Optimal Guard-band-aware Channel Assignment with Bonding and Aggregation

Gulnur Selda Uyanik¹, Mohammad J. Abdel Rahman², and Marwan Krunz² ¹Dept. of Computer Engineering, Istanbul Technical University, Istanbul, 34469, Turkey seldauyanik@itu.edu.tr

²Dept. of Electrical and Computer Engineering, University of Arizona, Tucson, AZ 85721 {mjabdelrahman, krunz}@email.arizona.edu

> Technical Report TR-UA-ECE-2013-1 Last update: March 27, 2013

Abstract

Channel assignment mechanisms in dynamic spectrum access (DSA) networks are often designed without accounting for adjacent-channel interference (ACI) between different secondary users (SUs). To prevent such interference, guard-bands are needed between channels that are assigned to different SUs. However, introducing guard-bands restricts the spectrum efficiency. In this paper, we consider the problem of designing an ACI-aware channel assignment for DSA networks that maximizes the spectrum efficiency. First, we consider a single link. The optimal assignment that maximizes the spectrum efficiency is formulated as a subset sum problem (SSP). An exponential-time dynamic programming (DP) exact algorithm, along with polynomial-time greedy and ϵ -approximate algorithms are proposed and compared. Next, a set of links is considered, and the optimal exponential-time assignment that maximizes the network spectrum efficiency is derived. A distributed implementation of the jointly optimal channel assignment for multiple links is presented. This distributed solution is compared with the sequential assignment, in which channels are assigned to links sequentially.

Index Terms

Channel assignment, dynamic spectrum access, spectrum efficiency.

I. INTRODUCTION

The continuous emergence of new wireless technologies has significantly increased the demand for more radio spectrum, resulting in over-crowded unlicensed frequency bands (e.g., ISM bands). Numerous studies have shown that licensed bands are vastly underutilized. Motivated by the need for more efficient utilization of the licensed spectrum and facilitated by recent regulatory policies, significant research has been conducted towards developing cognitive radio (CR) technologies for dynamic spectrum access (DSA) networks. CR devices utilize the available spectrum in a dynamic and opportunistic fashion without

This technical report is for [1].

interfering with co-located primary users (PUs). The communicating entities of an opportunistic CR network (CRN) are called secondary users (SUs).

Adjacent channel interference (ACI) is a form of power leakage from adjacent channels, attributed to imperfect design of filters and amplifiers in the radio device. The harmful impact of ACI on network throughput was demonstrated in [2]. Most previous channel assignment algorithms implicitly assume the existence of ideal filters and amplifiers, as shown in Figure 1(a). In this figure, two links *A* and *B* are assigned adjacent channels 1 and 2, respectively, assuming no power leakage between these channels. Figure 1(b) shows the actual power spectral density of channels 1 and 2 in a practical communication system. As discussed in [3], to mitigate ACI, *guard-bands* are needed between adjacent channels that belong to different SUs, as shown in Figure 1(c).



Fig. 1: Need for guard-band channels.

However, introducing guard-bands constrains the spectrum efficiency. In [3], the authors studied two models for utilizing guard-bands in a DSA network: guard-band reuse and no guard-band reuse. According to the guard-band reuse model, guard-bands can be shared by two adjacent (different) transmissions. In contrast, in the no guard-band reuse model two adjacent transmissions require two distinct guard-bands. As explained in [3], the guard-band reuse model is suitable for discontinuous-orthogonal frequency division multiplexing (D-OFDM)-based systems, whereas the no guard-band reuse model is suitable for FDM-based systems. In this paper, we adopt the guard-band reuse model. The guard-band-aware (GBA) channel assignment algorithm in [3] for the guard-band reuse case does not achieve the maximum spectrum efficiency, as will be shown later in this paper.

To support applications with high rate demands, the most recent IEEE 802.11n and the upcoming IEEE

802.11ac standards have adopted channel bonding [4]. Channel bonding refers to the bundling of multiple adjacent channels, which can then be treated as a single block whose data rate is approximately the sum of the individual channel data rates. On the other hand, bundling multiple non-adjacent frequency channels is referred to as channel aggregation. The channel assignment schemes proposed in this paper support both channel bonding as well as channel aggregation.

Our Contributions–The main contributions of this paper are as follows:

- We formulate and obtain the optimal GBA channel assignment for an SU link operating in a DSA network that adopts the guard-band reuse paradigm. This assignment achieves the maximum spectrum efficiency. The problem of obtaining the channel assignment that maximizes the spectrum efficiency is mapped to the subset sum problem (SSP) [5].
- We formulate and obtain the optimal GBA channel assignment for a DSA network consisting of multiple links, adopting the guard-band reuse paradigm. The joint assignment that maximizes the spectrum efficiency is obtained assuming a distributed setup.
- We evaluate the exponential-time optimal single link and multiple links assignment mechanisms and compare them with several polynomial-time approximate algorithms.

Paper Organization–The remainder of this paper is organized as follows. In Section II, we present the system model followed by the problem statement. The single-link optimal channel assignment is explained in Section III. Polynomial-time greedy and ϵ -approximate algorithms are also presented in the same section. In Section IV, we address the problem of optimal GBA channel assignment for multiple links, considered as a group. We provide an exponential-time exact algorithm along with an approximate sequential algorithm. We evaluate the single- and multiple-link assignment algorithms in Section V. Section VI gives an overview of related work. Finally, