Dynamic Channel Sharing in Open-Spectrum Wireless Networks

Wei Wang and Xin Liu
Department of Computer Science
University of California, Davis
Davis, CA 95616, U. S. A.
{wangw, liu}@cs.ucdavis.edu

***Abstract*— The current fixed spectrum allocation scheme leads to significant spectrum white spaces. It requires a more effective spectrum allocation and utilization policy, which allows unused parts of spectrum to become available temporarily for commercial purposes so that the scarcity of the spectrum can be largely mitigated. This paper is an early attempt to study such wireless networks with opportunistic spectrum availability and access. We studied the dynamics in the available channels caused by the location and traffic load of the primary users and proposed several distributed algorithms to exploit the available channels for secondary users. The performance of different algorithms is evaluated in networks with static and time-varying channel availability.**

I. Introduction

The current fixed spectrum allocation scheme can lead to significant spectrum white spaces (spectrum that is not used at all for some time). Experiments conducted by Shared Spectrum Company indicate as much as 62% of white space below 3GHz band even in the most crowed area near downtown Washington DC, where both government and commercial spectrum usage are intensive [1].

Such low utilization and increasing demand for the radio spectrum suggest a more effective spectrum allocation and utilization policy, which allows unused parts of spectrum to become available temporarily for commercial purposes so that the scarcity of the spectrum can be largely mitigated.

In this paper, we focus on the *opportunistic exploration of the white space* by users other than the primary licensed ones on a non-interfering 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., [2], [3]. Further, 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 refer to spectrum users who are not owner of the spectrum and operate based on agreements/etiquettes 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 paper.) 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 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.

II. A Framework for Open Spectrum Channel Sharing

We first introduce a model for 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 interferences 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. Due to the differences in the geographical location of each 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 1, we show a model of such networks. The five vertexes 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 (vertexes 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.

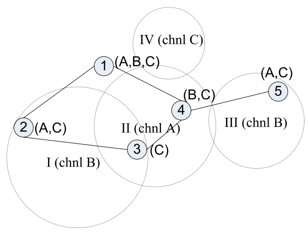


Fig. 1. A color graph

Due to the sharing agreement, channels used by primary users cannot be utilized by secondary users in vicinity. Therefore, we assume that nodes within certain ranges of the primary users 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 bandwidth used by that 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 bandwidths. 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 bandwidth.

Note that Figure 1 shows a snapshot of the network. In practice, time-varying channel availability could be 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 the probability p_{11}; if a primary user is idle on one channel, it will occupy a channel in the next time slot with the probability p_{01}. Then, the channel availability of a secondary user varies at each time slot depending on all the primary users. Such 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 ability of information exchange and experience delay during information exchange because secondary users coexist in an ad-hoc manner. This is different from celluar systems where dedicated (and private) signalling channels exist between cells. Limit information exchange imposes an additional challenge on opportunistic channel sharing algorithms.

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 of 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.

III. Problem Formulation

Using the model described earlier, we formulate the channel allocation problem as a graph coloring problem. We abstract the network as a undirected graph $G = (V, E, L)$, where vertexes represent users, and edges represent interferences 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 bandwidth. 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 same frequencies. Note that since G is an un-directional graph, E is symmetric. In a similar notation, we represent the availability of frequencies at vertexes of G by a $N \times K$ 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 1 is represented by the matrices

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

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,

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

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

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

$$\begin{aligned} \underset{S}{maximize} \quad & \sum_{i=1}^{N}\sum_{k=1}^{K} s_{ik} \\ \text{subject to} \quad & s_{ik} \leq l_{ik}, \\ & s_{ik}s_{jk}e_{ij} = 0, \\ & s_{ik} = 0, 1, \end{aligned} \tag{1}$$

for all $i, j = 1, \cdots, N, k = 1, \cdots, K$. The above problem is sometimes referred to as a list multi-coloring 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 instance.

The corresponding decision list-coloring problem is formulated below.

Definition 1: ($DListColor$ Problem) Given a graph $G = (V, E, L)$ and a positive integer B. Is there a solution such that

$$\sum_{i=1}^{N}\sum_{k=1}^{K} s_{ik} > B\,, \tag{2}$$

with the same set of constraints as in Eq. (1)?

