Computer Communications 36 (2013) 411–420

Contents lists available at SciVerse ScienceDirect

Computer Communications

journal homepage: www.elsevier.com/locate/comcom

Effective channel assignments in cognitive radio networks

Jie Wu [a], Ying Dai [a,*], Yanchao Zhao [a,b]

[a] Department of Computer and Information Sciences, Temple University, Philadelphia, PA 19122, United States
[b] State Key Lab of Novel Software, Department of Computer Science and Technology, Nanjing University, PR China

ARTICLE INFO

Article history:
Received 5 January 2012
Received in revised form 20 October 2012
Accepted 23 October 2012
Available online 5 November 2012

Keywords:
Channel assignment
Cognitive radio networks
Dynamic spectrum access
List edge coloring
Localized algorithms

ABSTRACT

Cognitive radio networks (CRNs) promise to be the next generation of the key enabling technology that enables *dynamic spectrum access* (DSA). The *channel assignment* (CA) problem is one of the most important issues in CRNs, with the objective of satisfying the interference constraints, and maximizing the number of nodes with channels assigned. 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.

1. Introduction

Today we are facing a dilemma of rapidly increasing demand of wideband wireless access, and shrinking out of unallocated spectrum. Studies indicate that, at any given time and location, there exists a large portion of under-utilized licensed spectrum [1]. Thus, people are exploiting new ways of transmitting on licensed bands when these bands are not fully used.

Cognitive radio networks (CRNs) are the key technology that xenables next generation communication networks [2], also known as *dynamic spectrum access* (DSA) networks. Cognitive radio technology allows the secondary users to utilize the spectrum more efficiently in an opportunistic fashion, without interfering with the primary users. As shown in Fig. 1, the secondary users $N1, N2, N3$, and $N4$, are able to make opportunistic use of channels for transmission without disturbing primary users $PU1, PU2$, and $PU3$. One of the most challenging problems is how to assign the available channels so that certain optimization objectives, such as throughput, spectrum efficiency, the number of nodes served, and fairness can be achieved. The channels here refer to the spectrum bands that have a specific central frequency. In our model, the whole spectrum is assumed to be divided into multiple channels. We will explain this setting later in Section 3.

The *channel assignment* (CA) problem is well-studied in traditional wireless networks, with the objective of satisfying the interference constraints and maximizing 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 [3]. An extensive survey of CA in wireless networks, including CRNs, can be found in Ref. [4].

The fundamental difference between CRNs and traditional wireless networks is that the available channel sets are dynamic, and their availabilities vary over time and space. In CRNs, the CA problem has been studied from different perspectives. Some of them aim to maximize the spectrum utilization [5], subject to interference constraints. Some other works study the cross-layer optimization, including using power control [6,7], and considering both network and link layers [8]. Our focus here is CA at the link layer only.

Our work differs from previous works 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, so as to maximize connectivity in multihop CRNs. The main contributions can be summarized as follows:

- We present 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" (which resemble 2-level trees) where a localized match between links and channels is feasible.

* Corresponding author. Tel.: +1 267 443 0287.
E-mail addresses: jiewu@temple.edu (J. Wu), ying.dai@temple.edu (Y. Dai), yczhao@dislab.nju.edu.cn (Y. Zhao).

0140-3664/$ - see front matter © 2012 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.comcom.2012.10.008

- Based on the proposed framework, we further develop three conflict resolution strategies to make our scheme versatilely adaptable to different network conditions.
- We also propose an extension, which can trim nonessential links to further enhance performance.
- 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 the paper is organized as follows: in Section 2, we introduce the related works. Then, the network model and problem formulation are presented in Section 3. We propose two basic localized algorithms in Section 4. In Section 5, we give an advanced localized algorithm, and present some details of implementation. In Section 6, we discuss some possible extensions of our model. Section 7 shows simulation methods and results. Finally, we conclude this paper in Section 8.

2. Related works

Optimal conflict-free CA that satisfies a global optimal objective is often NP-hard [9]. 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 [10], which assigns every edge to a color from a prescribed list.

Centralized approximations in CRNs, such as [11,7], formulate the problem as a mixed integer programming problem. However, centralized CA approaches suffer from many limitations, including the lack of a global common control channel to support the centralized control, and poor scalability due to the difficulty of capturing consistent global information in a dynamic environment.

Various distributed approximations are proposed. In the graph color model, distributed list-edge coloring algorithms usually rely on a modified version of edge-coloring algorithms. Several distributed algorithms using $O(\Delta)$ (Δ is the maximum node degree) colors have been proposed in literature [12–14]. In [15], 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 another 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 are based on observing local interference patterns and access spectrum, based on a set of rules [16] that aims at maximizing some system utilities [5]. 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.

3. Preliminaries

3.1. System model & problem formulation

We consider a CRN as a graph $G = (V, E)$, where $V = \{u, v, w, \ldots\}$ is the node set representing wireless nodes, 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, u. We assume that the whole spectrum is divided into multiple channels of different central frequencies. To simplify our model, we treat the performance (e.g., bandwidth) of each channel as the same. Our model can be extended to situations, where the bandwidth of each channel is different, by adding the channel performance as the weight to each

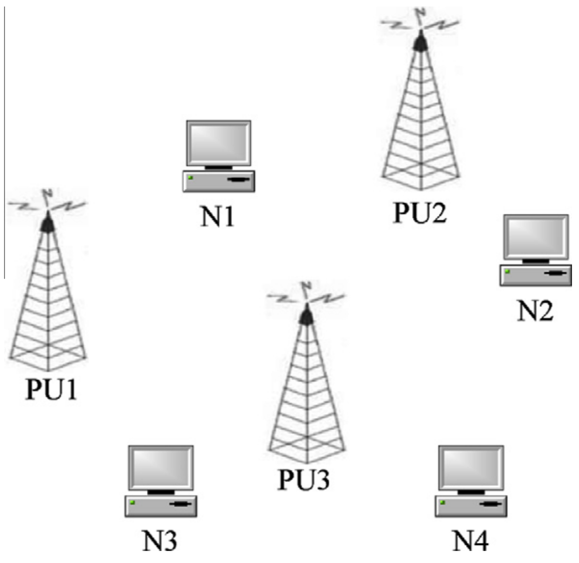


Fig. 1. An example of CRNs.

link. $C_u \subseteq C$ denotes the set of available channels on u, where C denotes the set of total available channels in the network. C_u is not equal with C, due to the different interference ranges of primary users at different locations. The notations used in this paper are listed in Table 1. We also assume that there is a common control channel for nodes to exchange information.

