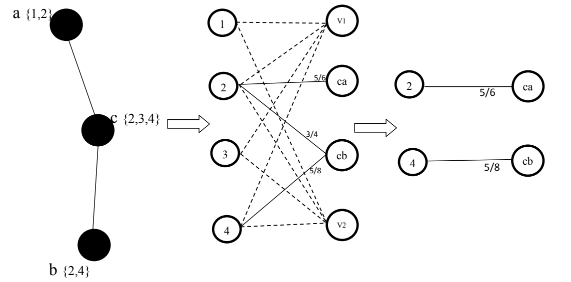


Fig. 2. The channel assignment process of the star charged by node c.

B. Node-link-based Algorithm

Combining the processes above, we can give Algorithm 3: the *node-link-based* selection algorithm. Suppose the host of link uv is u. u needs to collect C_u, C_v, and C_w for all $w \in N_v \bigcup N_u$ to calculate channel weight for uv. Therefore, two-hop information is needed (i.e., a neighbor's neighbors). This process can be done through two rounds of exchanges using a common channel. Step 1 of Algorithm 3 requires one round of exchange and is calculated only once. Step 5 needs to be re-calculated at each round as G changes.

The maximum matching is done through constructing a bipartite graph with channels at the left side and adjacent links at the right side. The weight value of each mapping edge is the corresponding channel weight on the link. We apply the Hungarian's algorithm [14] to find the maximum matching, which can be done in polynomial time, $O((|C|\Delta)^4)$, where $|C|\Delta$ is the maximal number of links in each bipartite graph. The number of channels and links can be made equal by adding virtual nodes at either side so that the number of channels and the number of links are the same. To apply perfect matching, the bipartite must satisfy *Hall's matching theorem* [14] by adding virtual edges from the virtual nodes. This would not affect the final result.

Theorem 1. The adding of virtual node and virtual edges would not affect the result of perfect matching achieved by the Hungarian's algorithm.

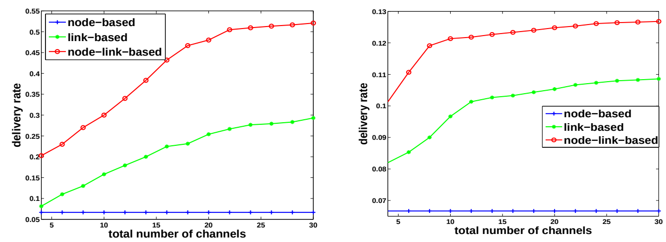
In step 9, resolving conflicts requires exchanges among host nodes, which correspond to two rounds of exchanges. There are several ways to resolve conflicts through priority. One priority is the combined effective node degree of two end nodes. Another priority is based on channel weight $w_{uv}(c)$. The higher the weight, the higher the priority. $|C_{uv}|$ can also be used as a priority with a small value corresponding to a higher priority. Steps 10 and 11 requires only local operations.

C. Examples

We now give a specific example to better illustrate our Algorithm 3. Considering the topology in Fig. 1, suppose that $ID(a) = 1, ID(b) = 2, ..., ID(g) = 7$. The admissible channel set for each link is shown in Table II. We compute the conflict probability and the weight of each channel on each link. The results of weight are shown in Table III.

TABLE IV
SIMULATION SETTINGS

total number of nodes	$[10, 40]$
communication range of each node	$[50, 70]$
total number of channels	$[4, 30]$
total number of PUs	10
interference range of PUs	$[40, 140]$



(a) simple model (b) physical interference model

Fig. 3. Comparison of delivery rate in two models.

The original graph in Fig. 1 is partitioned into three stars and links are only connected with nodes in stars. Here, we take the channel assignment on the star charged by node c for an example. We construct a bipartite graph and add two virtual nodes on the link side to conduct the perfect matching. Each edge in the bipartite graph has a weight, as computed in Table III. The weight of virtual edges connecting virtual nodes is 0. Next, we conduct the maximum matching shown in Fig. 2. The other three stars conduct their channel assignments in the same way. The final results are in Fig. 1. The number on each link is the channel assigned to it. Since link ge cannot get any channel, we use a dotted line to represent this link.

