Chapter 3

Exploring Opportunistic Spectrum Availability in Wireless Communication Networks

Wei Wang and Xin Liu and Hong Xiao

Contents

3.1 Introduction ...........................................................62
3.1.1 Related Work...................................................63
3.1.2 Contributions...................................................64
3.1.3 Organization ...................................................64
3.2 A Framework for Opportunistic Spectrum Sharing .............65
3.2.1 Effective Nonopportunistic Bandwidth .......................67
3.3 Problem Formulation ..................................................68
3.3.1 Color Decoupling ..............................................70
3.4 Proposed Algorithms ..................................................71
3.4.1 Optimal Solutions: Benchmark .................................71
3.4.2 Distributed Greedy Algorithm .................................72
3.4.3 Distributed Fair Algorithm .....................................73
3.4.4 Randomized Distributed Algorithm ...........................74
3.5 Numerical Results .....................................................75
3.5.1 Simulation Setup ...............................................76
3.5.2 Performance under the Snapshot of the Network ............78
3.5.3 Performance under Time-Varying Channel Availability .....82
3.1 Introduction

The radio spectrum is among the most heavily regulated and expensive natural resources around the world. In Europe, the 3G spectrum auction yielded 35 billion dollars in England and 46 billion in Germany. The question is whether spectrum is really this scarce. Although almost all spectrum suitable for wireless communications has been allocated, preliminary studies and general observations indicate that much of the radio spectrum is not in use for a significant amount of time, and at large numbers of locations. For instance, experiments conducted by Shared Spectrum Company indicate 62% percent of “white space” (unused space) below the 3GHz band even in the most crowded area near downtown Washington, DC, where both governmental and commercial spectrum usage are intensive.\(^1\) In the experiment, a band is counted as white space if it is wider than 1MHz and remains unoccupied for 10 minutes or longer. Furthermore, spectrum usage levels vary dramatically in time, geographic location, and frequency. A lot of precious spectrum (below 5GHz), perfect for wireless communications that are worth billions of dollars, sits there silently. The large proportion of white space indicates that opportunistic or dynamic spectrum usage may significantly mitigate the spectrum scarcity.

In this paper, we focus on the opportunistic exploration of the white space by users other than the primary licensed ones on a noninterfering or leasing basis. Such usage is being enabled by regulatory policy initiatives and radio technology advances. First, both the Federal Communications Commission (FCC) and the federal government have made important initiatives towards more flexible and dynamic spectrum usage, e.g., Refs. 2–6. Furthermore, opportunistic spectrum sharing is enabled by software-defined-radio or cognitive-radio technologies, where these technology advances provide the capability for a radio device to sense and operate on a wide range of frequencies using appropriate communication mechanisms, and thus enable dynamic and more intense spectrum reuse in space, time, and frequency dimensions.

We focus on the study of the secondary users who observe the channel availability dynamically and explore it opportunistically. Here, secondary users refers to spectrum users who are not owners of the spectrum and operate on the basis of agreements/etiquette imposed by the primary users/owners of the spectrum. We study the impact of the opportunistic spectrum availability on the secondary users who explore the spectrum when allowed by the primary users of the spectrum. (Note that the secondary users may have their own licensed/allocated bandwidth where they
are primary users, which is not the concern of this chapter.) Because of the traffic load and the distribution of the primary users, the available channels observed by the secondary users are time-varying and location dependent. We study the impact of the characteristics of primary users on the spectrum's opportunistic availability. We present a general framework to model the correlation between primary and secondary users and introduce a new metric to capture the impact of potential opportunistic spectrum sharing. We also propose several distributed spectrum access algorithms and study their performance under the above-mentioned time-varying and location-dependent channel availabilities.

3.1.1 Related Work

Channel allocation schemes have been studied in the literature for over three decades. Various fixed, dynamic, and hybrid channel allocation schemes have been proposed. In Ref. 7, the authors present a comprehensive survey of the channel assignment strategies. In addition, channel allocation has been modeled as a graph coloring, list coloring, or maximum packing problem (e.g., Refs. 7–14 and references therein).

The main differences between our work and the traditional approaches are multifold: (1) the allocation requirement is different, (2) channel availability is location-dependent and time-varying in this work, and (3) our algorithms may need to operate under the assumption of limited information exchange.

First, the objective is different. The traditional channel allocation problem is closely coupled with call admission control/handoff in circuit-switched cellular networks. When a call is generated (or a handoff occurs), a channel allocation algorithm is triggered to find a channel for the new call. The objective is to minimize the channel usage, while each call is assigned to one channel. Similarly, in most literature in list coloring, the objective is to find a coloring scheme such that each node can be assigned one channel from its channel list. On the other hand, the objective here is to fully utilize the available spectrum that is unused by the primary users while avoiding possible interference between neighboring nodes. The underlying assumption is that secondary users have elastic data traffic and can fully utilize the available bandwidth.

Second, the spectrum availability is different. In cellular networks, usually the available channels observed by the users are uniform and static because these channels are statically allocated to them. In our work, the available channels are location-dependent and time-varying because they are determined by the activities of the primary users. This makes our problem different from the network planning and graph coloring problems that are used to model the channel allocation in cellular networks.
Third, in our model, the channel availability may change before a channel allocation algorithm converges. Therefore, the channel allocation algorithms may have to work under scenarios where users have limited feedback information from their neighbors. The algorithm design needs to take this into account.