In wireless networks, 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. Two links are adjacent if they share one end node. Conflicts exist if two adjacent links are assigned the same channel. The goal of our paper is to perform CA so that the maximum number of links exist 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 clear and concise. Our model can be extended to more sophisticated ones, as shown in our simulation.
2. Each link is only assigned with a single channel.

3.2. Example topology

We use the topology of Fig. 2 to illustrate our algorithms. The available channel set on the whole network is $C = \{1, 2, 3, 4\}$. The

Table 1
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)

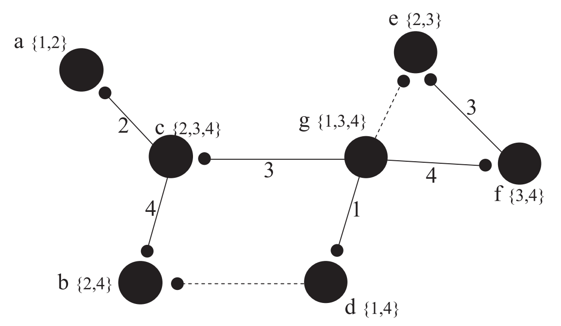


Fig. 2. The example topology.

node set is $V = \{a, b, c, d, e, f, g\}$ and the link set E is the edge set, shown in Fig. 2. (The link connections and link labels will be described later.) Suppose there is a certain number of primary users in the network, which results in different available channel sets among nodes, as shown in Fig. 2. For example, 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. The available channel set on node c is $\{2, 3, 4\}$ since it is within the interference range of primary users using channel 1.

4. Basic local algorithms for CA

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

The node-based algorithm, which uses only local channel information at each node to select channels for adjacent links, is given in Algorithm 1. It 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 being needed to complete the algorithm.

The link-based algorithm is described in Algorithm 2. We first give the definition of admissible channels, which will also be used later.

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
```

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 selections for nodes instead of links. So the nodes in the algorithm description below are actually the original links. This is the conflict graph construction process, which can also be found in Ref. [17].

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. The coordination here is achieved by two end nodes exchanging their available channel information through the common control channel. However, Algorithm 2 does not take priorities of different links into consideration; this 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.

5. Advanced local algorithm for CA

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

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
```

5.1. Basic definitions

For the initialization part, we propose a notion of "star," given in Definition 2.

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 minimize channel-conflict probability. This will maximize channel weight, which is defined in Definition 3.

Suppose a link $uv, uv \in s_u$, selects a channel $c \in C$, the *conflict probability*, with its neighbors, is depicted as the following modified $|C_{uw}|$:

$$p_{uv}(c) = \frac{T_{uv}(c)}{\sum_{uw \in s_u} \sum_{c' \in C} T_{uw} c'}, \quad (1)$$

where

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

$E_{uw}(c)$ 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 $d_u = |N_u|$.

We can easily derive that for each star s_u,

$$\sum_{uv \in s_u} \sum_{c \in C} p_{uv}(c) = 1.$$

Based on the definition of conflict probability, we define the following channel weight:

Definition 3. The channel weight of channel c over link uv is defined as

$$w_{uv}(c) = d_{uv} \times (1 - p_{uv}(c)).$$

For the conflict resolution part, we propose a local and greedy solution by considering various priorities, which are related to the importance of each node in resolving conflicts. Let d_u be the *effective node degree* of node u, defined as the total number of neighbors minus the number of 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.