VI. SIMULATION

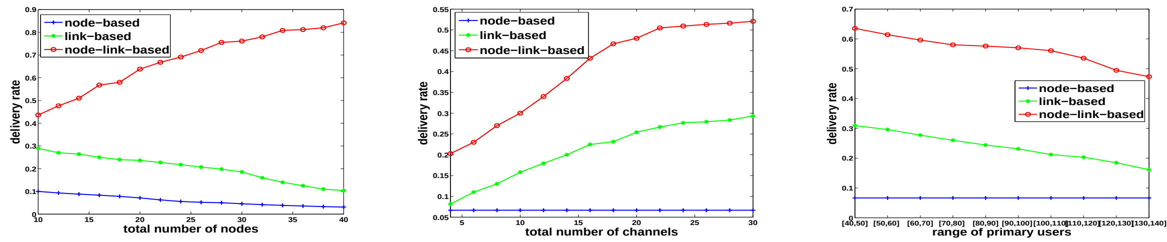
In this section, we present simulation results for our three algorithms. In addition, we implement two other algorithms: the distributed greedy algorithm in [12] and an optimal algorithm for comparison.

A. Simulation Settings & Methodology

We randomly distribute nodes in a 200×200 unit square. Also, we randomly generate a certain number of primary users (PUs). Each primary user occupies one channel. Each node in the network has its own available channel set according to the positions of PUs. The settings of parameters are shown in Table IV.

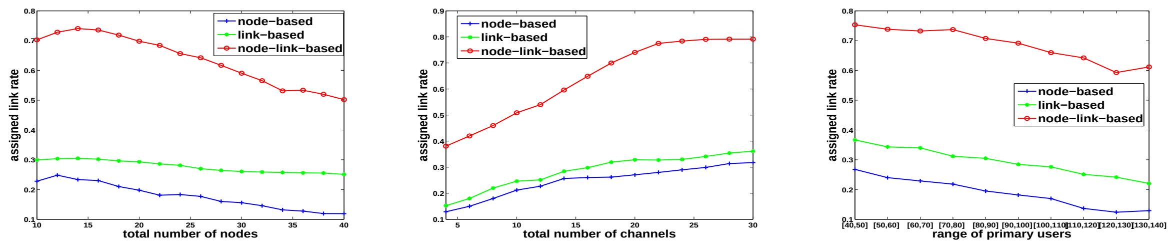
The three parameters, total number of nodes, total number of channels, and interference range of primary users, are tunable. Each time we change one of the three and compare the algorithms using three metrics: (1) assigned link rate: the ratio of assigned links over possible links; (2) delivery rate: the ratio of the maximum broadcast reachable nodes over total number of nodes; (3) number of rounds: the number of rounds needed by CA. The higher the assigned link rate, the better the result. The same applies for the delivery rate. A small number of rounds indicates higher efficiency.

Note that in our assumption (1) of Section III, the interference range equals the communication range. However, the



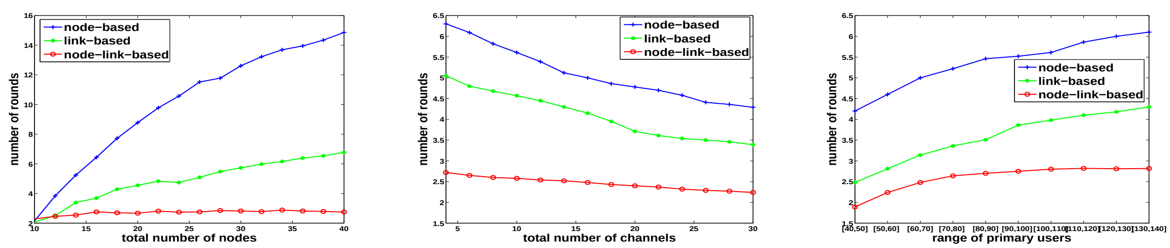
(a) change nodes (b) change channels (c) change primary users

Fig. 4. Comparison of delivery rate among three proposed algorithms.



(a) change nodes (b) change channels (c) change primary users

Fig. 5. Comparison of assigned link rate among three proposed algorithms.