The multichannel wireless network is another emerging research area related to our work. The major difference between our work and research in multichannel wireless networks is that the available channels in multichannel networks are also statically allocated for all the nodes. So they are not location-dependent or time-varying. Another difference is that we assume in our work that neighboring nodes cannot use the same channel simultaneously, while in multichannel networks, neighboring nodes can and need to share the same channel (the average throughput lower if the channel is shared). Last, our focus is on the handling of the dynamics in channel availability.

3.1.2 Contributions

This chapter studies wireless networks with opportunistic spectrum availability and access. The contributions of our work are as follows:

- We present a framework to model the location-dependent and time-varying channel availability observed by secondary users that is by the activities of primary users. To the best of our knowledge, this is the first chapter to identify and model these two unique characteristics in such wireless networks. We also propose a new metric, the effective nonopportunistic bandwidth, to capture the inherent properties of white space.

- Building on this framework, we formulate the channel allocation problem as a list coloring problem and develop several distributed algorithms for channel sharing. We validate their performance under scenarios with location-dependent and time-varying channel availabilities. It is worth noting that our algorithms work well under scenarios where secondary users have limited capability of information exchange.

3.1.3 Organization

The chapter is organized as follows: we first introduce the framework and a new performance metric in Section 3.2. Using the framework, we formulate the channel allocation problem as a graph coloring problem in Section 3.3. Then we propose several distributed approaches for opportunistic spectrum sharing with various degrees of complexity in Section 3.4. A numerical study is presented in Section 3.5, followed by conclusions in Section 3.6.
3.2 A Framework for Opportunistic Spectrum Sharing

We first introduce a model for the channel availability observed by the secondary users. Note that such availabilities are location-dependent and time-varying, which is incurred by the activities of the primary users. We abstract each network topology into a graph, where vertexes represent wireless users such as wireless lines, WLANs, or cells, and edges represent interference between vertexes. In particular, if two vertexes are connected by an edge in the graph, we assume that these two nodes cannot use the same spectrum simultaneously. In addition, we associate with each vertex a set, which represents the available spectra at this location. Because of the differences in the geographical locations of the vertex, the sets of spectra of different nodes may be different. Furthermore, a node may observe time-varying channel availability due to the traffic load variation of the primary users.

In Figure 3.1, we show a model of such a network. The five 1–5 represent five different secondary or opportunistic users. There are three frequency bands, namely A, B, and C, which are communication channels that are opportunistically available to the secondary users (1–5 in this figure). We assume that all channels have the same bandwidth, which can be generalized easily. In addition, four primary users I–IV are present, using bands B, A, B, and C, respectively. Due to the sharing agreement, channels used by primary users cannot be utilized by secondary users in the vicinity. Therefore, we assume that nodes within a certain range of each primary user I–IV cannot reuse the same frequency. In other words, if a vertex is within the dashed circle of a specific primary user, it cannot access that band used by the primary user. For instance, node 2 is within the interference range of primary user I, who uses channel B. Therefore, channel B is not available for node 2. As a consequence, each node has access to a different set of band. In our figure, the available channels are (A,B,C) at vertex 1, (A,C)
at vertex 2, etc. The resource allocation problem is how they should share these bands.

Note that Figure 3.1 shows a snapshot of the network. Time-varying channel availability is introduced by the mobility of users (both primary and secondary) and the traffic load variation of primary users. For instance, in the numerical results, we introduce the time-varying channel availability at secondary users by varying the usage of primary users. We consider a time-slotted system. In a generic time slot, if a primary user occupies one channel, it will keep the same channel in the next time slot with probability \( p_{11} \); if a primary user is idle on one channel, it will occupy a channel in the next time slot with probability \( p_{01} \). Then the channel availability of a secondary user varies at each time slot depending on all the primary users. Such a policy can be considered as an approximate exponential ON/OFF traffic model for the primary users. Other traffic models can be introduced similarly.

When channel availabilities change, secondary users need to adjust their channel allocation accordingly. They may also need to exchange information with neighboring nodes. However, secondary users may have limited capability of information exchange and experience delay during information exchange because secondary users coexist in an ad hoc manner. This is different from cellular systems where dedicated (and private) signaling channels exist between cells. Limited information exchange imposes an additional challenge on opportunistic channel sharing algorithms.

Our framework can be generalized so that more sophisticated and realistic scenarios are taken into account. For instance, a more sophisticated interference model can be introduced, such that two connected secondary users may be able to use the same channel but with lower data rates because of the co-channel interference. In the model described above, the underlying assumption is that the size of the node, which can be a WLAN or a cell, is small compared to the distance between neighboring nodes. On other hand, the location and transmission power of each user in a WLAN can be taken into account at the cost of complexity. In addition, we can generalize the communication range of a user from a unit disk to a more general shape due to the randomness in the propagation environment.

We make the following assumptions in the paper. We assume that channel availabilities are given to secondary users, and focus on how to share the opportunistic spectrum among secondary users. We acknowledge that it is a challenging problem in itself to decide whether a spectrum can be used by the secondary users. The proposals for solving this problem include carrier sensing, signal-strength-based methods, beacon-based methods (a beacon indicates the availability or unavailability of a spectrum), location-and database-based mechanisms (a node has location information and can check a database about the availability of a spectrum), etc. Furthermore, we assume that secondary users have elastic data traffic and can fully
utilize the amount of spectrum being allocated. This is the benchmark case in terms of spectrum requirements. On the other hand, our proposed algorithms can easily adapt to the cases where nodes have (different) limited requirements. In particular, a node can stop requiring more channels after all its needs have been satisfied.