5.2. Node-link-based algorithm

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: $\mathbf{E}' = \mathbf{E}$
11: for $\forall uv \in E'$ do
12:     if $uv$ and any link in $N_{uv}$ have conflicts then
13:         remove the channel from the link with a lower
            effective degree
14: for $\forall s_v \in S$ do
15:     for $\forall vu \in s_v$ do
16:         if $A_{vu} > 0$ then
17:             $s_v \leftarrow s_v - \{vu\}$
18:             $E' \leftarrow E' - \{vu\},\ C_{vu} \leftarrow C_{vu} - A_{vu}$
19: if $\forall s_v$ satisfies $|C'_v| > 0$ and $A_{vu} = 0$ for $\forall vu \in s_v$ then
20:     go to step 3
```

Combining both processes introduced 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 exchanges, and is calculated only once. Step 5 needs to be re-calculated at each round, as G changes at steps 15, 16, and 17.

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 [18] here to find the maximum matching, which can be completed in polynomial time, $O((|C|\Delta)^4)$, where $|C|\Delta$ is the maximal number of links in each bipartite graph. Moreover, the Hungarian's algorithm is a local greedy algorithm, since it only needs the weight information on each adjacent link in one star. The number of channels and the number of 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* [18] 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.*

Proof. Given a bipartite graph $G = (V = (L, R), E)$, such that for each $(u, v) \in E$, $u \in L$ and $v \in R$. Before adding virtual nodes and virtual edges, suppose the maximal perfect match achieved by the Hungarian's algorithm is $M = (V, E')$, $|L| = |R|$ and $E' \subset E$. In addition, for each $u \in L(v \in R)$, there is exactly one $v \in R(u \in L)$, such that $(u, v) \in E'$. The maximal perfect match after adding virtual nodes and virtual edges is $M' = (V', E'')$. Let $M_0 \subset M'$, such that $M_0 = (V_0, E_0)$ is composed of virtual edges whose weight are 0, where $V_0 \subset V'$ are virtual nodes and $E_0 \subset E''$ are virtual edges. First, assume the weight of all edges in M, $W(M)$, is less than the weight of all edges in $M', W(M')$. We exclude all the virtual edges from E'' and obtain $W(M' - M_0)$. $W(M' - M_0) = W(M') - W(M_0) = W(M')$ because $W(M_0) = 0$. This means that $W(M) < W(M' - M_0)$ with $M' - M_0 \subseteq G$. That is, $M' - M_0$ can also be the perfect matching of G, which contradicts that M is the maximal matching, since $M = (V, E')$ is the output of the Hungarian's algorithm.

On another hand, assume that $W(M) > W(M')$. Then, $M = (V, E')$ can also be a matching in the bipartite graph with virtual nodes and virtual edges. This contradicts the fact that $M' = (V', E'')$ is the maximal matching of the bipartite graph with virtual nodes and virtual edges, which should be the maximal. Since virtual edges do not influence the final result, virtual nodes do not affect either, because edges coming from virtual nodes are virtual edges with weight 0. □

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, which we will discuss in details later. 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. Removing assigned links in steps 10 and 11 requires only local operations in hosts.

Table 2
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 3
Conflict probability of every channel on each link.

$p_{ca}(2)$	$p_{cb}(2)$	$p_{cb}(4)$	$p_{gc}(3)$	$p_{gc}(4)$	$p_{gd}(1)$
$\frac{4}{13}$	$\frac{4}{13}$	$\frac{5}{13}$	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{3}{16}$
$p_{gd}(4)$	$p_{ge}(3)$	$p_{gf}(3)$	$p_{gf}(4)$	$p_{db}(4)$	$p_{fe}(3)$
$\frac{1}{16}$	$\frac{1}{8}$	$\frac{3}{16}$	$\frac{3}{16}$	1	0

Table 4
Weight of every channel on each link.

$w_{ca}(2)$	$w_{cb}(2)$	$w_{cb}(4)$	$w_{gc}(3)$	$w_{gc}(4)$	$w_{gd}(1)$
$\frac{27}{13}$	$\frac{36}{13}$	$\frac{32}{13}$	$\frac{21}{4}$	$\frac{21}{4}$	$\frac{65}{16}$
$w_{gd}(4)$	$w_{ge}(3)$	$w_{gf}(3)$	$w_{gf}(4)$	$w_{db}(4)$	$w_{fe}(3)$
$\frac{75}{16}$	$\frac{35}{8}$	$\frac{65}{16}$	$\frac{65}{16}$	0	3

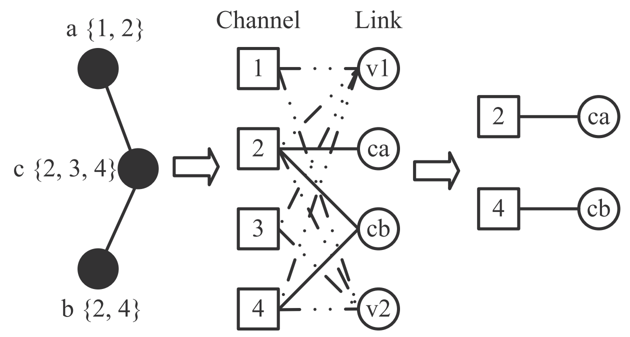


Fig. 3. The channel assignment process of the star charged by node c.

5.3. Examples