(a) change nodes (b) change channels (c) change primary users

Fig. 6. Comparison of rounds among three proposed algorithms.

physical interference model (i.e. the SINR model) is generally considered as a more realistic model. [15] provided a per-node interference range calculation method, which performs very close to the physical interference model. Our design could be easily extended to apply the model in [15]. We vary the number of channels from 4 to 30 while keeping the number of nodes at 15 and interference range of PUs at $[60, 70]$. The interference range of each node is computed separately for the physical driven model. We compare the delivery rate of our three algorithms before and after applying this model. Fig. 3(a) and Fig. 3(b) only have proportional differences. Therefore, we only consider the simple model in subsequent simulations.

B. Simulation Results

1) Comparison among our algorithms: We compare the three algorithms: node-based, link-based, and node-link-based.

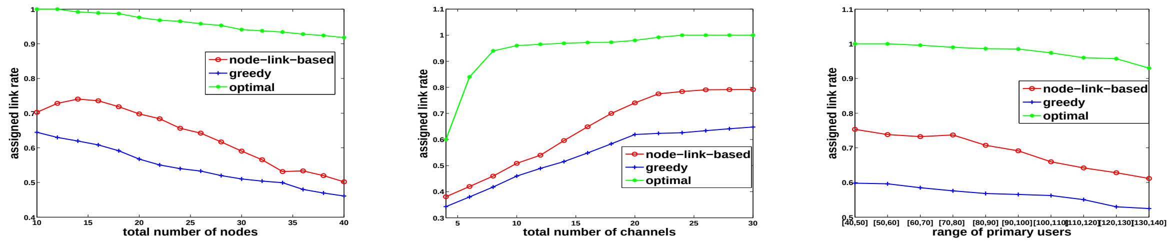
First, we compare the delivery rate by changing the three parameters. In Fig. 4(a), we vary the number of nodes from 10 and 40 while keeping the total number of channels as 10 and the interference range of primary users is randomly in $[60, 70]$. In Fig. 4(b), we vary the number of channels from 4 to 30 while keeping the number of nodes at 15 and interference range of PUs in $[60, 70]$. In Fig. 4(c), we vary the interference range of primary users while keeping the number of nodes at 15 and number of channels at 10. The results of Fig. 4 show that the node-link-based algorithm is almost 50% more than others. The trends of the three vary because more nodes and larger primary user range cause more conflicts while more channels cause fewer conflicts.

Second, we compare the assigned link rate. The settings are the same as above. The results are shown in Fig. 5. The node-link-based algorithm has almost 2.5 times the other two. Reasons of the trends are the same as the above argument.

Finally, we compare the number of rounds the three algorithms need to complete CA. With the same settings, the results are shown in Fig. 6. The node-link-based algorithm needs the least number of rounds, which is always less than three based on our simulation.

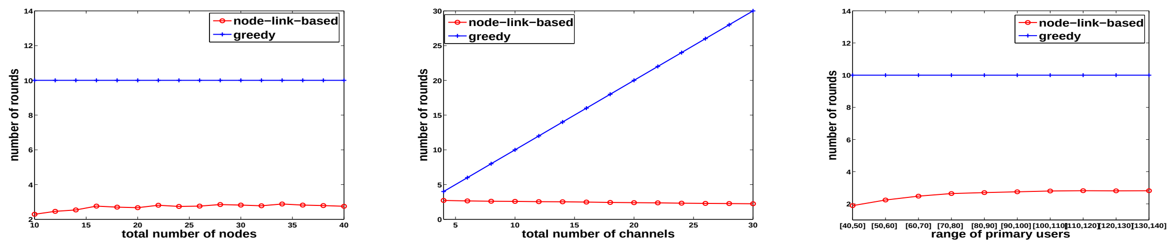
2) Comparison with alternative methods: We compare the node-link-based algorithm with alternative algorithms. One is an optimal algorithm which maximizes the assigned link rate, without consideration of the number of rounds. Another one is the distributed greedy algorithm in [12].

From above simulations, we can find that the two metrics,



(a) change nodes (b) change channels (c) change primary users

Fig. 7. Comparison of assigned link rate among node-link-based, greedy, and optimal algorithms.