Proposition 1: The $DListColor$ problem is NP-complete.

Proof: This problem is clearly in NP since once a valid coloring assignment S is obtained, condition (2) may be verified in $O(|V| \cdot K)$ time.

We now show that the *maximum clique* problem[1] 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 complimentary 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.

[1]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.

A. 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 Eq. (1), nodes are allowed to be assigned zero channels. The problem of assigning each nodes with a set of colors may be solved by coloring the graph in sequence with individual colors:

$$\underset{S}{maximize} \sum_{i=1}^{N}\sum_{k=1}^{K} s_{ik} \Leftrightarrow \sum_{k=1}^{K} \underset{S_k}{maximize} \sum_{i=1}^{N} s_{ik} \tag{3}$$

where S_k denotes the channel allocation with respect to channel (color) k. More specifically, S_k is the kth column in the assignment matrix S. Note that the equality in (3) does not hold in general situations, e.g., a graph coloring problem that requires each node to be colored with non-empty 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.

IV. 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.

A. Optimal Solutions: Benchmark

Given the list-coloring problem and a graph $G = (V, E, L)$, we seek the solutions to the following optimization problem (1). As discussed in Section II, 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 (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 combination 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 conflict between the current assignment attempt and neighboring nodes whose channels have already been selected, we abort this assignment.

The complexity can be reduce if we use the color decoupling property described in Section III-A, 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.

B. 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 propertied discussed in Section III-A. 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 1, the subgraph for color A consists of node 1, 2, and 5, and a link between 1 and 2.

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 less assigned colors has higher priority. If the nodes have the same number of assigned colors, ties are broken randomly. The algorithm can result in very unfair allocation. Node with lower link degrees will obtain more resource in general.

C. Distributed Fair Algorithm

As discussed in the previous section, the greedy algorithm can result in very unfair allocations. 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 [7], although the link degrees of nodes are not taken into account in [7] 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 color at each node), and a random number. The edges are oriented from higher color degree to lower; i.e., nodes with smaller number of colors are the receivers. If two connected nodes have the same number of colors, the edge is oriented from high 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 directional graph is generated. 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.

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 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 non-sink 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 performs 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 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 reset token, the system resets and go to step 1 with their remaining colors. If a node is out of available color, it quits the operation. When no nodes have available colors remaining, the operation stops.

D. 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.

Each node generates a random number for each of its available colors uniformly from [0, $window$]. 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 then wins the color. Then each node exchanges information with its neighbors of what colors it gets and deletes the colors obtained by its neighbors from its own available color list. If it loses one color, it doubles its $window$; If it wins one color, it divides its $window$ by 2. Then each node recomputes its random number from [0, $window$] and start 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.

V. 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 first show study a snapshot of the network. It means that the channel availability is fixed during the executions 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

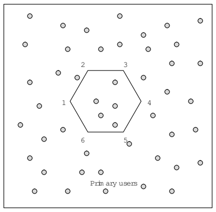


Fig. 2. Ring Topology

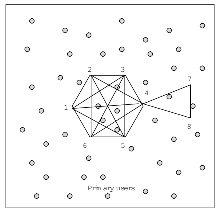
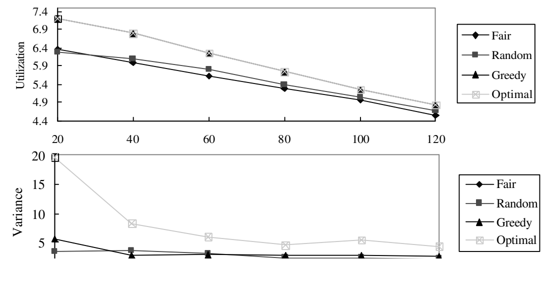


Fig. 3. Hexagon-Triangle Topology



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.

A. Simulation Setup

Given a topology geographically, we use the following model to generate a snapshot of the channel availability for the nodes of interests. Each channel has N_I primary users (also refereed to as interferers) that are uniformly distributed in a unit square area. With probability p_I, each of the primary users is active, independent 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 1 except that all primary users have the same interference range, R_I.