We now give a specific example based on Algorithm 3 to better illustrate our algorithm. Considering the topology in Fig. 2, suppose that $ID(a) = 1, ID(b) = 2, \ldots, ID(g) = 7$. The admissible channel set for each link is shown in Table 2. The original graph is partitioned into three stars, as shown in Fig. 2, in which links are only connected with nodes in stars. The three stars are: c with its attached links, g with its attached links, and f with its attached links. We compute the conflict probability of every available channel on each link, which is in Table 3. Next, we can get the weight of each channel on each link. The results are in Table 4.

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 4. The weight of virtual edges connecting virtual nodes is 0. Next, we conduct the maximum matching shown in Fig. 3. The other three stars conduct their channel assignments in the same way. The total view of channel assignment results is in Fig. 2. 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.

There is a conflict between links cb and db, because they both are assigned channel 4. We resolve this conflict based on the effective node degree of two end nodes, which turns out to be the same. Therefore, we remove one randomly. During the next round of our algorithm, the result is not changed. Thus, the channel assignment process is completed.

5.4. Other conflict resolutions

In the above discussion, we use the effective degree for conflict resolution. There are several other ways to resolve conflicts through priority. In this section, we propose another two conflict resolutions.

The first alternative is based on the remaining number of channels. For example, $|C_{vw}|$, which is the number of unused channels on vw, can also be used as a priority, with a smaller value corresponding to a higher priority. This strategy assigns a higher priority to links with fewer choices of channels.

The second alternative is based on whether the link is *essential* or *nonessential*. First, we assume that the priority of each node i is $ID(i)$, based on alphabetical order, such as $ID(a) < ID(b)$. Here, for simplification, the priority is decided by nodes' IDs. In real life, other meaningful metrics, for example, bandwidth and capacity, can be applied to decide the priority. Then, we introduce the priority assignment method for each link [19].

Link priority assignment: For each link vw, its priority is defined as $Pr_{uv} = (ID(u), ID(v))$.

Thus, the priority of a link is a 2-tuple, which is based on the lexicographic order. To decide the priority, first compare the first element in the 2-tuple, and then compare the second element. The first element of the 2-tuple is the priority of the start node, and the second element is the priority of the end node. Therefore, each link has a total order in the network. Next, we give the definition of a nonessential link, based on the link priority.

Definition 4. A link uv is a nonessential link if it satisfies the following conditions: (1) There is a "replacement" path P from u to v, which does not pass link uv, and (2) all the intermediate links on P have higher priorities than link uv.

Any link that does not satisfy the above condition is essential. For example, suppose that there is a path from u to v, which is $u \to w \to v$. Then, link uv is nonessential, because the intermediate links uw and wv both have higher priorities than link uv.

Therefore, when a conflict happens among two adjacent links, we can first check if they are essential. The nonessential ones will be removed directly. If the adjacent links are both essential, we will use other methods – effective node degree, or the number of channels remained on the link – to decide which one should be removed.

6. Discussion

In this section, we will discuss additional issues regarding our algorithms.

6.1. Benefits of edge-trimming

We first discuss the possibility of edge-trimming. Edge-trimming is the process of cutting off edges from a network before conducting the channel assignment algorithms. The intuition is that, since the resolution of the conflicts among edges will leave low-priority edges with less chance to get assigned, we can just cut some low-priority edges out prior to the channel assignment stage, according to the resource condition. In this way, the assignment algorithm will converge faster without affecting the network performance. The edge-trimming strategy must follow these design objectives:

