Distributed Opportunistic Channel Acquisition Mechanism in Dynamic Spectrum Access Networks

S. Anand
Department of ECE
Stevens Institute of Technology
NJ 07030
Email: asanthan@stevens.edu

S. Sengupta
Department of Mathematics & Computer Science
John Jay College, City University of New York
NY 10019
Email: ssengupta@jjay.cuny.edu

R. Chandramouli
Department of ECE
Stevens Institute of Technology
NJ 07030
Email: mouli@stevens.edu

Abstract—We present a distributed opportunistic channel acquisition mechanism in dynamic spectrum access (DSA) networks. A novel graph-multicoloring algorithm is proposed for secondary networks, in which, each network makes use only of the local topology information (i.e., information about itself and its one-hop neighbors) to resolve contentions during channel access. We show that the proposed algorithm results in improvements by about one order of magnitude in the spectrum utilization for the secondary networks. We further present a modified probabilistic heuristic that improves the termination time of the algorithm by two orders of magnitude. The proposed algorithm results in Jain’s fairness index of about 0.9.

Index Terms — DSA Networks, Graph Coloring, Distributed Opportunistic Spectrum Access.

I. INTRODUCTION

The paradigm of dynamic spectrum access (DSA) [1] has provided new avenues for research due to the flexibility provided by allowing un-licensed users to use licensed spectrum bands. These un-licensed “secondary users” can opportunistically use the portions of the spectrum (called white spaces) that are unused by the licensed “primary users”. The network service providers that provide services to the secondary users are called secondary networks. In order to efficiently utilize the white spaces, the un-used spectrum should be re-used by as many secondary networks as possible. However, the set of secondary networks that re-use the same set of spectrum bands should satisfy the re-use constraints. These constraints are typically characterized by a re-use distance, $d_{\text{min}}$, indicating that users (or in some cases, networks) that re-use the same spectrum bands should be spaced by a distance of at least $d_{\text{min}}$ from each other.

Due to the scarcity of the available spectrum and the growing demands for spectrum from the users, it is essential to develop efficient mechanisms to allocate the spectrum to the secondary networks by maximizing the re-use and yet satisfy the re-use constraints. The spectrum allocation problem is typically modeled as a graph coloring problem [2] where each spectrum band represents a color, each network represents a vertex in the graph and two vertices are adjacent if the corresponding networks cannot re-use the same band. The graph coloring problem is known to be an NP-complete problem [2] and thus only heuristic solutions exist.

A broad survey on resource allocation in cellular networks though graph coloring mechanisms can be found in [3]-[6] and in the references therein. However, most of these works do not consider the dynamic availability of spectrum bands due to the presence of primary users and thus can not be directly applied to DSA networks. As far as dynamic opportunistic spectrum access and management are concerned, there is an emerging body of work that deal with different decision making aspects, issues and challenges in cognitive radio network setting. Peng et al [7] discuss graph coloring approaches to improve throughput for opportunistic spectrum access. Subramanian et al [8] presented spectrum allocation mechanisms using a Max K-cut approach for cellular dynamic spectrum access networks with a centralized controller. WillKomm et al [9] present graph coloring approaches to enable dynamic frequency hopping in cognitive radio networks. A survey of spectrum management in cognitive radio networks can be found in [10]. However, most of these aforementioned researches assume the presence of somewhat centrally controlled approaches.

Centralized graph coloring for spectrum allocation requires knowledge of the entire topology of the network, which cannot be obtained in practice in DSA networks due to the large number of secondary networks. Moreover, DSA networks typically do not have a centralized controller to manage the spectrum for all the secondary networks. It is noted that for each node, the re-use constraints affect only the node itself and its one-hop neighbors, i.e., the nodes that share an edge with it. Hence, if each node, knows information about its one-hop neighbors, then all the nodes can access the spectrum in a distributed manner and yet satisfy the re-use constraints. A more practical channel access mechanism would therefore be to obtain spectrum bands based on the local information, i.e., using a distributed graph coloring algorithm.

Studies on distributed graph coloring mostly propose heuristics that perform single coloring of graphs. In most such approaches, fairness is obtained by disallowing nodes that have obtained a color, to participate further in the contention for additional colors. This could lead to under-utilization of the spectrum in some scenarios. As an example, consider the graph shown in Fig. 1 (a), which contains four vertices and four edges. In this graph, note that the node $B$ can re-use the same color as nodes $C$ and $D$. However, according to the existing graph coloring heuristics node $B$ obtains the same color as
node C or that as node D but not both. In terms of spectrum management in DSA networks, this indicates that the network B would re-use the same set of spectrum bands as those used by network C or network D. However, by having a distributed algorithm to perform a graph multi-coloring, we can assign two colors to node B such that it shares the same color as nodes C and D. In the DSA context, this means that network B can now re-use those spectrum bands used by networks C and network D, thus improving the overall utilization of the spectrum. Therefore, it is essential to provide channel access mechanisms that are opportunistic with respect to not only the primary activity but also other secondary networks.