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 2. 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. 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 do nodes 7 and 8.

B. 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 4 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

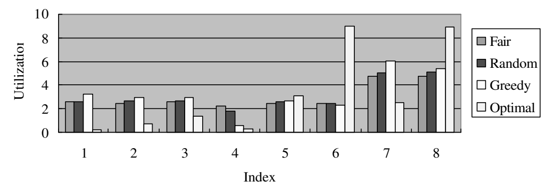


Fig. 5. Hexagon-triangle topology bar

node observes less available channels, and thus lower average utility. As an illustration of fairness, the second subplot shows the variance among different nodes of the four algorithms, which is the variance of each allocation schemes averaged over 100 simulations. The fair algorithm has the lowest average variance value and thus claims the most fair. The random and greedy algorithm 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 difference in the design principle of the three algorithms.

Hexagon-triangle: The performance of the three algorithms is similar to that under the Ring topology, which is ignored here. To further understand the principle of the algorithms, let us take a look at Figure 5. 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 which interferes the most 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. Similar result is also observed for the randomized algorithm, where Node 4 needs to compete with 8 other nodes.

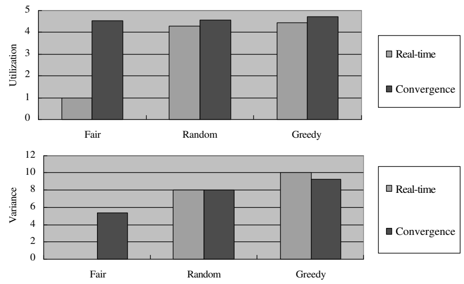


Fig. 6. Real-time vs. convergence in the Ring topology

C. Performance Under the 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 due to 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 II. In the simulation, we set $p_{11} = 0.8$ and $p_{01} = 0.2$. Such 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 of channel availabilities are relatively fast and the information exchanges channel between secondary users have 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 off-line converged performance, i.e., 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 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 6 show the performance comparisons among three distributed algorithms in the Ring topology (similar performance is observed under H-T topology). 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 the utilization between the real-time and convergence performance, followed by the greedy algorithm. The fair algorithm has the largest difference between the real-time and convergence 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 convergence performance is not very large, especially for the randomized and greedy algorithms. Under the time-varying channel availability, the randomized algorithm has the smallest difference between the realtime and convergence 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 availabilities.

VI. CONCLUSIONS

In this paper, we present a framework to illustrate the relationship between channel availability of secondary users and the usage of primary users. We pointed out two unique properties are inherent to systems with opportunistic spectrum availabilities and are independent to resource allocation schemes. They are location-dependency and time-variance.

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 availabilities. In particular, the greedy and the randomized algorithms are shown to remain effective under scenarios where users have only limited information exchange ability. Due to the time-variance of the channel availability, algorithms with low complexity and communication overhead are preferred.

REFERENCES

[1] M. McHenry, "Spectrum white space measurements," presented to New America Foundation BroadBand Forum, June 2003, http://www.newamerica.net/Download_Docs/pdfs/DocFile_185_1.pdf.

[2] FCC, “In the matter of promoting efficient use of spectrum through elimination of barriers to the development of secondary markets, WT Docket No. 00-230, FCC Rcd 20604.”

[3] ——, “Facilitating opportunities for flexible, efficient, and reliable spectrum use employing cognitive radio technologies, notice of proposed rule making and order, FCC 03-322.”

[4] ——, “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.”

[5] ——, “Promoting efficient use of spectrum through elimination of barriers to the development of secondary markets, WT Docket No. 00-230, report and order and further notice of proposed rulemaking, FCC 03-113 (secondary markets report and order and FNPRM).”

[6] DARPA, “Defense advanced research projects agency (DARPA) neXt Generation (XG) communications program,” http://www.darpa.mil/ato/programs/XG/.

[7] N. Garg, M. Papatriantafilou, and P. Tsigas, “Distributed long-lived list colouring: how to dynamically allocate frequencies in cellular networks,” *Wireless Networks*, vol. 8, no. 1, pp. 49–60, 2002.