- Cut as many non-essential edges as possible, so that the algorithms will converge faster.
- Make sure the performance of networks does not degrade after channel assignment.

To achieve a better tradeoff between these two objectives, the most crucial part is how to evaluate the importance of edges. Our method is marking the edges as essential or non-essential, according to Definition 4.

We can develop more strategies to perform edge-trimming, like the conflict possibility and effective degrees. Due to space limitations, we will not extend these topics here.

6.2. Dealing with dense networks

In some cases, we may face situations of dense networks where the number of neighbors of each node is much larger than normal ones.

From Fig. 6(a), we can see that our algorithm's performance degrades drastically when the networks' density increases. This is mainly due to the growth of the number of conflicts. To mitigate this situation, we can take advantage of the adjustable transmission range to form a cluster backbone using methods in Refs. [20,21].

In essence, we can construct a backbone of the whole network, so that the assignment will perform in a small scale. In the backbone consisting of cluster heads with longer transmitting range, our proposed algorithms, can be implemented to assign channels. Meanwhile, the inner cluster communication can be conducted by using CSMA. To make sure that there is no inter-cluster interference, cluster heads should perform other rounds of vertex coloring algorithms to assign inner-cluster communicating channels.

6.3. Dealing with highly dynamic spectrum availability

We are also confronting a situation where admissible channels on each node are changing all the time, due to the dynamic access and appearance of a primary user. In this situation, we give a local adjustment method in the following. This can be used as an extension to the proposed assignment algorithm, which could quickly reassign channels in response to the change in admissible channels.

A possible method is that the nodes that changed their assignments actively coordinate with neighbors, among which the links are affected. During the coordination, every node collects residual and acceptable channels on each link, and performs a maximal matching. The residual channels refer to the possible and conflict-free channels to the other links, emitted from current nodes. The acceptable channels refer to those channels which are conflict-free on both end nodes. In this way, the adjustment operation will not affect the current allocated links. Obviously, the adjustment methods will maintain the topology of the whole network as much as possible. However, the long-term performance will not be guaranteed.

7. Simulation

In this section, we present simulation results for our three algorithms. In addition, we implement two other algorithms: the distributed greedy algorithm in Ref. [15] and an optimal algorithm for comparison, which are compared with our node-link-based algorithm. The greedy algorithm iteratively assigns channels to the node with minimum channel choices. Our algorithm does not need the iterative process. We calculate the maximal matching among links and channels, and then resolve the conflicts. Also, we compare the three conflict resolutions of our third algorithm. Moreover, we implement the approach proposed in our discussion part, and make some comparisons.

Table 5
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	$[40, 140]$

7.1. Simulation settings & methodology

We randomly distribute nodes in a 200×200 unit square. Also, we randomly generate a certain number of nodes and primary users (PUs) with different interference ranges. Each primary user occupies one channel in a certain range. If a node is within a PU's range, it cannot use the channel occupied by the PU. Therefore, 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 5. Under a certain setting, we run our simulation 100 times to achieve stable results.

