MASPECT: A Distributed Opportunistic Channel Acquisition Mechanism in Dynamic Spectrum Access Networks

S. Anand*, S. Sengupta† and R. Chandramouli*
*Department of ECE, Stevens Institute of Technology, New Jersey
† Department of Math. & Comp. Sc., John Jay College, City University of New York
asanthan@stevens.edu, ssengupta@jjay.cuny.edu, mouli@stevens.edu

Abstract

We present a distributed opportunistic channel acquisition mechanism in dynamic spectrum access (DSA) networks. A novel graph MAximum SPECTrum packing algorithm (MASPECT), is proposed for a system with N secondary networks, in which, each secondary 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. The proposed algorithm also adapts easily to topological changes. Most approaches to channel acquisition either use centralized graph coloring approaches or use distributed approaches that result in under utilization of spectrum resources. We show that the proposed MASPECT algorithm results in improvements by upto one order of magnitude in the spectrum utilization for the secondary networks and results in Jain's fairness index of about 0.9. We further present a *modified* probabilistic heuristic, PMASPECT, that improves the termination time of the algorithm by upto two orders of magnitude. *To the best of our knowledge this is the first heuristic proposed to reduce the termination time, that makes use of the relative degrees of a node and its one hop neighbors.* We also study the impact of the proposed MASPECT algorithm on each individual secondary network in the system in the presence of primary activity. For each secondary network, the proposed algorithm is observed to result in upto two orders of magnitude of reduction in the new call blocking and in the call dropping probabilities in the presence of primary activity.

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. Secondary users sense the spectrum for availability of spectrum holes and follow specific protocols in accessing the spectrum (e.g., [2]). 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_{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_{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 [3] 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 approach to spectrum assignment can be applied to adhoc wireless networks [4] as well as to networks with infrastructure [5] (e.g., cellular networks). The graph coloring problem is known to be an NP-complete problem [3] and thus only heuristic solutions exist.

A broad survey on resource allocation in cellular networks though graph coloring mechanisms can be found in [5]-[10] 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* [11] discuss graph coloring approaches to improve throughput for opportunistic spectrum access. Subramanian *et al* [12] presented spectrum allocation mechanisms using a Max K-cut approach for cellular dynamic spectrum access networks with a centralized controller. Centralized approaches to graph channel assignment in cognitive radio networks were also presented in [13]. WillKomm *et al* [14] present graph coloring approaches to enable dynamic frequency hopping in cognitive radio networks. Fan and Zhang [15] present an interference graph coloring approach for spectrum access in DSA networks. A survey of spectrum management in cognitive radio networks can be found in [16]. However, most of these 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.

A distributed graph edge coloring heuristic was proposed in [17]. More detailed descriptions of centralized and distributed graph coloring approaches can be found in [18]. 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, 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 ***maximum spectrum packing***, 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 coloring that provides multiple colors to certain nodes without violating the re-use constraints. We call this the graph *maximum spectrum packing* (MASPECT) algorithm. The proposed MASPECT algorithm results in more efficient spectrum management in DSA networks as it provides better utilization of spectrum resources. The MASPECT 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 [19] with a specified graph density and apply the proposed graph maximum spectrum packing 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 both at a system level (i.e., including all the secondary networks in the system) and at a network level (i.e., the benefit provided by the proposed algorithm to an individual secondary network). The proposed algorithm is observed to result in 20% to one order of magnitude of improvement in the total spectrum bandwidth obtained by the secondary networks. For an individual secondary network, we compare the performance in terms of the new call blocking probability and call dropping probability in the network in the presence of primary activity. The proposed algorithm results in a reduction by two orders of magnitude in the new call blocking probability in the call dropping probability, compared to traditional graph coloring approaches.

We then introduce a ***modified heuristic***, PMASPECT, where nodes request for colors according to a *probabilistic heuristic* as against a deterministic manner. In [20], a scheme was proposed according to which, nodes request for any one of n available colors with equal probability. While this scheme does reduce the termination time, it does not exploit the relative degrees of the nodes in the graph. We propose a heuristic according to which, nodes request for different colors with a probability that depends on the relative degree of the node and its one-hop neighbors. *To the best of our knowledge, there has been no previous* *work in the literature that has presented heuristics that utilize the relative degrees of the nodes, to reduce the termination time of the graph coloring algorithm.* The proposed probabilistic heuristic, PMASPECT, 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 PMASPECT 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 compared to the MASPECT algorithm.

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 Opportunistic Spectrum Access

Consider a system with N secondary networks represented as an $N-$node graph and $C_{\max}$ 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 begin by applying the same rule in the proposed algorithm. We first present the initial version of the algorithm, MASPECT, (Section II-A) and then present an improvement to the MASPECT algorithm using the novel probabilistic heuristic called PMASPECT (Section II-B).

A. MASPECT Algorithm

The MASPECT is a distributed algorithm that enables the nodes in the graph acquire a color. It is therefore essential that the different nodes in the graph (i. e., the different networks in the system) exchange information about each other, periodically, on a dedicated control channel. The information exchanged are listed below.

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 maximum spectrum packing 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.

It is noted that the set of information listed above are exchanged only between each node at its adjacent nodes (i.e., one-hop neighbors). Hence only local information is used by all the nodes in the distributed graph maximum spectrum packing (MASPECT) algorithm, (described in detail in Algorithm 1), for a graph $G(V, E, C_{\max})$.