We use this as a motivation to provide a distributed algorithm for graph multi-coloring for efficient spectrum management in DSA networks. The proposed algorithm begins like most traditional approaches by assigning a color to the node of the highest degree. We however, present a novel mechanism that not only ensures that all nodes obtain at least one color, but also improves spectrum management by enabling certain nodes obtain multiple colors, opportunistic with respect to other nodes in addition to the primary, while yet satisfying the re-use constraints. We consider a system with N secondary networks. In order to account for all possible re-use constraints, we generate random Bernoulli graphs with a specified graph density and apply the proposed graph multi-coloring algorithm for the thus generated graphs. We show that the proposed distributed algorithm has an $O(N(N + e))$ complexity for a graph with N nodes and e edges. We compare the performance of the proposed distributed algorithm with that of traditional graph coloring algorithms and show that the proposed algorithm results in 20% to one order of magnitude of improvement in the total spectrum bandwidth obtained by the secondary networks.

We then introduce a modified algorithm where nodes request for colors according to a probabilistic heuristic as against a deterministic manner. To the best of our knowledge, there has been no previous work in the literature that has presented such heuristics to reduce the termination time of the graph coloring algorithm. The probabilistic heuristic reduces the probability of two neighboring nodes contending for the same color, thus resulting in all nodes obtaining colors faster. We show that such a heuristic can reduce the number of rounds of iterations and result in faster termination of the spectrum access procedure. The probabilistic heuristic is observed to provide 15% improvement (in terms of speed of termination) for sparse graphs and about two orders of magnitude of improvement for dense graphs.

The rest of the paper is organized as follows. We describe our algorithms and analysis in Section II. Numerical results are provided in Section III. Conclusions are drawn in Section IV.

II. DISTRIBUTED GRAPH COLORING ALGORITHM

Consider a system with N secondary networks represented as an N-node graph and $C_{\text{max}}$ available channels (or colors). The basic principle applied in graph-coloring heuristics is that the node with the highest degree (i.e., maximum number of one hop neighbors) should be assigned a color first since this reduces the conflict for colors for the other nodes thus resulting in faster allocation. We apply the same rule in the proposed algorithm. We first present the initial version of the algorithm (Section II-A) and then present an improvement to the initial algorithm using the novel probabilistic heuristic (Section II-B).

A. Initial Algorithm

Since the objective is to provide a distributed graph-coloring algorithm, it is essential that the different nodes in the graph (i.e., the different networks in the system) exchange information about each other, periodically. The information exchanged are

1) The node DEGREE (i.e., the number of one hop neighbors for each node). This information can be easily obtained from the neighbor list in infrastructure networks and from the beacons in ad hoc wireless networks. Since these messages are periodically broadcast in practical wireless systems, each node can easily obtain this information and update them periodically.

2) RANDOM-BACKOFF This is a randomly generated integer that helps nodes resolve the case when two neighboring nodes are of the same degree. In our heuristic, we consider the node that generates the smaller number to win the contention.

3) COLOR-OBTAINED. This information element allows each node to mention the color it has obtained to all its neighbors so that they can refrain from trying to acquire the same color (this is essential to satisfy the re-use constraints).

4) ALL-NEIGHBORS-DONE. This is a binary variable which takes the values ‘1’ if all neighbors of a node have obtained at least one color. This message enables the proposed distributed algorithm to perform a multi-coloring as will be seen in Algorithm 1.

5) ALL-NEIGHBORS-ISSUED-TERMINATE. This is a bit that a node broadcasts as a ‘1’ to indicate that all of its neighbors have broadcast an ALL-NEIGHBORS-DONE = 1 and ‘0’ to indicate otherwise. This information from all the nodes will be used as a termination condition for the proposed algorithm.