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 parameters to compare the algorithms using the following metrics:

- *assigned link rate*: the ratio of assigned links over possible links,
- *delivery rate*: the ratio of the maximum broadcast reachable nodes over total number of nodes, and
- *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. Based on the above system settings and evaluation metrics, the simulation results are presented in the next section.

Note that in our assumption (1) in Section III, the interference range equals the communication range. Because of the accumulative effect of interference, the physical interference model (i.e. the SINR model) is generally considered as a more realistic model. Yang et al. [22] provided a per-node interference range calculation method, which performs very closely to the physical interference model.

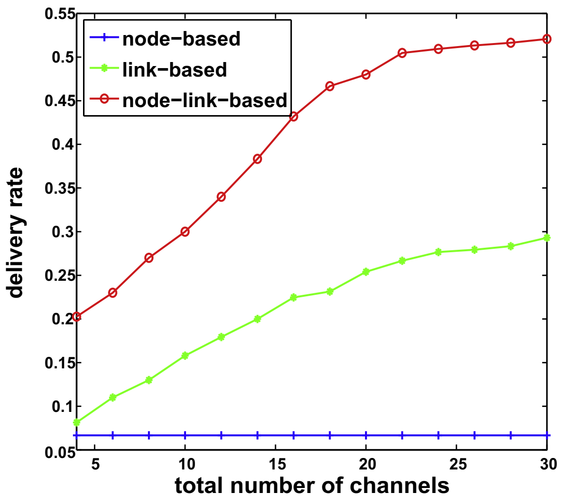
Our design could be easily extended to apply the model in Ref. [22]. We vary the number of channels from 4 to 30, while keeping the number of nodes at 15 and interference range of PUs at $[60, 70]$. The interference range of each node is computed separately for the physical driven model. We compare the delivery rate of our three algorithms before and after applying this model. Fig. 4(a) and (b) have similar comparison results. In both figures, the performance of the node-link-based algorithm is the best, and the node-based algorithm is the worst. In addition, we implement two other algorithms. One is an optimal algorithm which maximizes the assigned link rate, without consideration of the number of rounds it takes. Another one is the distributed greedy algorithm in Ref. [15]. We compare the assigned link rate of the two algorithms and our node-link-based algorithm. Comparison results in Fig. 4(c) and (d) show that the comparison results are the same, with similar trends for each algorithm. Comparisons of other metrics are also similar. Therefore, in subsequent simulations, we only consider the simple model.

7.2. Simulation results

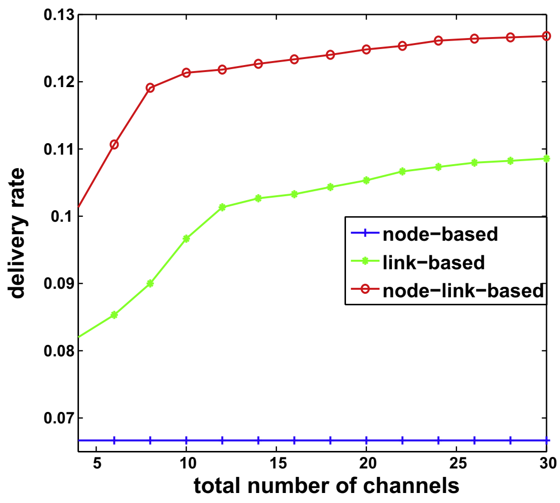
7.2.1. Comparison among our algorithms

We compare the three algorithms, node-based, link-based, and node-link-based, in terms of the three metrics discussed in the previous subsection.

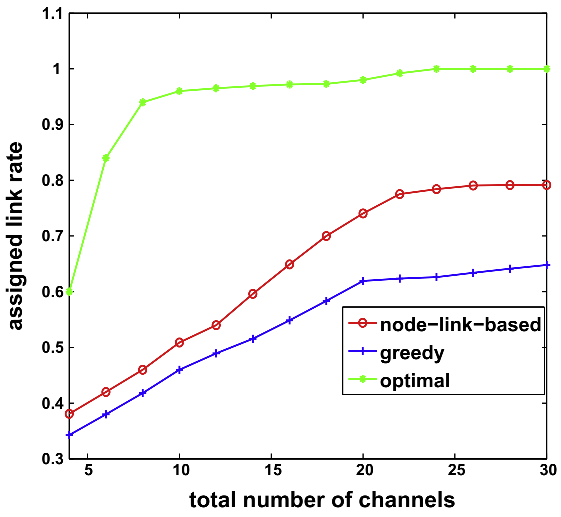
First, we compare the delivery rate by changing the three parameters. In Fig. 5(a), we vary the number of nodes from 10 and 40, while keeping the total number of channels as 10. The



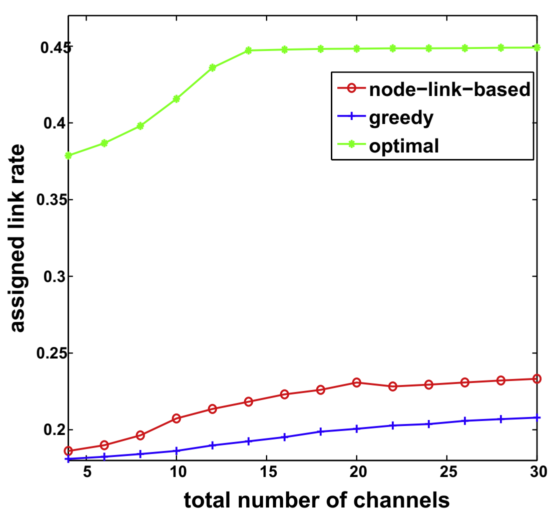
(a) simple model



(b) physical interference model

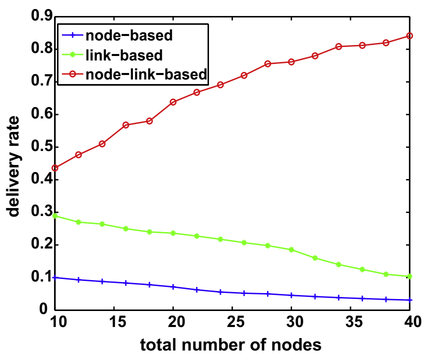


(c) simple model

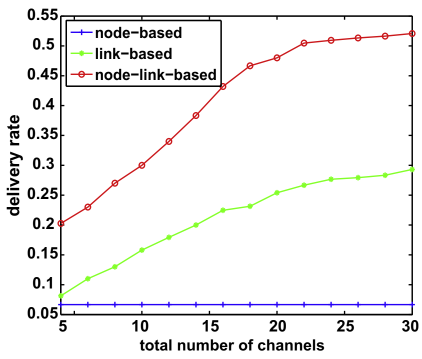


(d) physical interference model

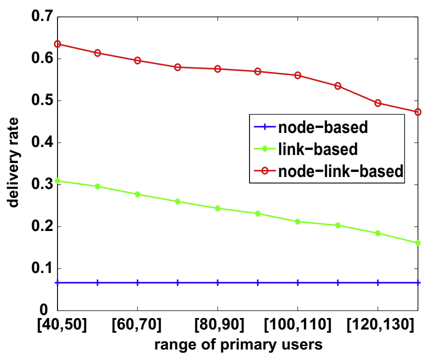
Fig. 4. Comparison of delivery rate in two models.



(a) change nodes



(b) change channels



(c) change primary users

Fig. 5. Comparison of delivery rate among three proposed algorithms.

interference range of primary users is randomly generated among [60, 70]. In Fig. 5(b), 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]. In Fig. 5(c), we vary the interference range of primary users, while keeping the number of nodes at 15, and the number of channels at 10. Based on the settings of the three parameters, we randomly generate a topology each time. The results of Fig. 5 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 ranges cause more conflicts, while more channels cause fewer conflicts. The trend of the node-link-based algorithm in Fig. 5(a) is increasing because, even though more conflicts cause fewer assigned links, the assigned links are more connected; this causes the increase in the delivery rate.

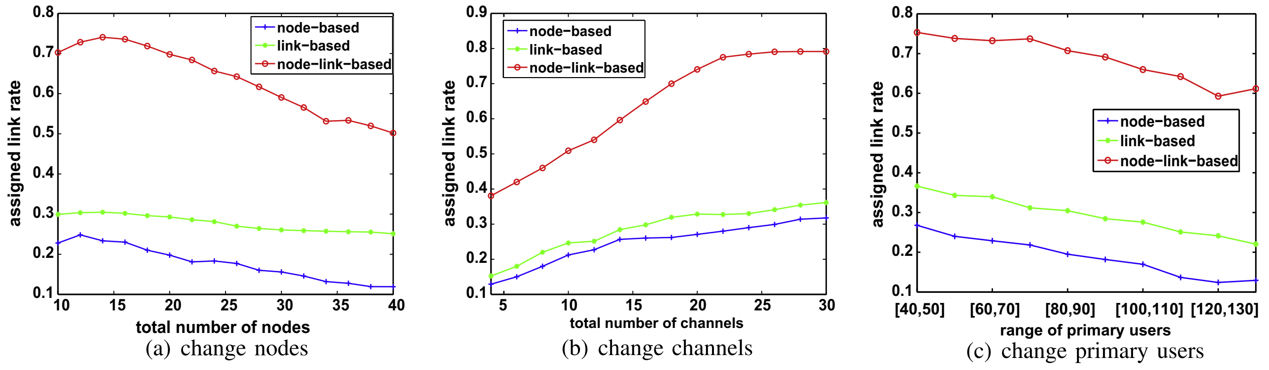


Fig. 6. Comparison of assigned link rate among three proposed algorithms.

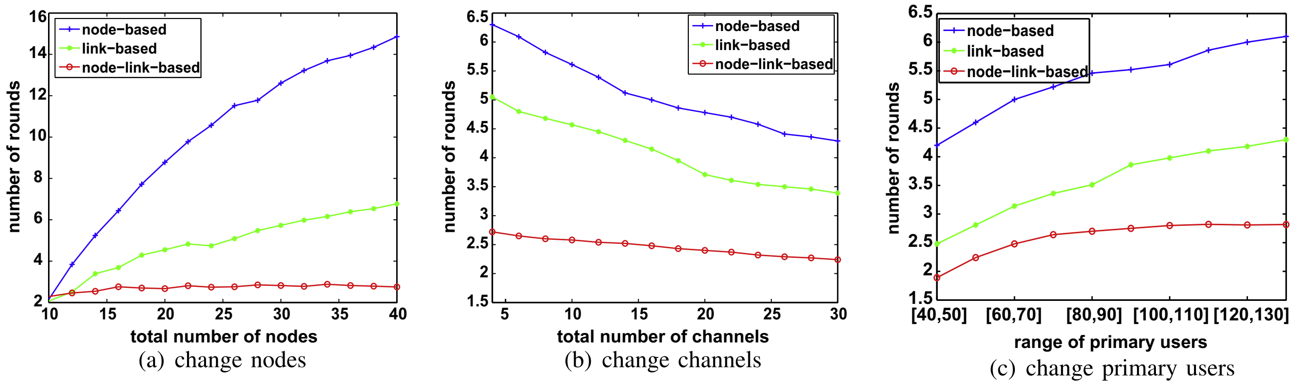


Fig. 7. Comparison of rounds among three proposed algorithms.