### 3.2.1 Effective Nonopportunistic Bandwidth

Consider the following question: suppose that there is 62% of white space under 3G. If we can fully utilize such white space, is it equivalent to gaining an additional spectrum band of \(0.62 \times 3\text{GHz}\)? The answer is that it depends. To address this question, we introduce the notion of **effective non-opportunistic bandwidth**. It is defined as the equivalent non-opportunistic bandwidth required to achieve the same performance as in the case of opportunistic spectrum availability. A nonopportunistic channel is referred to a channel that is always available to all (secondary) users in consideration, which is how spectrum is allocated in the traditional command-and-control manner. This metric is designed to study the impact of the opportunistic availability of spectrum.

We elaborate the idea with a naive example next. Consider a simple network with only two nodes, which cannot use the same channel simultaneously due to interference. Consider a channel with bandwidth \(W\) that is opportunistically available to these two nodes. Assume the channel is available at each node with probability \(p\) independently. Suppose that a user obtains one unit of throughput per unit of spectrum. Then the total throughput gained by the two users is:

\[
W(p^2 \times 1 + 2p(1 - p) \times 1 + (1 - p)^2 \times 0) = Wp(2 - p).
\]

The first term is the case where the channel is available to both nodes and only one of them can use it due to interference. The second term is the case where only one of the users has the spectrum availability and uses it. The last term is the case where the spectrum is available to neither of the two users. To achieve the same total throughput, \(B_e = Wp(2 - p)\) units of nonopportunistic bandwidth are required. To elaborate, when a spectrum of \(B_e\) is available to both users in the traditional way (i.e., the spectrum is always available to both users), the throughput is \(Wp(2 - p)\) because only one of the users can use it at any given time. Thus, we claim that \(B_e = Wp(2 - p)\) is the effective non-opportunistic bandwidth in this simple example. Suppose \(W = 3\text{GHz}\) and \(p = 62\%\). In this example, we see the impact of 62% white space under 3G is equivalent to \(Wp(2 - p) = 2.76\text{GHz}\) of spectrum in the traditional way.

Note that spectrum is not being “created” by secondary users. Instead, they simply explore the spectrum holes generated by the given usage...
pattern of primary users. Primary users are legacy users, whose behavior we do not change. The inherent characteristics of the primary users, such as communication range, transmission power, traffic pattern, node density, and topology, determine the spectrum opportunities of secondary users. The notion of effective non-opportunistic bandwidth is a metric to quantify the potential of such spectrum opportunities for a given topology of secondary users.

The intuition is similar to that of the effective bandwidth used to capture the statistical multiplexing gain. However, what is being captured here is the degree of spatial reuse given the correlation of spectrum availabilities at secondary users. For instance, in the above example, if two secondary users are very close and also observe the same spectrum opportunity, then $B_e = Wp$ instead of $Wp(2 - p)$ in the independent case. In general, because of the characteristics of primaries mentioned above, users observe different channel availabilities, which in general yield higher gain, which is quantified by the effective nonopportunistic bandwidth.

The above example is oversimplified, but serves to illustrate the idea. In general, the effective nonopportunistic bandwidth is difficult to calculate because the network topologies are more complicated and the availabilities of channels at different nodes are correlated. Thus, we quantify such effects using numerical results in Section 3.5. In summary, this metric is introduced to capture the inherent impact of opportunistic spectrum availability.

### 3.3 Problem Formulation

Using the model described earlier, we formulate the channel allocation problem as a graph coloring problem. We abstract the network as an undirected graph $G = (V, E, L)$, where vertexes represent users and edges represent interference, so that no channels (frequency bands) can be assigned simultaneously to any adjacent nodes. For simplicity, we assume that the interference graph is the same for all frequency bands. This can be generalized to the case where each frequency has its own interference graph, a possible scenario due to the different propagation properties in the environment associated with individual bands. Furthermore, let $K$ be the number of available channels in $G$. Although it is possible that different channels have different bandwidths, we treat all channels the same for simplicity. We also refer to the graph $G$ as the interference graph. In the paper, we use “channel” and “color” interchangeably.

Let $N = |V|$ denote the total number of users. Let edges be represented by the $N \times N$ matrix $E = \{e_{ij}\}$, where $e_{i,j} = 1$ if there is an edge between vertexes $i$ and $j$, and $e_{i,j} = 0$ implies that $i$ and $j$ may use the same frequencies. Note that since $G$ is an undirected graph, $E$ is symmetric. In a
similar notation, we represent the availability of frequencies at vertexes of $G$ by an $N \times K$ matrix $L = \{l_{ik}\}$, which we refer to as the coloring matrix. In particular, $l_{ik} = 1$ means that color (channel) $k$ is available at vertex $i$, and $l_{ik} = 0$ otherwise. For instance, Figure 3.1 is represented by the matrices

$$E = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix}, \quad L = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix}.$$ 

Let us denote a color/channel assignment policy by an $N \times K$ matrix $S = \{s_{ik}\}$, where $s_{ik} = 0$ or 1, and $s_{ik} = 1$ if color $k$ is assigned to the node $i$ and 0 otherwise. We call $S$ a feasible assignment if the assignments satisfy the interference graph constraint and the color availability constraint. More specifically, for any node $i$, we have $s_{ik} = 0$ if $l_{ik} = 0$ (i.e., a color can be assigned only if it is available at the node). Furthermore,

$$s_{ik}s_{jk}e_{ij} = 0, \quad \forall i, j = 1, \ldots, N, k = 1, \ldots, K.$$ 

In other words, two connected nodes cannot be assigned the same colors.