The distributed graph multi-coloring algorithm for a graph $G(V, E, C_{\text{max}})$ is described in detail in Algorithm 1 (see next page). Before proceeding any further, we present an illustrative example to explain how the proposed distributed graph multi-coloring algorithm works. Fig. 1 shows the sequence of color assignments to nodes when Steps 1-5 of Algorithm 1 are applied to the graph in Fig. 1(a) with $C_{\text{max}} = 9$. Since Node A has the largest degree, it obtains color 1 at the end of round 1, where a round denotes one complete set of operations from Steps 2) to 4(c)(i). In round 2, Node B obtains color 2 and among Nodes C and D, both of which have equal degree, one of them (say, Node C) obtains color 2 due to generating a smaller RANDOM-BACKOFF. At the end of round 2, the value of ALL-NEIGHBORS-DONE = 1 for Node B and that for Node A = 0. Therefore, Node B can request for an additional color (color 3) in round 3. Thus, at the end of round
Algorithm 1 Initial version of the algorithm to perform a distributed graph multi-coloring.

1) INITIALIZE
   a) REQUESTED-COLOR=1 for all vertices
   b) OBTAINED-COLOR=0 for all vertices
   c) ALL-NEIGHBORS-DONE[v]=0 for all vertices
   d) ALL-NEIGHBORS-ISSUED-TERMINATE[v]=0 for all vertices
   e) PAUSE[v]=0 for all vertices. If a vertex obtains a color then it shall not contend for more colors until all its neighbors obtain at least one color. PAUSE[v] enables enforcement of this.

2) DECIDE WINNER
   a) Among vertices such that PAUSE[v]=0 and those contending for the same color, the one with the highest degree is the winner.
   b) If two vertices have same degree, then the one with the smaller RANDOM-BACKOFF is the winner

3) ASSIGN COLOR: If vertex \( v \) is a winner
   a) Assign its requested color, i.e., OBTAINED-COLOR=Requested color for \( v \)
   b) Broadcast OBTAINED-COLOR to the neighbors
   c) PAUSE[v]=1. This prevents \( v \) from contending further for colors

4) CHECK IF ALL NEIGHBORS OBTAINED COLOR
   a) for all vertices \( v \)
      i) If OBTAINED-COLOR for all neighbors > 0, ALL-NEIGHBORS-DONE[v]=1
      ii) MAX=maximum of the OBTAINED-COLOR of itself and its neighbors
      iii) REQUESTED-COLOR[v]=MAX+1
   b) for all vertices \( v \)
      i) If ALL-NEIGHBORS-DONE[v]=1 and ALL-NEIGHBORS-ISSUED-TERMINATE[v]=0, PAUSE[v]=1. This step allows \( v \) to contend for another color after obtaining the first color. Essentially this is the condition that enables multi-coloring of the graph. This step implies that a vertex can contend for additional colors if all its neighbors obtain atleast one color and there exists some other vertex in the graph which has not obtained a color.

5) If ALL-NEIGHBORS-ISSUED-TERMINATE[v]=1 for all vertices, go to Step 6 else go to Step 2.
6) \( K'=\) the largest color assigned to any node in the graph.
7) for \( v \in V \)
   a) for \( i \in \) set of colors assigned to \( v \)
      i) Assign colors \( \{i+K',i+2K',\cdots,i+\lfloor C_{\text{max}}-1\rfloor K'\} \) to \( v \).
   b) end
8) end

Fig. 1. Color assignment sequence when Steps 1-5 of Algorithm 1 are applied to a sample graph.

Proof: The proof is by induction on the number of rounds. In the first round color 1 is assigned and the statement is trivially true. Let at the end of round \( n \), the maximum color assigned be \( c-1 \) and let the statement hold true. At the end of the \((n+1)^{th}\) round, The maximum color that any node can request is \( c \) from Steps 4(a)i)-4(a)iii). Thus color \( c \) is assigned before color \( c+1 \) and the statement is true.

Theorem 2.2: The network uses \( n \) colors at the end of the \( n^{th} \) round.

Proof: This is proved by induction on \( n \). During the first round, all nodes request for color 1. At least one node wins a color and thus the statement is trivially true. Let the network use \( m \) colors at the end of round \( m \). Therefore, according to Theorem 2.1 the biggest color that a node can request for during round \((m+1)\) is \( m+1 \). Thus some nodes request for color \( m+1 \) and others may request for a smaller color (i.e., color \( k < m+1 \)). In the \((m+1)^{th}\) round, at least one of the nodes that contend for color \( m+1 \) wins and obtains color \( m+1 \). The other nodes that win contending for color \( k < m+1 \) obtain color \( k < m+1 \). Thus, at the end of round \( m+1 \), \( m+1 \) colors are used by the network.

Theorem 2.2 results in the following corollary and theorem:

Corollary 2.1: Let Algorithm 1 terminate at the end of \( K_{\text{min}} \) rounds. Then, \( K_{\text{min}} = K' \), where \( K' \) is as specified in Step 6.

Theorem 2.3: The Algorithm 1 terminates in at most \( N \) rounds for an \( N \) node network.