Second, we compare the assigned link rate. The settings are the same as the ones comparing the delivery rate. The results are shown in Fig. 6. The node-link-based algorithm has almost 2.5 times the other two in the assigned link rate. Reasons of the trends are the same as the above argument.

Finally, we compare the number of rounds the three algorithms need to complete channel assignment. The settings are the same, with results shown in Fig. 7. The node-link-based algorithm needs the least number of rounds to complete channel assignment, which is always less than three, based on our simulation.

7.2.2. Comparison with alternative methods

For better comparison, we compare the node-link-based algorithm with the distributed greedy algorithm and optimal algorithm. We present the comparison results in the following:

From the above comparison results, we can find that the two metrics, delivery rate, and assigned link rate give similar results. Here, we only compare delivery rate and number of rounds.

First, we compare the assigned link rate of the three algorithms. Using the same settings as before, results are shown in Fig. 8. We can tell that the node-link-based algorithm achieves almost 70% of the optimal algorithm, and is 10% higher than the distributed greedy algorithm. This is because the optimal algorithm performs CA without consideration of resources.

To better evaluate, 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. 9 show that the node-link-based algorithm takes less rounds than the distributed greedy algorithm. The number of

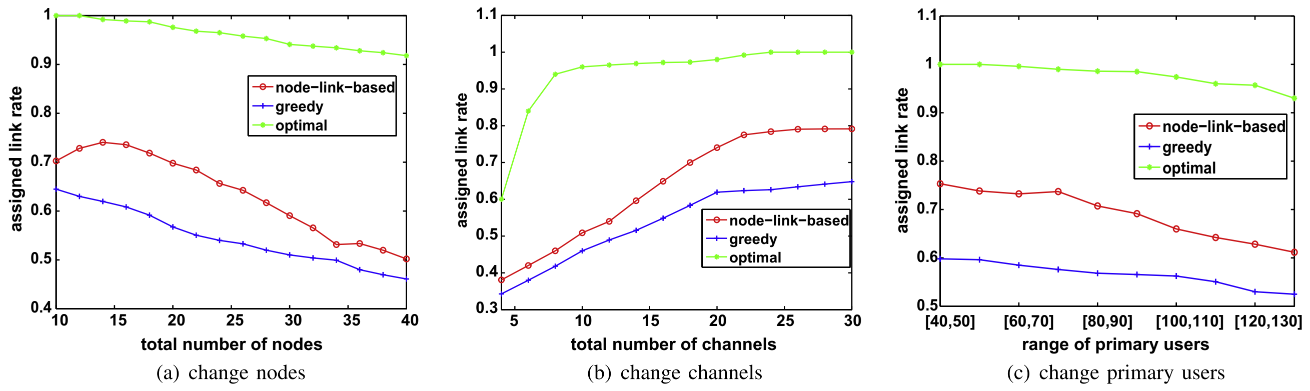


Fig. 8. Comparison of assigned link rate among node-link-based, greedy, and optimal algorithms.

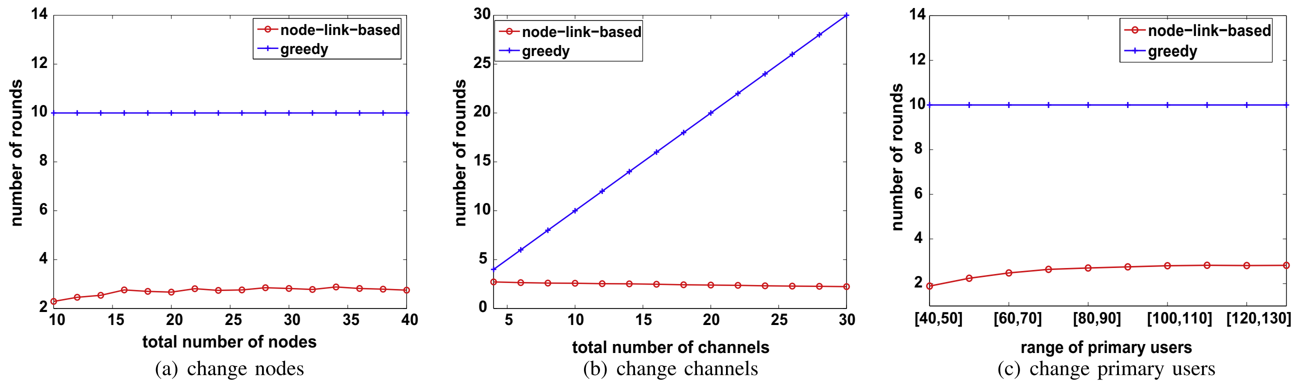


Fig. 9. Comparison of rounds among node-link-based, greedy, and optimal algorithms.

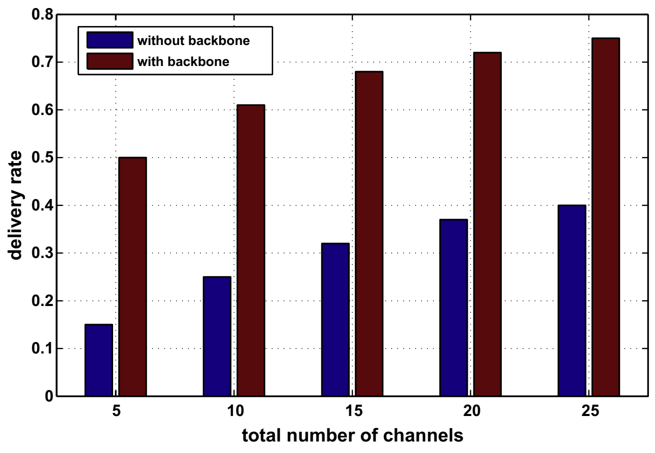


Fig. 10. Comparison of delivery rate with and without backbones.

rounds in the greedy algorithm equals the number of channels. This is because the greedy algorithm needs to run once for each available channel.

7.2.3. Comparison of conflict resolutions

In this subsection, we implement the three resolution strategies: effective degree, essential edges, and the number of residual channels. We use the three conflict resolutions to implement the node-link-based algorithm. Under the same settings as before, the results are shown in Fig. 11(a). We can see that resolving conflict through essential edges uses a smaller number of rounds than the other two strategies. Also, we compare the delivery rate of the three conflict resolution strategies by changing the number of nodes. The other settings remain the same. The results in Fig. 11(b) show that conflict resolution based on essential edges has the highest delivery rate. This is because the replacement paths of nonessential edges do not reduce network connectivity.

7.2.4. Comparison of with and without backbones

We increase the density of the network through increasing the number of nodes to 70. We compare the delivery rate of our node-link-based algorithm with backbones and without backbones, varying the total number of available channels. The other settings are the same as the comparison of delivery rate, as in the above. The comparison results are shown in Fig. 10. From the results, we can tell that when the network density increases, the delivery rate of our algorithm with backbones can achieve almost twice

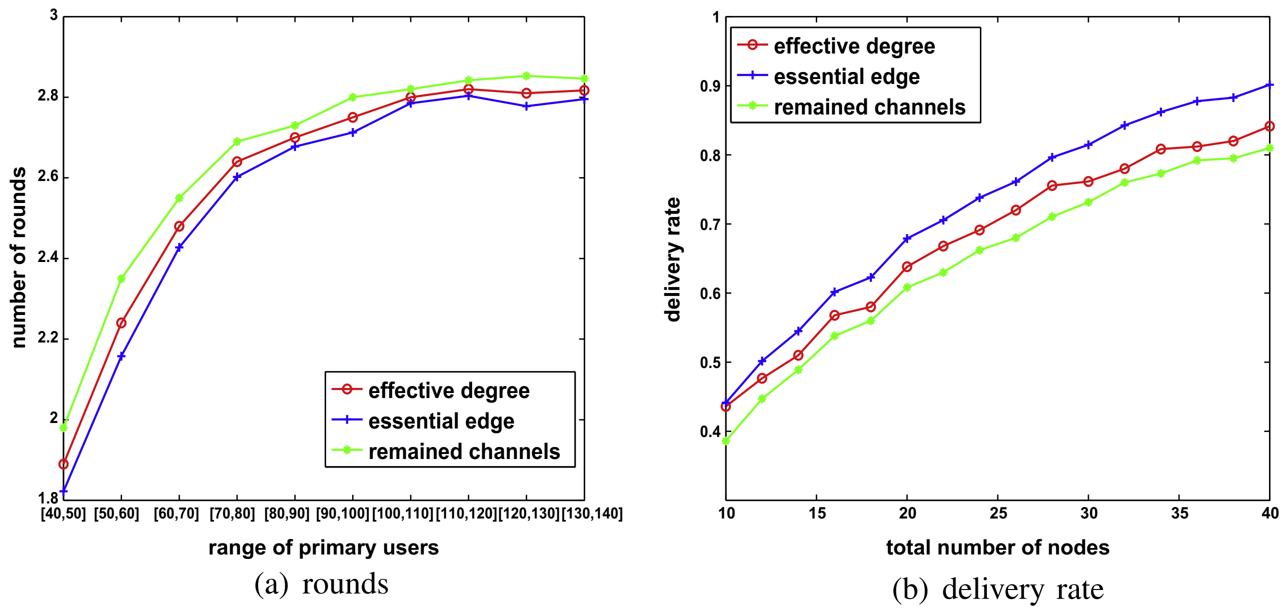


Fig. 11. Comparison among three conflict resolution strategies.

the delivery rate compared to the same algorithm without backbones.

7.3. Simulation summary

Simulation results conclude that the node-link-based algorithm has almost twice the delivery rate and assigned link rate, compared to the node-based algorithm and the link-based algorithm. 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. In addition, from the comparison of the node-link-based algorithm, distributed greedy algorithm, and optimal algorithm, 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. We can also conclude that the node-link-based algorithm needs the least number – fewer than 3 – of rounds to complete channel assignment. From the comparison of the three different conflict resolution strategies of the node-link-based algorithm, resolving conflict based on essential edges needs the least number of rounds, while achieving the highest delivery rate of the three. The comparison of delivery rate with and without backbones shows that the node-link-based algorithm with backbones in networks with high densities can achieve almost twice on the delivery rate, as compared to the same algorithm without backbones.