Before proceeding any further, we present an illustrative example to explain how the proposed MASPECT algorithm works. Fig. 2 shows the sequence of color assignments to nodes when Steps 1-5 of Algorithm 1 are applied to the graph in Fig. 2(a) with $C_{\max} = 9$. 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 3, Nodes B and D obtain color 3. *Node B therefore obtains multiple channels (colors) by opportunistically exploiting the local topology*. At Step 6 of the MASPECT algorithm, $K' = 3$ and hence, upon executing Steps 7 and 8 with $C_{\max} = 9$, the set of colors obtained by Node A, $\chi_A = \{1, 4, 7\}$. Similarly, $\chi_C = \{2, 5, 8\}$, $\chi_D = \{3, 6, 9\}$ and $\chi_B = \{2, 3, 5, 6, 8, 9\}$.

We now describe how our algorithm adapts to changes in the topology (Section II-A1) and then proceed to perform the complexity analysis of Algorithm 1 in Section II-A2. We end this subsection with the discussion on how the proposed MASPECT algorithm can impact the performance of an individual secondary network (which represents an individual node in the graph) in the presence of primary activity (Section II-A3).

1) Adaptation to topological change: The MASPECT algorithm also adapts itself easily to topological changes in the system. As an example, if a new node is added to the graph (corresponding to a new secondary network entering the system), then the new node can query its one-hop neighbors to obtain the list of colors obtained by them at the end of Step 5 in Algorithm 1. The new node then acquires all the missing colors. For example, consider the graph shown in Fig. 3(a). Nodes A, C and D are present in the system initially. After executing Steps 1-5 in Algorithm 1, nodes A, C and D obtain colors 1,

Algorithm 1 The MASPECT algorithm that performs a distributed graph maximum spectrum packing.

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-DONE=1 for all neighbors of v, ALL-NEIGHBORS-ISSUED-TERMINATE$[v] = 1$
   c) for all vertices v
      i) If ALL-NEIGHBORS-DONE$[v] = 1$ and ALL-NEIGHBORS-ISSUED-TERMINATE$[v] = 0$, PAUSE$[v] = 0$. This step allows v to contend for another color after obtaining the first color. Essentially this is the condition that enables maximum spectrum packing 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=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 $\left\{i + K', i + 2K', \cdots, i + \lfloor \frac{C_{\max}-i}{K'} \rfloor K'\right\}$ to v.
   b) end
8) end

2 and 3, respectively as shown in Fig. 3(b)-3(d). Let $C_{\max} = 9$. Then at the end of Step 7 in Algorithm 1, $\chi_A = \{1, 4, 7\}$, $\chi_C = \{2, 5, 8\}$ and $\chi_D = \{3, 6, 9\}$. Let node B enter the system and let A be its only one-hop neighbor as shown in Fig. 3(e). Node B queries node A for the list of the colors it obtained at the end of Step 5 and the value of K'. From this information, node B computes χ_A. Since node B knows that $C_{\max} = 9$, it knows that the set of available colors, $\mathcal{S} = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$ and hence, acquires the colors $\chi_B = \mathcal{S} \backslash \chi_A = \{2, 3, 5, 6, 8, 9\}$. This is the same set of colors obtained by node B if Algorithm 1 is completely executed from Steps 1-8 for the graph shown in Fig. 3(e). Note however, that this mechanism may result in inefficient color allocations for some topologies due to the inherent NP-completeness of the graph coloring problem [3].

When an existing node leaves the system (corresponding to a secondary network leaving the market), then the one-hop neighbors of that node execute Algorithm 1. However, at each round, nodes only request for those colors that are not acquired by their one-hop neighbors. As an example, consider the graph shown in Fig. 2(d). It was shown earlier that for $C_{\max} = 9$, $\chi_A = \{1, 4, 7\}$, $\chi_C = \{2, 5, 8\}$, $\chi_D = \{3, 6, 9\}$ and $\chi_B = \{2, 3, 5, 6, 8, 9\}$. If node D leaves the system, then nodes A and C execute Algorithm 1 with color 3 being the only available color. Node A knows that node B has acquired color 3 and hence, will not contend for color 3. Thus, node C obtains color 3 and according to Step 7, also obtains colors 6 and 9.

2) Complexity analysis: We now present the performance analysis of Algorithm 1. In order to determine the complexity of the algorithm, it is essential to determine the number of rounds taken for the algorithm to terminate. This number, in turn, depends on the sequence in which colors are assigned to different nodes in the graph. Theorem 2.1 below, provides insight into the sequence in which colors are assigned to nodes.

Theorem 2.1: **In Algorithm 1, a color c is assigned at any round only after all colors 1 through $c-1$ are assigned in the previous rounds.**

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 2.1 indicates that the algorithm does not assign a "bigger" color before assigning all "smaller" colors. This is utilized in the following theorem to provide the maximum number of colors used at the end of each round.

Theorem 2.2: **The graph 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 graph.

■

Theorem 2.2 results in the following corollary and theorem which yields the maximum number of rounds required for the algorithm to terminate.

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

Theorem 2.3: **The Algorithm 1 terminates in at most N rounds for an N node graph.**

Proof: The chromatic number of an N node graph is at most N. From Theorem 2.2, N colors are used at the end of round 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 graph 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(c)i) 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 $O(NK_{\min}) = O(N^2)$. Thus Algorithm 1 is $O(N(N+e))$.