Proof: The chromatic number of an \( N \) node network is at most \( N \). From Theorem 2.2, the number of colors used at the end of round \( N \) is \( N \). Thus at the end of round \( N \), \( N \) colors are used, i.e., all nodes obtain at least one color and thus Algorithm 1 terminates.

Theorem 2.3 shows that it takes at most \( N \) rounds for all nodes in an \( N \)-node network to obtain a color irrespective of the topology. This is used in the following theorem to determine the complexity of Algorithm 1.

Theorem 2.4: The complexity of Algorithm 1 is \( O(N(N+e)) \), where \( e \) is the number of edges in the graph.

Proof: Steps 2) and 4) are \( O(e) \), and Steps 3) and 5) are \( O(N) \). Therefore, one sequence of operations from Steps 2-4(e)ii) is \( O(N+e) \). From Theorem 2.3, at most \( N \) rounds are required for termination. Hence, Steps 2-6 is \( O(N(N+e)) \).
Steps 7 and 8 are $O(NK_{\text{min}}) = O(N^2)$. Thus Algorithm 1 is $O(N(N + \epsilon))$.

If the primary returns in any of the channels obtained by any of the secondary networks, then the secondary network cannot use the corresponding channel for the duration which the primary occupies the channel.

**B. Probabilistic Heuristic**

Consider a scenario of a complete graph (a graph in which all pairs of nodes have an edge between them). In this case, Algorithm 1 will take $N$ rounds to terminate because, in each round, only one node can win the contention (since all nodes are neighbors of each other and with equal degree). The number of rounds for termination is very close to $N$ for dense graphs (graphs in which most pairs of nodes have an edge between them). This happens because many nodes contend for the same color. If neighboring nodes contend for different colors, then they can all obtain their requested color in the same round, thus leading to faster termination of the channel access procedure.

We now present a means to perform the distributed graph coloring in fewer rounds by making nodes request for colors in a probabilistic manner. In order to achieve this, each node considers the subgraph consisting of itself and its neighbors and computes a rank for each node in the subgraph (including itself), by sorting the nodes in the non-increasing order of the degrees. Thus, the subgraph considered by node $k$ of degree $M - 1$ consists of $N$ nodes. The node with the highest degree in the subgraph obtains rank 1, the node with the next highest degree obtains rank 2, and so on. Let the rank of node $k$ be $\alpha$ and let the smallest unused color be $\omega$ $(1 \leq \omega \leq \chi$, where $\chi$ is the total number of available colors). If $\alpha = 1$, then node $k$ requests for color $\omega$, else, node $k$ requests for color $\omega + \delta$, with probability $p_{\delta} \geq 0$ such that $\sum_{\delta=0}^{\infty} p_{\delta} = 1$.

We present a scenario in which the probability, $p_{\delta}$, is computed according to a truncated geometric distribution. Let $0 < p, \rho < 1$ and let the subgraph consisting of node $k$ and its neighbors have $M$ nodes. Let the rank of node $k$ be $\alpha$. When $\alpha > 1$, node $k$ requests for color $\omega + \alpha - 1$ with probability, $p$ and color $\omega + \delta$ with probability

$$p_{\delta} = \begin{cases} \rho^{|\delta - \alpha|} & |\delta - \alpha| \leq M - 1 \\ 0 & \text{otherwise} \end{cases}.$$  

Using the fact that $\sum_{\delta=0}^{\infty} p_{\delta} = 1$ in (1), $p$ can be obtained as

$$p = \frac{1 - \rho}{1 + \rho - \rho^\alpha - \rho^{M-\alpha+1}}.$$  

**Theorem 2.5:** The expression for $p$ in (2) satisfies $0 < p < 1$ for $\rho < 1$.

**Proof:** From (2), it is obvious that for $0 < \rho < 1$, $p > 0$. For $p < 1$, $\rho^{\alpha - 1} + \rho^{M - \alpha} < 2 \Rightarrow 0 < 2 - \rho^\alpha - \rho^{M-\alpha+1} \Rightarrow 1 - \rho < 1 + \rho - \rho^\alpha - \rho^{M-\alpha+1} \Rightarrow p = \frac{1-p}{1+\rho^\alpha - \rho^{M-\alpha+1}} < 1$. 