The objective of the resource allocation is to maximize the spectrum utilization. This problem can be formally represented as the following nonlinear integer programming problem.

$$\max_S \sum_{i=1}^{N} \sum_{k=1}^{K} s_{ik} \quad (3.1)$$

subject to $s_{ik} \leq l_{ik},$

$$s_{ik}s_{jk}e_{ij} = 0,$$

$$s_{ik} = 0, 1,$$

for all $i, j = 1, \ldots, N, k = 1, \ldots, K$. The above problem is sometimes referred to as a list multicoloring problem. When time is taken into account, a time index can be introduced into the equation where the objective is to maximize the utilization averaged over time and the three constraints are satisfied at each time instant.

The corresponding decision list coloring problem is formulated below.
Problem 3.1 (DListColor Problem) Given a graph $G = (V, E, L)$ and a positive integer $B$, is there a solution such that

$$\sum_{i=1}^{N} \sum_{k=1}^{K} s_{ik} > B,$$  \hspace{1cm} \text{(3.2)}

with the same set of constraints as in Equation (3.1)?

Proposition 3.1

The DListColor problem is NP-complete.

Proof This problem is clearly in NP since once a valid coloring assignment $S$ is obtained, condition (3.2) may be verified in $O(|V| \cdot K)$ time.

We now show that the maximum clique problem can be reduced to the DListColor problem in polynomial time, and that the maximum clique problem has a solution if and only if DListColor has a solution.

Let $G = (V, E)$ be the undirected graph of the maximum clique problem. We construct the graph $G' = (V', E', L)$ for our DListColor problem, such that $V' = V$, and $E'$ is the complementary set of $E$. Furthermore, the color matrix $L$ is of dimension $|V| \times 1$, where $L = [1, 1, \ldots, 1]^T$. Since any pair of nodes connected in $G$ are not connected in $G'$ and vice versa, we cannot simultaneously assign nodes in $G'$ the same color if these nodes form a clique in $G$. Therefore, there exists a clique in $G$ of size at least $m$ if and only there is no solution for DListColor for $B = |V| - m$. This reduction is obviously polynomial-time.

Q.E.D.

3.3.1 Color Decoupling

The list coloring problem may be reduced to a set of maximum-size clique problems when fairness is not a consideration. In other words, in the process of finding the maximum in Equation (3.1), nodes are allowed to be assigned zero channels. The problem of assigning each node a set of colors may be solved by coloring the graph in sequence with individual colors:

$$\text{maximize } \sum_{i=1}^{N} \sum_{k=1}^{K} s_{ik} \leftrightarrow \sum_{k=1}^{K} \text{maximize } \sum_{i=1}^{N} s_{ik},$$  \hspace{1cm} \text{(3.3)}

where $S_k$ denotes the channel allocation with respect to channel (color) $k$. More specifically, $S_k$ is the $k$th column in the assignment matrix $S$. Note that the equality in (3.3) does not hold in general situations, e.g., a graph

* A clique is a fully connected subgraph; i.e., a clique consists of a set of nodes any pair of which has an edge in between.
coloring problem that requires each node to be colored with nonempty colors. Note that when fairness is taken into account, e.g., each node has to be assigned at least one color, then the decoupling property does not apply.

3.4 Proposed Algorithms

In this section, we discuss several approaches to the resource allocation problem formulated above. We prefer distributed algorithms because of their robustness and scalability. We use a brute force search algorithm with which we find optimal solutions to serve as a benchmark. Because the resource allocation problem is NP-complete, optimal solutions may only be found when graphs are relatively small. We then present a distributed greedy algorithm, a distributed fair algorithm, and a distributed randomized algorithm, with various complexity and performance.

3.4.1 Optimal Solutions: Benchmark

Given the list coloring problem and a graph \( G = (V, E, L) \), we seek the solutions to optimization problem (3.1). As discussed in Section 3.2, the optimization problem is NP-complete. Therefore, in order to find the optimal solution(s), we must search through all valid color assignments, and find the one(s) that maximizes (3.1).

We carry out this search in a breadth-first recursive manner, with the starting node chosen arbitrarily. More specifically, when the node \( i \) is visited, we enumerate all combinations of channel allocations for this node permissible by the available channels at \( i \), and iterate through each configuration and tentatively assign it to the node. If there is a conflict between the current assignment attempt and neighboring nodes whose channels have already been selected, we abort this assignment. The complexity of this algorithm is \( \prod_{k}^{N} 2^k = O(2^{NK}) \). This algorithm is easily modified to find optimal solutions under additional constraints such as fairness, variance of resource allocation, etc. For example, suppose that we seek to maximize (3.1) but with the constraint that each node has at least one channel assigned (if such an assignment exists). We may modify the process so that zero channel assignment is not a permissible option.