■

3) Impact on each secondary network: The MASPECT algorithm also impacts the performance of each individual secondary network (i.e., each node in the graph). As an example, let the traditional graph coloring approaches enable a network acquire $n_{\text{traditional}}$ number of channels (i.e., colors) and let the MASPECT algorithm yield n_{proposed} number of channels to the same network. Users in the network use one of the acquired spectrum bands (channels) for each communication (also called a "call"). Newly arriving calls that do not find a channel are "blocked". Similarly, if the primary returns in any of the channels on which there is an "ongoing" call by a user belonging to the secondary network of interest, then the user cannot use the corresponding channel for the duration which the primary occupies the channel. The ongoing call incident on the user then has to be switched to another channel acquired by the secondary network, which is neither used by the primary nor used by other secondary calls. If such a channel is not available, the call is "dropped". Depending on the values of $n_{\text{traditional}}$ and n_{proposed}, the performance of the secondary network differs when deploying traditional channel acquisition mechanisms and when deploying the proposed MASPECT algorithm. The performance of the secondary network is measured in terms of the fraction of newly arriving calls that are blocked (termed as new call *blocking probability*) and the fraction of ongoing calls that are dropped (termed as call *dropping probability*).

In order to analyze the performance of the network, we make the following assumptions

- The secondary network of interest acquires n spectrum bands or channels. When deploying the proposed algorithm, $n = n_{\text{proposed}}$ and when deploying traditional graph coloring approaches, $n = n_{\text{traditional}}$.
- Secondary calls arrive according to a Poisson process with arrival rate, λ_s.
- The call holding times of secondary calls are exponentially distributed with mean holding time, $\frac{1}{\mu_s}$.
- Primary arrives on any channel according to a Poisson process with arrival rate, λ_p.
- The primary holds the channel for a time period which is exponentially distributed with mean holding time, $\frac{1}{\mu_p}$.
- Each secondary call uses one channel for communication. No two secondary calls in the same network can use the same channel.
- Primary can arrive only on those channels which are unused or on those on which there is an ongoing secondary call. No two primary calls can use the same channel.

Let (s, p) be the two-tuple that represents the number of channels occupied by secondary and primary calls, respectively, in a secondary network with n channels. Based on the assumptions listed above, the two-tuple (s, p) can be modeled as a two-dimensional continuous time Markov chain (CTMC) with transition rates as shown in Fig. 4.

The transitions in Fig. 4 are explained as follows.

- Transition from state (s, p) to $(s + 1, p)$ occurs when a new secondary call arrives in the system. This transition takes place with rate, λ_s (the arrival rate of secondary calls).
- Transition from state (s, p) to $(s, p + 1)$ corresponds to a new primary call arrival. Since there are p active primary calls in the system, the new call can arrive on any of the remaining $n - p$ channels, at a rate, $(n - p)\lambda_p$.
- Transition from state (s, p) to $(s - 1, p)$ corresponds to completion of any one of the s secondary calls and occurs with rate $s\mu_s$. Similarly transition from state (s, p) to $(s, p - 1)$ corresponds to the completion of any one of the p primary calls and occurs with rate, $p\mu_p$.
- When the system is "full" i.e., in state (s, p) such that $p < n$ and $s + p = n$, then a primary call arriving at rate, $(n - p)\lambda_p$ causes one of the s secondary calls to evacuate the channel. The secondary call will not be able to find an alternate channel because, all the channels are occupied by other secondary calls or primary calls. Therefore, the call will be dropped. This corresponds to the transition from state (s, p) to $(s - 1, p + 1)$ with rate, $(n - p)\lambda_p$.

Let $\pi(s, p)$ be the steady state probability of there being s channels used by secondary calls and p channels being used by primary calls. This can be evaluated numerically for the CTMC shown in Fig. 4. However, we simplify the analysis by approximating the CTMC in Fig. 4 by one shown in Fig. 5, i.e., neglecting the transitions from states (s, p) (with $p < n$ and

$s+p=n$) to $(s-1,p+1)$. The CTMC in Fig. 5 now corresponds to a two-dimensional birth-death process for which the steady state probability of the channel occupancy (s,p), $\pi(s,p)$, can be written as [21]

$$\pi(s,p) = \begin{cases} \frac{1}{G}\frac{\rho_s^s}{s!}\binom{n}{p}\rho_p^p & s+p \leq n \\ 0 & \text{otherwise,} \end{cases} \tag{1}$$

where $\rho_s = \frac{\lambda_s}{\mu_s}$, $\rho_p = \frac{\lambda_p}{\mu_p}$ and

$$G = \sum_{s+p\leq n} \frac{\rho_s^s}{s!}\binom{n}{p}\rho_p^p. \tag{2}$$

Consider the network with the channel occupancy in state, (s,p), such that $s+p=n$. In this state, if a newly arriving call is a secondary call, then it is blocked. Similarly, if the newly arriving call is a primary call, then it causes an ongoing secondary call to be dropped. Therefore, the new call blocking probability, p_{block}, and the dropping probability, p_{drop}, can be written as

$$p_{\text{block}} = \sum_{s+p=n} \frac{\lambda_s}{\lambda_s + (n-p)\lambda_p}\pi(s,p) \tag{3}$$

$$\text{and } p_{\text{drop}} = \sum_{\substack{s+p=n \\ p<n}} \frac{(n-p)\lambda_p}{(n-p)\lambda_p + \lambda_s}\pi(s,p). \tag{4}$$

The expression for $\pi(s,p)$ in (1) is used in (3) and (4). In (1)-(4), $n = n_{\text{proposed}}$ when deploying the proposed graph coloring approach and $n = n_{\text{traditional}}$ when deploying traditional approaches. Since the values of $n_{\text{traditional}}$ and n_{proposed} are expected to be different, the corresponding values of the new call blocking probability and call dropping probability are also expected to be different when deploying traditional approaches and when deploying the proposed MASPECT algorithm.

In the following subsection, we present a modified version of the MASPECT algorithm (called PMASPECT), which provides a reduction in the termination time.