III. RESULTS AND DISCUSSION

In order to perform the numerical computations, we generate random graphs with specified number of nodes and graph density. The graph density, $d$, is given by $d = 2e/N(N-1)$ for a graph with $N$ nodes and $e$ edges. Note that $0 \leq d \leq 1$ characterizes how dense or sparse a graph is. When $d \to 0$, $e \to 0$ and the graph is sparse. Similarly $d \to 1$ represents a graph with $e \to (\binom{N}{2})$, i.e., a complete graph. Thus values of $d$ closer to 1 represent dense graphs. We perform the distributed graph coloring on the randomly generated graphs using the initial algorithm (Section II-A) and the probabilistic heuristic (Section II-B) with $C_{\text{max}} = 300$ channels. Fig. 2(a) presents the minimum number of colors used to color the graph, $K_{\text{min}}$ (which is as specified in Corollary 2.1), when using a centralized algorithm and when using Algorithm 1. As observed, the proposed distributed algorithm and the centralized graph coloring algorithm yield the same value of $K_{\text{min}}$. Thus, the proposed algorithm uses only local information and yet provides the same performance as a centralized algorithm in terms of the minimum number of required colors.

Fig. 2(b) presents the total number of spectrum bands (colors) obtained by each network on an average, when Algorithm 1 terminates. It is observed that for traditional graph coloring approaches, each network obtains one spectrum band. However, the proposed initial algorithm results in multiple spectrum bands to nodes even for dense graphs (graph density=0.9). For sparse graphs (graph density=0.1), there are fewer edges enabling more re-use. Thus, there are two orders
of enhancement in the total spectrum obtained per network. Note that these improvements are obtained while still using the same number of colors (i.e., the same amount of overall bandwidth) as a centralized approach.

Typically, a band consists of multiple sub-carriers. If each band has a bandwidth of 1 MHz, then traditional graph single-coloring approaches provide 1 band per node resulting in 1 MHz of bandwidth per node. For a spectral efficiency of 3 b/s/Hz, this results in a maximum data rate of 3 Mbps in each network. The proposed algorithm is observed to provide about 1.2 (for dense graphs) to 20 (corresponding to sparse graphs) colors per node. This leads to a corresponding increase in the data throughput in the network and hence, the system capacity (i.e., by a factor 1.2 for dense graphs and 20 for sparse graphs). In other words, the proposed initial algorithm can provide 20% to two orders of magnitude of improvement to the system capacity.

The fairness of the proposed algorithm is measured in terms of the Jain’s fairness index [12]. Typically, systems with a fairness index larger than 0.5 are considered to provide fairness. The results depicted by the fairness index presented in Fig. 3(a), indicate great deal of fairness among the different networks in terms of the amount of spectrum obtained. The fairness index for the modified algorithm is presented in Fig. 3(b). It is observed that the fairness index is larger (closer to 1) for dense graphs and is lesser for sparse graphs. This is because, for dense graphs almost all nodes obtain only a single color. However, for sparse graphs, some nodes obtain more colors than the others depending on the topology. The fairness index is higher in some cases for the initial algorithm and higher in some cases for the probabilistic heuristic.

Figs. 4(a) and 4(b) present the rate of termination (in terms of number of rounds required to terminate), for the initial algorithm and the probabilistic heuristic with respect to the number of nodes and the graph density, respectively. From Fig. 4(a), it is observed that the probabilistic heuristic can improve the rate of termination by two orders of magnitude. The improvement is larger when the number of nodes increases. This is because, according to Theorem 2.3, the initial algorithm requires $O(N)$ rounds to terminate. The probabilistic heuristic terminates much faster because, all nodes request for different colors in the first round and thus most nodes obtain their required color in the first round itself. By a similar argument, improvements are more significant for dense graphs than sparse graphs, as observed from Fig. 4(b).

The significance of the improvement in the rate of termination is explained as follows. Typically, messages and information in wireless networks are broadcast in different frames or sub-frames. A larger value of the rate of termination represents a correspondingly larger value in the number of sub-frames required for control signaling and hence, correspondingly larger value of the initial access delay. By providing an improvement by two orders of magnitude in the rate of termination, the probabilistic heuristic also provides an equivalent improvement in the initial access delay.

![Graph showing rate of termination vs number of networks](image)

![Graph showing rate of termination vs graph density](image)

**Fig. 4.** Rate of termination. The legend “Modified Algorithm” indicates the performance of the probabilistic heuristic.

**IV. CONCLUSION**

We presented an opportunistic distributed graph coloring algorithm for spectrum access in DSA networks. We presented an initial multi-coloring algorithm that resulted in 20% to about an order of magnitude improvement in the capacities of the networks. We then presented a modification to the initial algorithm where nodes request for colors in a probabilistic manner, with the probabilities dependent on the degrees of nodes and their neighbors. The probabilistic heuristic was found to provide two orders of improvement in reducing the termination time with respect to the initial algorithm. This corresponds to two orders of reduction in the initial access delay.

**REFERENCES**