The complexity can be reduced if we use the color decoupling property described in Section 3.3.1 and modify our search algorithm so that the color assignment is optimized subsequently for each color. The resulting algorithm has a complexity of \( O(K 2^N) \). However, this reduction of complexity is not achieved without penalty. For example, since channels are assigned independently of each other, it is no longer straightforward to find a policy that assigns each user at least one channel.
3.4.2 Distributed Greedy Algorithm

Because the resource allocation problem is NP-complete, heuristics are needed to study large graphs. In this section, we present a distributed greedy algorithm with the objective of maximizing the utilization.

Because of the color decoupling property discussed in Section 3.3.1, the distributed greedy algorithm handles colors one by one. For each color, a greedy assignment is calculated to maximize the number of nodes assigned to this color. To elaborate, consider the assignment of the color $i$. A subgraph $G_i = \{V_i, E_i\}$ is generated, where a node belongs to $V_i$ if and only if it belongs to $V$ and color $i$ can be used at the node. An edge connects two nodes in $V_i$ if and only if these two nodes are connected in the original graph $G$. For instance, for the case presented in Figure 3.1, the subgraph for color A consists of nodes 1, 2, and 5, and a link between 1 and 2.

The following is the description of the greedy algorithm.

1. Each node looks up the color $i$ in its available color list. It finds the nodes in its neighbor list that also have this color available, denoting this sublist as $N_i$ (including itself).
2. If its link degree is the smallest in $N_i$, it picks the color $i$. Note that the link degree here means the number of nodes it connects to that also have this color available. If not, it will look up the next available color and go back to (1).
3. If one or more neighboring nodes has the same smallest link degree (a tie exists), it uses its color degree to break the tie. The color degree means the number of available colors a node has. If its color degree is the smallest among the nodes with the tie, it will also pick color $i$. Otherwise, it will look up the next available color and go back to (1).
4. If a tie again exists, the nodes involved will generate random numbers independently. If this node has the smallest random number, it will pick the color. Otherwise, it will switch to the next available color and go back to (1).
5. It stops when all of its available colors are processed.

The greedy algorithm performs as if nodes are ranked according to their link degrees from low to high for each color. Then the color is assigned to the nodes according to their link degrees from low to high. When a tie exists, the number of assigned colors of each node is used to break the tie. Nodes with fewer assigned colors have higher priority. If the nodes have the same number of assigned colors, ties are broken randomly. The worst-case complexity of the algorithm is $O(KN^2)$, where $K$ is the number of colors and $N$ is the total number of nodes. The algorithm can result in very unfair allocation. Nodes with lower link degrees will obtain more resources in general.
3.4.3 Distributed Fair Algorithm

As discussed in the previous section, the greedy algorithm can result in very unfair allocation. In this section, we discuss a distributed algorithm with fairness considerations. The algorithm has three steps.

Step 1 is to build an acyclic directional graph. The building of the acyclic directional graph is motivated by Ref. 9, although the link degrees of nodes are not taken into account in Ref. 9 and no iteration is required due to the difference in objectives. All nodes exchange information about their link degree, color degree (color degree is the number of available colors at each node), and a random number. The edges are oriented from higher color degree to lower; i.e., nodes with a smaller number of colors are the receivers. If two connected nodes have the same number of colors, the edge is oriented from higher link degree to lower link degree. If there is a tie again, the edge is oriented from the node with the larger random number. Thus, an acyclic graph is generated. Figure 3.2 shows the graph for the case presented in Figure 3.1. A node is a sink node if there is no edge oriented from it. A node is a source node if there is no edge oriented to it. Note that it is possible that there are multiple source and sink nodes. Node x is node y’s upstream neighbor if the edge is oriented from x to y, and downstream otherwise. Coloring starts from sink nodes to source nodes as described in Step 2.

The heuristics of the direction is as follows: nodes with more available colors should yield to nodes with fewer colors to maintain a balance. When this tie, link degrees are used to break leads to a tie. A node with lower link degree has higher priority so that fewer nodes are affected by its color selection. If it ties again, then the random numbers chosen by the nodes are used to break ties. Note that each node has one fixed random number in each iteration to avoid loops. In comparison, the fair algorithm takes into account the color degree first and link degree second, while the greedy algorithm takes the opposite order. This results in different performance in terms of fairness and throughput.

Step 2 is to assign colors. At most one color is assigned to each node in one iteration. The color assignment starts with sink nodes. If a node is

![Figure 3.2 Graph.](image-url)
a sink node, it picks a color that can be used by the minimum number of neighbors. It updates this information to all neighbors, who remove the color from their available color lists. In addition, a set-color token is passed to the neighbors to enable their color searching. If a nonsink node obtains set-color tokens from all its downstream neighbors, the node becomes a sink node and repeats the same process. Thus, the color selection is performed from sink to source nodes step by step.

Step 3 is to start the next iteration if needed. After a source node performs the process, it generates a reset token and passes it to all its neighbors. If a node receives reset tokens from all its upstream neighbors, it sends a reset token to all its downstream neighbors. After all nodes receive a reset token, the system resets and goes to step 1 with the remaining colors. If a node is out of available colors, it quits the operation. When no nodes have available colors remaining, the operation stops.

As shown in Figure 3.2, in the first iteration, sink nodes 3 and 5 select colors first (both select color C). Then the tokens are passed to nodes 2 and 4, who select colors A and B respectively. Last, the source node 1 selects color C. After the reset process, explained in Step 3, the second iteration begins, where node 5 picks color A and ends the process. Note that in each iteration, a new graph is generated to reflect the remaining color availability.