B. PMASPECT: 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 M 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 α and let the smallest unused color be ω $(1 \leq \omega \ \leq \chi$, where χ is the total number of available colors). If $\alpha = 1$, then node k requests for color ω, else, node k requests for color δ with probability $p_\delta \geq 0$ such that $\sum_{\delta=1}^{\chi} p_\delta = 1$.

We present a scenario in which the probability, p_δ, 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 α. When $\alpha > 1$, node k requests for color $\omega + \alpha - 1$ with probability, p and color $\omega + \delta$ with probability

$$p_\delta = \begin{cases} p\rho^{|\delta-\alpha|} & |\delta - \alpha| \leq M-1 \\ 0 & \text{otherwise.} \end{cases} \tag{5}$$

Using the fact that $\sum_\delta p_\delta = 1$ in (5), p can be obtained as

$$p = \frac{1-\rho}{1+\rho-\rho^\alpha-\rho^{M-\alpha+1}}. \tag{6}$$

Since p is a probability, it should satisfy $0 < p < 1$. The following theorem establishes conditions on ρ that yields $0 < p < 1$.

Theorem 2.5: **The expression for p in (6) satisfies $0 < p < 1$ for $0 < \rho < 1$.**

Proof: From (6), it is obvious that for $0 < \rho < 1$, $p > 0$. For $\rho < 1$,

$$\begin{aligned} \rho^{\alpha-1} + \rho^{M-\alpha} &< 2 \\ \Rightarrow 0 &< 2\rho - \rho^\alpha - \rho^{M-\alpha+1} \\ \Rightarrow 1-\rho &< 1+\rho-\rho^\alpha-\rho^{M-\alpha+1} \\ \Rightarrow p = \frac{1-\rho}{1+\rho-\rho^\alpha-\rho^{M-\alpha+1}} &< 1. \end{aligned} \tag{7}$$

■

III. Results and Discussion

We present the numerical results in two stages. First, we compare the performance of the proposed MASPECT algorithm and PMASPECT heuristic with that of traditional approaches in terms of the performance of the whole system with all the secondary networks (Section III-A). We demonstrate the effectiveness of the proposed algorithm in terms of efficient spectrum management. We then compare the network level performance, i.e., the performance of each individual secondary network, when deploying the the proposed MASPECT algorithm (Section III-B). Specifically, we compare the performance of the secondary networks in terms of the call blocking and call dropping probabilities when deploying the proposed algorithm and when deploying traditional spectrum acquisition approaches.

A. System Level Performance

In order to perform the numerical computations, we generate random graphs on a C-based simulator on UBUNTU LINUX platform as in [13], 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 $C_{\max} = 300$ channels. Fig.6(a) presents the minimum number of colors used to color the graph, $K_{\min}$ (which is as specified in Corollary 2.1), when using a centralized algorithm and when using the MASPECT algorithm (Algorithm 1). As observed, the proposed distributed MASPECT algorithm and the centralized graph coloring algorithm yield the same value of $K_{\min}$. Thus, the proposed MASPECT 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. 6(b) presents the total number of spectrum bands (colors) obtained by each network on an average, when the MASPECT algorithm terminates. It is observed that for traditional graph coloring approaches, each network obtains one spectrum band. However, the proposed MASPECT 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 is more than one order of magnitude 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 MASPECT algorithm can provide 20% to more than one order of magnitude of improvement to the system capacity.

The fairness of the proposed algorithm is measured in terms of the Jain's fairness index [22]. 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. 7(a), indicate great deal of fairness among the different networks in terms of the amount of spectrum obtained. The fairness index for the probabilistic heuristic, PMASPECT (described in Section II-B), is presented in Fig. 7(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 MASPECT algorithm and higher in some cases for the probabilistic heuristic, PMASPECT.

Figs. 8(a) and 8(b) present the rate of termination (in terms of number of rounds required to terminate), for the MASPECT algorithm and the probabilistic heuristic, PMASPECT, with respect to the number of nodes and the graph density, respectively. From Fig. 8(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 MASPECT algorithm requires $O(N)$ rounds to terminate. The PMASPECT 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. 8(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 PMSAPECT heuristic also provides an equivalent improvement in the initial access delay.

B. Network Level Performance

As mentioned in Section II-A3, the increase in the spectrum utilization shown in Fig. 6(b) in Section III-A impacts the performance of each secondary network, in the presence of primary activity. We now compare the performance of the proposed MASPECT algorithm with that of traditional channel acquisition mechanisms on each secondary network. We consider a particular secondary network in a system with 100 secondary networks. The number of channels in the network, n is taken to be $n_{\text{traditional}}$ and n_{proposed} when deploying traditional channel acquisition mechanisms and when deploying the proposed MASPECT algorithm, respectively. The values of $n_{\text{traditional}}$ and n_{proposed} are obtained from Fig. 6(b).

The metrics of interest are the new call blocking probability and the call dropping probability. The parameters for the analysis are taken to be as in [23] and listed in Table I. The number of users in the secondary network are varied to yield different values of λ_s and hence, different values of ρ_s and thus, different values of p_{block} and p_{drop}..

Fig. 9 depicts the performance a secondary network in terms of the new call blocking probability (Fig. 9(a)) and the call dropping probability (Fig. 9(b)), when deploying the proposed MASPECT algorithm and when deploying traditional channel

TABLE I
VALUES OF PARAMETERS USED IN THE ANALYSIS AND SIMULATIONS.