(a) change nodes (b) change channels (c) change primary users

Fig. 8. Comparison of rounds among node-link-based, greedy, and optimal algorithms.

delivery rate and assigned link rate, give similar results. Here, we only compare assigned link rate. Using the same settings as before, results are shown in Fig. 7. The node-link-based algorithm achieves almost 70% of the optimal algorithm, and is 10% higher than the distributed greedy algorithm.

We compare the number of rounds needed by the node-link-based and distributed greedy algorithms (the number of rounds is too large in the optimal algorithm). We vary the number of nodes and the number of channels each time. The results in Fig. 8 show that the node-link-based algorithm takes less rounds than the distributed greedy algorithm.

C. Simulation Summary

Simulation shows that the node-link-based algorithm is almost twice in the delivery rate and assigned link rate compared to the node-based and link-based algorithms. The number of rounds needed by the node-based and link-based algorithms are on average twice more, sometimes three times more than the node-link-based algorithm. From the comparison of the node-link-based algorithm with alternative algorithms, the node-link-based algorithm reaches around 70% of the optimal algorithm and almost 10% more than the greedy algorithm in the assigned link rate. The node-link-based algorithm needs the least number - fewer than 3 - of rounds.

VII. Conclusion

In this paper, the channel assignment (CA) problem in cognitive radio networks (CRNs) is studied. We propose three algorithms: node-based, link-based, and node-link-based. In the node-link-based algorithm, we are able to achieve the best localized initialization by using a "star" structure and maximal matching. Extensive simulations are conducted to compare our algorithms from different aspects. Results show that our advanced algorithm outperforms the an existing method.

Acknowledgments

This research was supported in part by NSF grants CCF 1028167, CNS 0948184, and CCF 0830289.

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, pp. 2127–2159, 2006.

[2] Y. Yuan, P. Bahl, R. Chandra, T. Moscibroda, and Y. Wu, "Allocating dynamic time-spectrum blocks in cognitive radio networks," in *Proc. of ACM Mobihoc*, 2007.

[3] A. T. Hoang and Y.-C. Liang, "Maximizing spectrum utilization of cognitive radio networks using channel allocation and power control," in *Proc. of IEEE VTC*, 2006.

[4] A. T. Hoang and Y. C. Liang, "A two-phase channel and power allocating schemes for cognitive radio networks," in *Proc. of IEEE PIMRC*, 2006.

[5] Y. Shi, Y. T. Hou, S. Kompella, and H. D. Sherali, "Maximizing capacity in multi-hop cognitive radio networks under the SINR model," *IEEE Transactions on Mobile Computing*, vol. 99, no. PrePrints, 2010.

[6] Y. Ding and L. Xiao, "Routing and spectrum allocation for video on-demand streaming in cognitive wireless mesh networks," in *Proc. of IEEE MASS*, 2010.

[7] W. Hale, "Frequency assignment: theory and application," in *Proc. of the IEEE*, vol. 68, 1980, pp. 1497–1514.

[8] D. Marx, "NP-completeness of list coloring and precoloring extension on the edges of planar graphs," *Journal of Graph Theory*, 2004.

[9] A. Panconesi and R. Rizzi, "Some simple distributed algorithms for sparse networks," *Distributed Computing*, vol. 14(2), pp. 97–100, 2001.

[10] A. V. Goldberg, S. A. Plotkin, and G. E. Shannon, "Parallel symmetry-breaking in sparse graphs," in *SIAM J. Disc. Math*, 1987, pp. 315–324.

[11] G. De Marco and A. Pelc, "Fast distributed graph coloring with o(δ) colors," in *Proc. of ACM/SIAM SODA*, ser. SODA '01, 2001.

[12] W. Wang and X. Liu, "List-coloring based channel allocation for open-spectrum wireless networks," in *Proc. of IEEE VTC*, 2005.

[13] H. Zheng and L. Cao, "Device-centric spectrum management," in *Proc. of IEEE DySPAN*, 2005.

[14] J. A. Bondy and U. Murthy, "Graph theory with applications," *Elsevier Ltd*, 1976.

[15] L. Yang, L. Cao, and H. Zheng, "Physical interference driven dynamic spectrum management," in *Proc. of IEEE DySPAN*, 2008.