In each iteration, at least one color will be assigned to a node. Thus, there are at most $O(NK)$ iterations. Assume the maximum link degree is $\Delta$. The complexity of the scheme in each iteration is $O(\Delta NK \log K)$. Thus, the worst-case complexity is $O(\Delta N^2 K^2 \log K)$. In general, in each iteration, more than one color is assigned and the average complexity is better. The communication cost per iteration is $O(N)$ where broadcast is assumed.

3.4.4 Randomized Distributed Algorithm

The above distributed algorithms may need a large number of iterations when a large number of nodes and colors are involved. When the size of the network is large, large communication overhead will occur. So the distributed randomized algorithm is proposed to reduce the delay and communication cost. The algorithm is inspired by the IEEE 802.11 backoff algorithms in MAC protocols, although the objective is different. In 802.11, the station doubles its contention window to reduce the probability of collision when its transmission fails. In our algorithm, a node will increase its chance to win the next color when it fails to get a color contending with its neighbors. The description of the randomized algorithm are as follows:

1. Each node first generates a random number for each of its available colors uniformly distributed from $[0, \text{window}]$. At the beginning, window is set to START, which is a relatively small value (e.g., 1).
Then, each random number is divided by the color degree of a node.

(2) Nodes exchange information the random numbers and the available colors with their neighbors.

(3) Within one round, the node goes through all its available colors. If it has the highest random number among all its neighbors for one color, it will win the contention and be able to use the color from then on.

(4) At the beginning of the next round, each node exchanges information with its neighbors on what colors it gets. Each node deletes the colors that are obtained by its neighbors from its available color list. Each node then updates its window by the following rule: if it loses one color, it doubles its window (i.e., window = 2 * window); if it wins one color, it divides its window by 2 (i.e., window = window / 2).

(5) There is also a STOP value for the window evolution. The window value cannot exceed the STOP value even if a node keeps losing contentions. STOP can be several times the value of START. For example, STOP = 8 * START.

(6) After recomputing the window, each node regenerates the random numbers for its remaining colors and goes into the next round.

The randomized distributed algorithm has a small communication overhead. It also converges faster compared with the distributed fair algorithm, especially when the number of nodes and colors are large. For the first set of random numbers, we make it inversely proportional to the color degree of each node so that the node with a lower color degree should have a higher chance to pick colors in the first round. In the following rounds, the intuition is that the node that loses the contention should have a higher chance to win in the next round. On the other hand, the node that wins the contention should lower its chance to win in the next round. We may want to limit the window evolution within a certain range to avoid the situation that a node may have a very high chance to win its remaining colors because it loses many colors in the previous round. Thus, a certain level of utilization and fairness can be achieved.

### 3.5 Numerical Results

In this section, numerical results are used to illustrate the impact of the location-dependency and time-variance of the channel availability and evaluate the performance of the proposed algorithms. We focus on the following aspects: (1) the potential of opportunistic resource availabilities vs. fixed resource availability, (2) the impact of the characteristics of primary users,
and (3) the effect of time-varying channel availability and limited information exchange on the proposed algorithms. We first study a snapshot of the network. This means that the channel availability is fixed during the execution of the algorithms. Thus, the impact of the location-dependency of the channel availability is illustrated. Then we will introduce the time-variance into the channel availability and evaluate its impact.

We measure utilization by computing the average number of channels (colors) assigned to each vertex, which is also referred to as utility. The fairness metric is the variance of channel allocations in each assignment. The smaller the variance, the fairer the assignment. If all nodes obtain the same number of channels in an assignment, then the fairness metric is zero. In general, the fairness metric is not zero even if the assignment is max-min fair, because nodes have different available channels and it may be impossible for all nodes to obtain the same number of channels. The third metric used is the effective non opportunistic bandwidth discussed in Section 3.2, which is defined as the equivalent non opportunistic bandwidth required to achieve the same performance as the opportunistic spectrum band use.

### 3.5.1 Simulation Setup

Given a topology geographically, we use the following model to generate a snapshot of the channel availability for the nodes of interest. Each channel has \( N_I \) primary users (also referred to as interferers) that are uniformly distributed in a unit square area. With probability \( p_I \), each of the primary users is active, independently of other users. A channel is available at a vertex (the secondary user) if and only if it is not within the interference range \( R_I \) of any active primary users of the channel. We generate the availability of \( K \) different channels independently. This is similar to the scenarios shown in Figure 3.1 except that all primary users have the same interference range, \( R_I \). The purpose here is to generate correlated channel availability profiles; i.e., nodes closer to each other are likely to have similar channel availabilities. When a node in the graph represents a cell with a positive geographical size, such a generation scheme is less precise because the size of the cell is not taken into account. Nonetheless, the main purpose is to generate correlations among the channel availabilities of nodes, and thus similar results should be observed. We will present the way to generate the time-varying channel availability in a later part of the discussion.

We illustrate the results in two fixed topologies. In the fixed topology, unless otherwise specified, we set \( R_I = 0.1, \ p_I = 0.2 \) and \( K \), the total number of channels in the network, to be 15. The first topology is a simple six-node ring as shown in Figure 3.3. The objective here is to study symmetric topologies, where all nodes are identical: each node has two
neighbors and the same characteristics of available colors (in a statistical sense). Although not shown here, we observe that all six users obtain roughly the same amount of bandwidth averaged over simulations.