Parameter	Value
λ_p	1/hour
μ_p	1/24 second
λ_s	6/hour/user
μ_s	1/36 second

acquisition approaches. Results are shown for graph density (GD)$= 0.9$ and GD$= 0.5$. The new call blocking probability and the call dropping probability were found to be zero in graphs with density, GD$= 0.1$. The legends marked "analysis" in Figs. 9(a) and 9(b) represent the analytical results obtained by applying the analysis described in Section II-A3. The legends marked "simulation" represent results obtained by running C-based simulations run on LINUX platform as in [23]. It is observed that the analytical results and simulation results in Figs. 9(a) and 9(b) follow each other closely.

Fig. 9(a) demonstrates the effectiveness of the MASPECT algorithm in reducing the new call blocking probability. As an example, for 200 users, traditional channel acquisition mechanism results in a blocking probability of 0.3. The MASPECT algorithm results in a blocking probability of 0.1 in graphs with density 0.9 and 0.001 in graphs with density 0.5. This is a reduction of 70% to two orders of magnitude in the new call blocking probability. The *network capacity* is defined as the maximum number of users that can be supported by the network that results in a blocking probability less than a specified threshold. For a typical threshold of 2% [24], the traditional channel acquisition mechanism supports about 55 users. The proposed MASPECT algorithm supports about 85 users in graphs with density 0.9 and more than 200 users in a graph with density 0.5. Thus the proposed MASPECT algorithm can improve the network capacity by about 89% to more than one order of magnitude.

Fig. 9(b) presents the performance of the secondary network in terms of the call dropping probability. It is observed that the MASPECT algorithm is also very effective in reducing the call dropping probability. In a network with 200 users, traditional channel acquisition mechanisms result in a dropping probability of 0.003. The corresponding values for the MASPECT algorithm are 0.002 and 3^{-5} for graph densities of 0.9 and 0.5, respectively. This is a reduction by 33% to two orders of magnitude. If the network capacity is defined as the maximum number of users that results in less than a desired dropping probability (typically 0.1% [24]), then traditional channel acquisition mechanisms result in a capacity of 45 users while the MASPECT algorithm yields a capacity of 120 users for graphs with density 0.9 and more than 200 users for graphs with density 0.5. In other words, the MASPECT algorithm improves the capacity by one to two orders of magnitude.

IV. Conclusion

We presented an opportunistic distributed graph coloring algorithm for spectrum access in DSA networks called MASPECT that resulted in 20% to about an order of magnitude improvement in the spectrum utilization. We then presented a modification to the MASPECT which is a probabilistic heuristic called PMASPECT where nodes request for colors in a probabilistic manner, with the probabilities dependent on the degrees of nodes and their neighbors. The PMASPECT 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. We also demonstrated the effectiveness of the MASPECT algorithm on the performance of each secondary network in the system in the presence of primary activity. The MASPECT algorithm provided 70% to two orders of magnitude reduction in the new call blocking probability and 33% to two orders of magnitude reduction in the call dropping probability.

V. Acknowledgement

This research was partially funded by NSF CNS #0917008, NSF CCF #0916180 and by # 2009-92667-NJ-IJ

References

[1] S. Mangold and K. Challapali, "Coexistence of wireless networks in unlicensed frequency bands," *Wireless World Research Forum ♯ 9*, Jul. 2003.
[2] T. Chen, H. Zhang, X. Zhou, G. M. Maggio, and I. Chlamtac, "CogMesh: A cluster based cognitive radio mesh network," *Book chapter: Cognitive Wireless Networks, Springer*, p. 657=678, 2007.
[3] T. Cormen, C. Leisserson, R. Rivest, and C. Stein, *Introduction to Algorithms*. McGraw Hill, 2001.
[4] M. Nekovee and R. S. Saksena, "Simulation of large scale WiFi based wireless networks: Interdisciplinary challenges and applications," *Elsevier Jl. on Future Generation Computer Systems*, vol. 26, no. 3, pp. 514–520, Mar. 2010.
[5] A. Gamst, "Some lower bounds for a class of frequency assignment problems," *IEEE Trans. on Vehic. Technol.*, vol. 35, no. 1, pp. 8–14, Jan. 1986.
[6] S. Khanna and K. Kumaran, "On wireless spectrum estimation and generalized graph coloring," *Proc., IEEE INFOCOM'98*, Mar. 1998.
[7] A. Koster, "Frequency assignment: Models and algorithms," Ph.D. dissertation, Universiteit Maastricht, Maastricht, The Netherlands, 1999.
[8] L. Narayanan, "Channel assignment and graph multicoloring," *Handbook of wireless networks and mobile computing*, pp. 71–94, 2002.
[9] D. Youngs, "Frequency assignment for cellular radio networks," pp. 179 –183, 1995.
[10] A. Safwat, "Distributed connection admission control and dynamic channel allocation in ad hoc-cellular networks," *Intl. Conf. on Digital Telecommun. (ICDT'06)*, Aug. 2006.
[11] C. Peng, H. Zheng, and B. Y. Zhao, "Utilization and fairness in spectrum management for opportunistic spectrum access," *Mobile Network Applications*, vol. 11, pp. 555–576, Apr. 2006.
[12] A. P. Subramanian, H. Gupta, S. R. Das, and M. M. Buddhikot, "Fast spectrum allocation in co-ordinated dynamic spectrum access based cellular networks," *Proc., IEEE Symposium on New Frontiers in Dynamic Spectrum Access Networks (DySPAN'07)*, Apr. 2007.
[13] S. Sengupta, S. Brahma, M. Chatterjee, and N. S. Shankar, "Enhancements to cognitive radio based IEEE 802.22 air interface," *Proc. IEEE Intl. Conf. on Commun. (ICC'2007)*, pp. 5155–5160, Jun. 2007.
[14] D. WillKomm, M. Boghe, D. Hollos, J. Gross, and A. Wolisz, "Double hop: A new approach for dynamic fequency hopping in cognitive radio networks," *Proc., IEEE Symposium on Personal Indoor Mobile Radio Communications (PIMRC'08)*, Sep. 2008.
[15] Z. Fan and R. Zhang, "Spectrum allocation and medium access in cognitive radio wireless networks," *Proc., European Wireless 2009*, May 2009.
[16] I. F. Akyildiz, W. Lee, M. C. Vuran, and S. Mohanty, "A survey on spectrum management in cognitive radio networks," *IEEE Commun. Mag.*, vol. 46, no. 4, pp. 40–48, Apr. 2008.
[17] H. Omranpour, M. Ebadzadeh, S. Barzegar, and M. Shojafar, "Distributed coloring of the graph edges," *Proc., IEEE Intl. Conf. on Cybernetic Intelligent Systems (CIS'2008)*, Sep. 2008.
[18] V. S. Rao, "Allocation of oppourtunistic spectrum for cognitive radio adhoc networks," *Master of Science Dissertation, Delft University of Technology, Netherlands*, Oct. 2009.
[19] B. Bollobas, *Random Graphs*. Cambridge University Press, 2001.
[20] F. Kuhn and R. Wattenhofer, "On the complexity of distributed graph coloring," *ACM SIGACT-SIGOPS Principles of Distributed Computing (PODC'2006)*, pp. 7–15, Jul. 2006.
[21] S. Anand, A. Sridharan, and K. N. Sivarajan, "Performance analysis of channelized cellular systems with dynamic channel allocation," *IEEE Trans. on Vehic. Technol.*, vol. 52, no. 4, pp. 847–859, Jul. 2003.
[22] R. Jain, *The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation and Modeling*. John Wiley & Sons, 1991.
[23] Z. Jin, S. Anand, and K. P. Subbalakshmi, "Performance analysis of channelized dynamic spectrum networks under primary user emulation attacks," *Proc., IEEE Global Commun. Conf. (GLOBECOM'2010)*, Dec. 2010.
[24] T. S. Rappaport, *Wireless Communications: Principles and Practice*. Prentice Hall Inc., New Jersey, 1996.