8. 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. We also present three ways of conflict resolution. We introduce the notion of essential edges, and propose the possible extensions of our algorithm, with the goals of improving the efficiency while keeping the connectivity of the network. Extensive simulations are conducted to compare our algorithms with others from different aspects. Results show that our advanced algorithm outperforms existing methods.

Acknowledgments

This research was supported in part by NSF Grants ECCS 1128209, CNS 1065444, CCF 1028167, CNS 0948184, and CCF 0830289.

References

[1] FCC, Notice of proposed rule making and order, ET Docket No. 03-222 2003.
[2] I.F. Akyildiz, W.-Y. Lee, M.C. Vuran, S. Mohanty, Next generation/dynamic spectrum access/cognitive radio wireless networks: a survey, Computer Networks 50 (2006) 2127–2159.
[3] Y. Yuan, P. Bahl, R. Chandra, T. Moscibroda, Y. Wu, Allocating dynamic time-spectrum blocks in cognitive radio networks, in: Proc. of ACM Mobihoc, 2007.
[4] G.K. Audhya, K. Sinha, S.C. Ghosh, B.P. Sinha, A survey on the channel assignment problem in wireless networks, Wireless Communications and Mobile Computing (2010).
[5] A.T. Hoang, Y.-C. Liang, Maximizing spectrum utilization of cognitive radio networks using channel allocation and power control, in: Proc. of IEEE VTC, 2006.
[6] A.T. Hoang, Y.C. Liang, A two-phase channel and power allocating schemes for cognitive radio networks, in: Proc. of IEEE PIMRC, 2006.
[7] Y. Shi, Y.T. Hou, S. Kompella, 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.
[8] Y. Ding, L. Xiao, Routing and spectrum allocation for video on-demand streaming in cognitive wireless mesh networks, in: Proc. of IEEE MASS, 2010.
[9] W. Hale, Frequency assignment: theory and application, in: Proc. of the IEEE, vol. 68, 1980, pp. 1497–1514.
[10] D. Marx, NP-completeness of list coloring and precoloring extension on the edges of planar graphs, Journal of Graph Theory 49 (2004) 313–324.
[11] T. Shu, M. Krunz, Joint power/rate control and channel assignment in cognitive radio networks: a multi-level spectrum opportunity perspective, University of Arizona, Department of ECE, Tech. Rep., 2008.
[12] A. Panconesi, R. Rizzi, Some simple distributed algorithms for sparse networks, Distributed Computing 14 (2) (2001) 97–100.
[13] A.V. Goldberg, S.A. Plotkin, G.E. Shannon, Parallel symmetry-breaking in sparse graphs, SIAM Journal on Discrete Mathematics (1987) 315–324.
[14] G. De Marco, A. Pelc, Fast distributed graph coloring with $o(\delta)$ colors, in: Proc. of ACM/SIAM SODA Ser. SODA '01, 2001, pp. 630–635.
[15] W. Wang, X. Liu, List-coloring based channel allocation for open-spectrum wireless networks, in: Proc. of IEEE VTC, 2005.
[16] H. Zheng, L. Cao, Device-centric spectrum management, in: Proc. of IEEE DySPAN, 2005.
[17] E. Tragos, A. Fragkiadakis, I. Askoxylakis, V. Siris, The impact of interference on the performance of a multi-path metropolitan wireless mesh network, in: Proc. of the 16th IEEE Symposium on Computers and Communications (ISCC' 11), 2011.
[18] J.A. Bondy, U. Murthy, Graph Theory with Applications, Elsevier Ltd., 1976.
[19] S. Yang, J. Wu, F. Dai, Efficient directional network backbone construction in mobile ad hoc networks, IEEE Transactions on Parallel and Distributed Systems 19 (2) (2008).
[20] J. Wu, F. Dai, Virtual backbone construction in manets using adjustable transmission ranges, IEEE Transactions on Mobile Computing 5 (2006) 1188–1200.
[21] J. Wu, H. Li, On calculating connected dominating set for efficient routing in ad hoc wireless networks, in: Proc. of ACM Dial-M, ACM, 1999, pp. 7–14.
[22] L. Yang, L. Cao, H. Zheng, Physical interference driven dynamic spectrum management, in: Proc. of IEEE DySPAN, 2008.