The second topology shows a combination of symmetric and asymmetric nodes, as in Figure 3.4. In particular, node 4 is in the worst position, which interferes with all other nodes. Nodes 1–3 and 5–6 are symmetric and so are nodes 7 and 8.
We note that simulations have also been performed using a large number of random topologies. Similar observations and conclusions have been yielded and thus are not included.

3.5.2 Performance under the Snapshot of the Network

We first show the performance of the proposed algorithms under the snapshot of the network. We generate 100 different snapshots of the network for each topology. The performance is averaged over all nodes and over all snapshots.

**Ring:** Figure 3.5 compares the average performance (average over all nodes and over all simulations) of the four algorithms in different sets of parameters in the ring topology. The first subplot compares the average utility (number of channels assigned per node) of the four algorithms. We observe that in all cases, the greedy algorithm can always achieve the same utilization as the optimal solution, followed by the randomized and fair algorithms. With the increase of $N_I$, each node observes fewer available channels (‘avgcolor’), and thus lower average utility. As an illustration of
Exploring Opportunistic Spectrum Availability

fairness, the second subplot shows the variance among different nodes of the four algorithms, which is the variance of each allocation scheme averaged over 100 simulations. The fair algorithm has the lowest average variance value and thus a claim to being the most fair. The random and greedy algorithms have similar performance. In summary, in the ring topology, the greedy algorithm can achieve the best utilization at the cost of relatively high variance. The fair algorithm can achieve the smallest variance, but also with the smallest utilization. The randomized algorithm has the moderate performance among the three. The advantage of the randomized algorithm is that it has the smallest communication overhead and convergence time. Such performance reflects the differences in the design principles of the three algorithms.

The last subplot includes the average number of channels available per node, its variance, the average number of neighbors sharing the color, and the effective nonopportunistic channel. They are “avgcolor”, “varcolor”, “colorinf”, and “non-opp”, respectively, in the legend. The first three metrics reflect the characteristics of the channel availabilities. In particular, “avgcolor” is the average number of channels available per node. It is equivalent to $3 \times 0.62$GHz in the case where there is 62% of white space under 3GHz. It is a measure of channel availability. Its comparison with effective nonopportunistic bandwidth, denoted by “non-opp”, is to help us understand the question whether or not that 62% of white space under 3GHz is equivalent to an additional band of $3 \times 0.62$GHz. The second metric, “varcolor”, is the variance of color availability in each snapshot. The third metric “colorinf” is related to the likelihood a node obtains a channel. For instance, if a channel is available to nodes 1–3, then the value at node 2 is 3 (including itself, three nodes will share the channel). The larger the value is, the less the average utilization (for a given topology) higher-level competitions. The metric drawn in the figure is averaged over all nodes and all colors.

In this topology, the effective bandwidth is calculated as follows: one nonopportunistic frequency that is available to all nodes can be used by every other node. Thus, to assign an average $m$ channels to each node, the required number of effective nonopportunistic channels is $2m$. When the number of interferers is small (e.g., $N_I = 20$, the leftmost node), a color is likely to be available to most nodes, and thus the average channels per node is 13 with a small variance. Recall that the total number of channels is 15. In this case, the value of the effective nonopportunistic channel is close to that of average available channels. As the number of interferers increases channel availability to different nodes varies more, and the difference between the effective bandwidth and the average number of channels available per node is larger. This indicates that the impact of the opportunistic bandwidth allocation is more significant when the variance of the channel availability is high. This is intuitive and also also observed in other simulation results.
Hexagon–triangle: Figure 3.6 shows the result of the simulation of the hexagon–triangle (H–T) topology using the same set of parameters as in Figure 3.5. The figure is drawn using the same notations. The effective bandwidth in the H–T topology is calculated as follows:

\[
B_e = B_4 + \max \left( \sum_{i=1,2,3,5,6} B_i, \sum_{i=7,8} B_i \right),
\]

(3.4)

where \( B_i \) is the average number of channels of node \( i \). The rationale is that when node 4 is using a channel, no one else can use it. Similarly, only one of nodes 1, 2, 3, 5, 6 can use the channel, and one of nodes 7 and 8 can use the channel. In this simulation, the value of the utility of the fair algorithm is used to calculate the effective bandwidth shown in the third subplot, which is in fact a lower bound of the effective bandwidth because we cannot claim the optimality of the fair algorithm.
For the utilization and variance performance, compared with the ring topology, the relative order of the three algorithms is more obvious. The greedy algorithm can achieve the best utilization at the price of the highest variance. The fair algorithm has the lowest variance, but also with the lowest utilization. The randomized algorithm performs in between.

To further understand the principle of the algorithms, let us take a look at Figure 3.7. It shows the performance of different algorithms at different nodes. The four bars from left to right are the results of the fair, randomized, greedy, and optimal algorithms. Since node 4 is the node that interferes with the greatest number of other nodes, the optimal and greedy algorithms allocate least spectrum to it in order to maximize the total utilization. In contrast, the fair and randomized algorithms allocate much more spectrum to node 4 so that they are fairer than the optimal and greedy algorithms. For the fair algorithm, node 4 has a slightly lower allocation compared with other nodes because it is more likely to run out of colors first. A similar result is also observed for the randomized algorithm, where node 4 needs to compete with 8 other nodes.