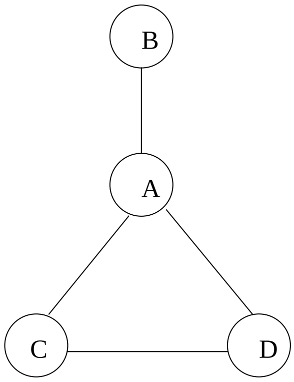


Fig. 1. An example graph representing a system with 4 secondary networks and their adjacency.

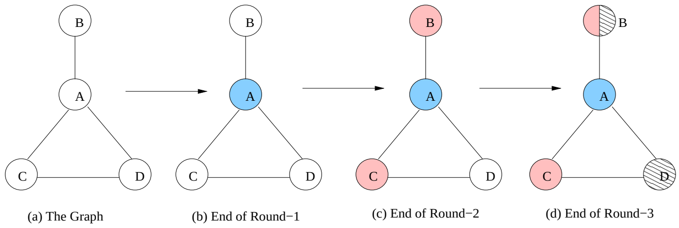


Fig. 2. Color assignment sequence when Steps 1-5 of the MASPECT Algorithm are applied to a sample graph.

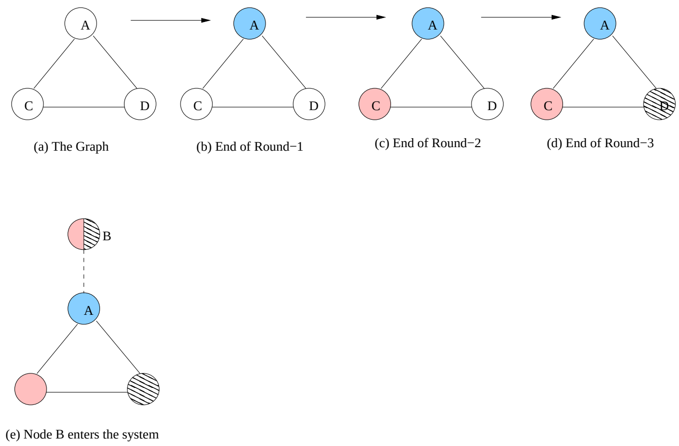


Fig. 3. Example to illustrate how Algorithm 1 adapts to the entry of a new node in the system. Color assignment sequence when Steps 1-5 of Algorithm 1 are applied to the graph in (a) are shown in $(b) - (d)$. A new node (node B) then enters the system.

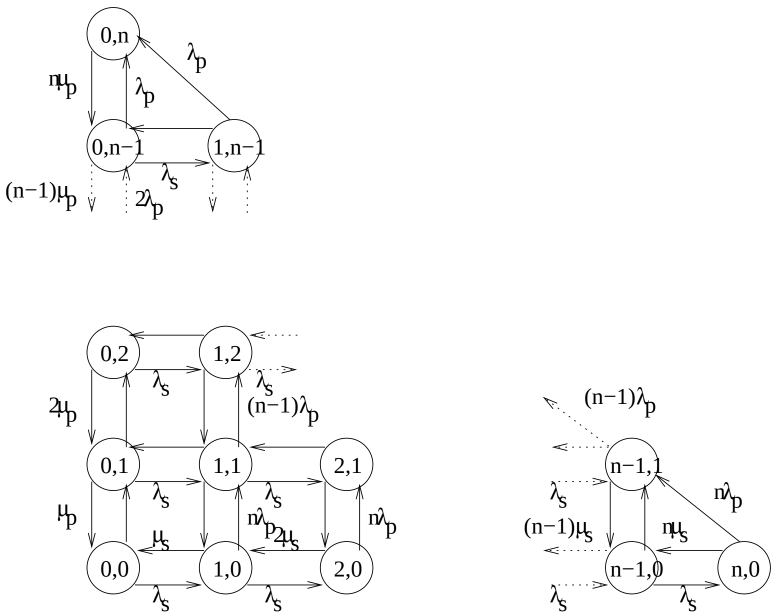


Fig. 4. CTMC that models the channel occupancy in a DSA network in the presence of primary activity.

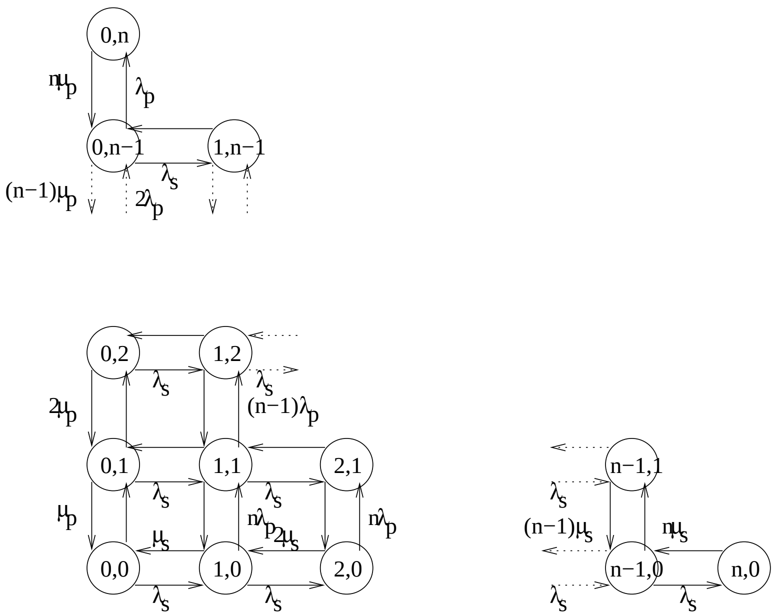
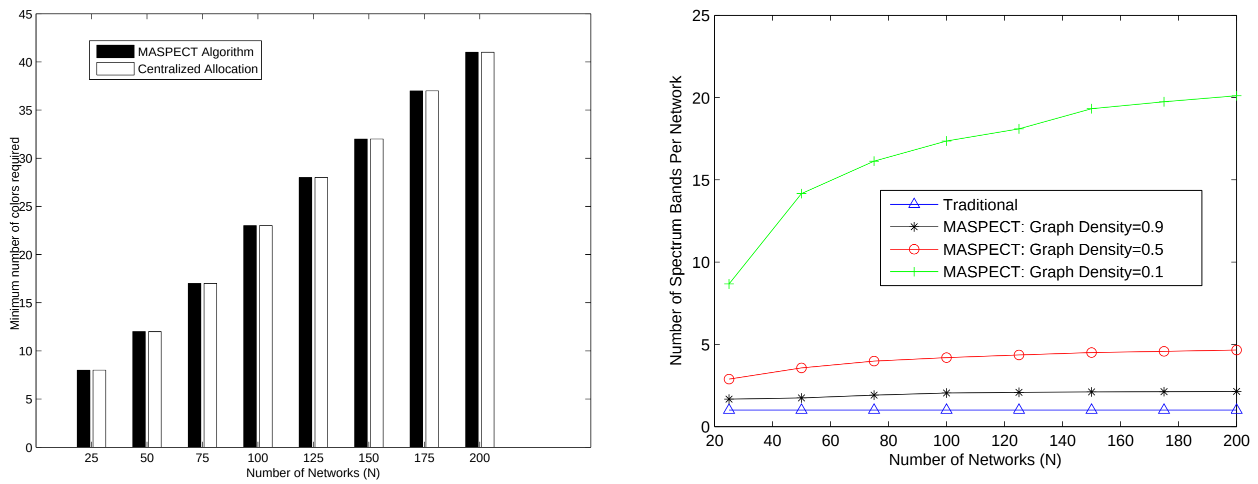


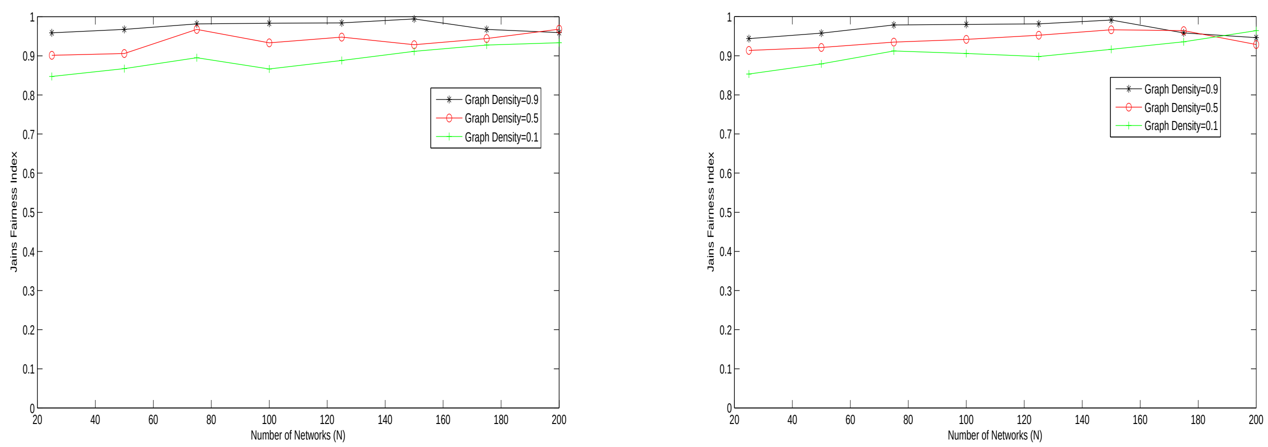
Fig. 5. Modified CTMC that approximately models the channel occupancy in a DSA network in the presence of primary activity. This is obtained by neglecting transitions from state (s, p) to $(s-1, p+1)$ when $s + p = n$ in Fig. 4.