For the last subplot in Figure 3.6, we can see that the effective bandwidth decreases much slower than that of the average utility per node as $N_I$ increases. This is due to the fact that the term $\sum_{i=1,2,3,5,6} B_i$ decreases slowly as $N_I$ increases because there is always a good chance a channel is available to at least one of these five nodes. This illustrates that the impact of opportunistic channel availability is more significant on dense topologies (the clique of nodes 1–6) than on sparse topologies in terms of the difference between effective bandwidth and average channel availability.*

* It is arguable that the effective bandwidth should be calculated using the total utility and should not distinguish among nodes. Our choice is to emphasize the variability of the topology.
3.5.3 Performance under Time-Varying Channel Availability

In the above studies, we focused on the snapshot of the system where the channel availability of the primary users is fixed after it is generated. In practice, the channel availability is time-varying the traffic activity of the primary users. So it is essential to study the impact of the time-varying channel property on the performance of the proposed algorithms.

We introduce the time-varying channel availability at secondary users by varying the usage of primary users as discussed in Section 3.2. In the simulation, we set $p_{11} = 0.8$ and $p_{01} = 0.2$. Such a policy can be considered as an approximate exponential ON/OFF traffic model for the primary users with an average ON/OFF period of 5 time slots.

We assume that in each time slot, secondary users can only exchange information once. This can happen when the changes in channel availability are relatively fast and the information exchange channel between secondary users has limited bandwidth and/or experience delay. This constraint is imposed to test the “real-time” performance of the proposed algorithms that may not converge in one iteration. That is, after one time slot, not all the available channels will be assigned. We call this partial assignment real-time performance. We are interested in the difference between the real-time performance and offline converged performance, i.e., the difference between the partial channel allocation done in one time slot and the converged result, where the latter is what we get under the snapshot of the network. Such a difference reveals the impact of the time-varying channel availability on the performance of the proposed algorithms. The smaller the difference, the better the performance when the proposed algorithms are used under time-varying channel availabilities.

Figure 3.8 and Figure 3.9 show the performance comparisons among three distributed algorithms in the ring topology and H-T topology, respectively. We do not show the performance of the optimal algorithm because it only serves as a benchmark under the snapshot of the network. In the simulation, the total number of channels is still 15, and the number of primary users is 80.

As shown in the figures, the randomized algorithm has the smallest difference in utilization between the real-time and converge performance, followed by the greedy algorithm. The fair algorithm has the largest difference between the real-time and converge performance. Recall that the randomized algorithm is designed with the objective of small communication overhead and short convergence time. It means that the randomized algorithm allocates most of the available channels in the first iteration. On the other hand, due to the complexity of the fair algorithm, it can only allocate a small portion of its available channels in one iteration, leading to the large difference. The small variance of the fair algorithm is caused by the small value of the assigned colors. Since the variance is defined as
the standard deviation, the variance could also be small when the value of
the assigned colors is small.
From the above results, we can conclude that the time-varying channel availability does affect the performance of the proposed algorithms. But the difference between real-time and converge performance is not

---

**Figure 3.8** Real-time vs. convergence in the ring topology.

**Figure 3.9** Real-time vs. convergence in the H-T topology.
very large, especially for the randomized and greedy algorithms. Under the time-varying channel availability, the randomized algorithm has the smallest difference between the real-time and converge performance due to its low complexity. The fair algorithm has the largest difference and the greedy performs in between.

In a network with opportunistic spectrum availability and access, the channel allocation algorithm may only have limited information exchange before the current channel availability changes. Therefore, the algorithm with low complexity and communication overhead is preferred due to the time-variance in channel availability.

3.6 Conclusions

In this chapter, we present a framework to illustrate the relationship between channel availability of secondary users and the usage of primary users. We pointed out that certain properties are inherent to systems with opportunistic spectrum availability and are independent of resource allocation schemes. The first is the location-dependency. As the channel availability observed by the secondary users is determined by the primary users, neighboring secondary users tend to have similar channel availability. The second is the time-variance. Due to the traffic load of the primary users, a secondary user may observe different channel availability over time. These two characteristics are unique to open-spectrum systems and are captured in this framework. We also introduce a new metric, effective nonopportunistic bandwidth, to quantify the impact of white space. Although the intuition of the spectrum sharing is clear—spatial reuse and statistical multiplexing improve throughput— our work sheds light on quantifying such improvements.

Based on the framework, we formulate the channel allocation problem as a list coloring problem with the objective of maximizing the total spectrum utilization. We show the optimal allocation problem is NP-complete, and develop several distributed algorithms. The proposed distributed greedy algorithm achieves close to optimal resource utilization. In addition, a distributed fair algorithm is proposed that achieves better fairness while maintaining a good level of spectrum utilization. Last, a distributed randomized algorithm is introduced with low complexity and communication overhead. The performance of the proposed algorithms is validated in scenarios with time-varying and location-dependent channel availability. In particular, the greedy and the randomized algorithms are shown to remain effective under scenarios where users have only limited information exchange capability. Due to the time-variance of the channel availability, algorithms with low complexity and communication overhead are preferred.
The chapter is an attempt to understand the impact of dynamic spectrum availability and the opportunistic exploration of it. Further studies include performance analysis, general definitions of utilization and fairness metrics, and experimental tests.

References

[3] FCC. Facilitating opportunities for flexible, efficient, and reliable spectrum use employing cognitive radio technologies, notice of proposed rulemaking and order, FCC 03-322.
[4] FCC. Establishment of an interference temperature metric to quantify and manage interference and to expand available unlicensed operation in certain fixed, mobile and satellite frequency bands, ET Docket No. 03-237, FCC 03-289.