(a) Minimum number of colors required

(b) Allocated spectrum per network

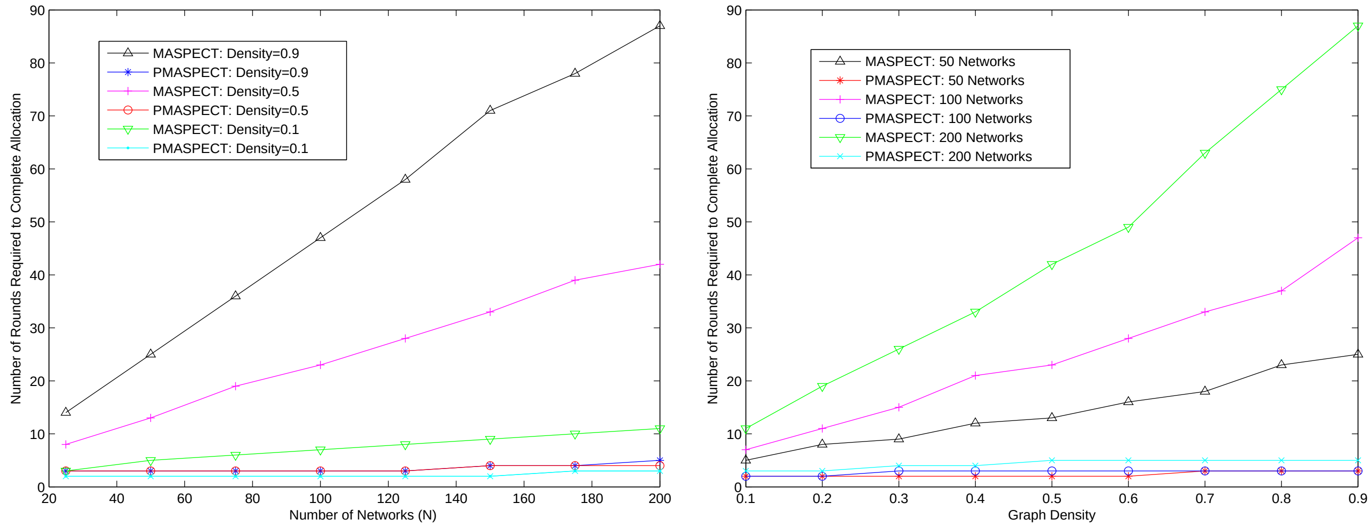
Fig. 6. Performance of the MASPECT algorithm.



(a) MASPECT algorithm

(b) PMASPECT heuristic

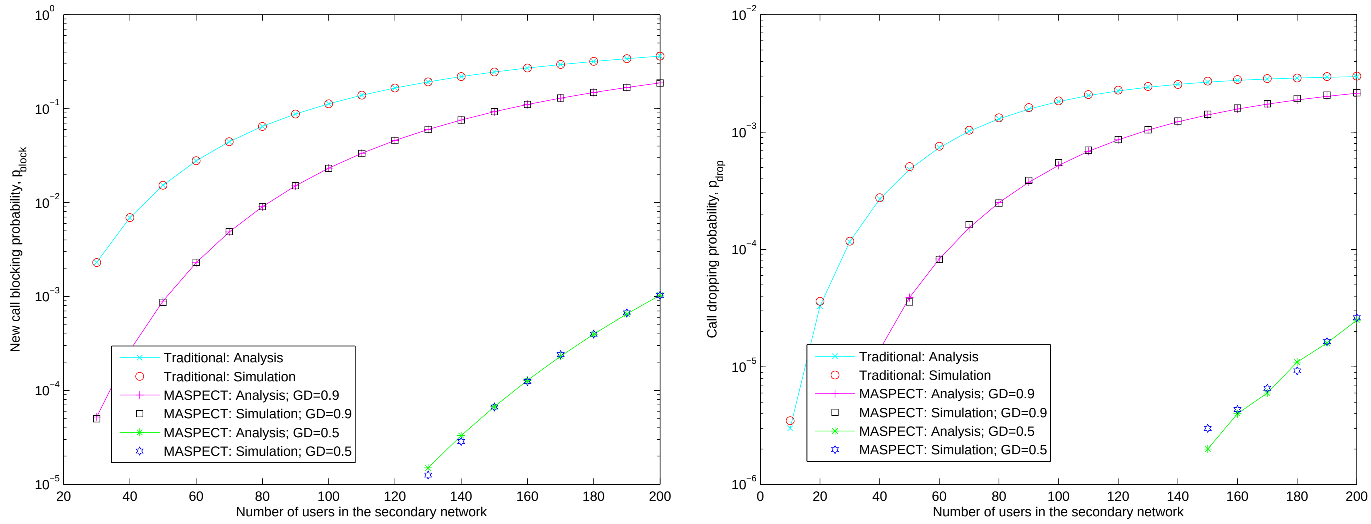
Fig. 7. Jain's Fairness Index



(a) Vs number of networks

(b) Vs graph density

Fig. 8. Rate of termination.



(a) New Call Blocking

(b) Call Dropping

Fig. 9. Performance of a secondary network. GD represents graph density.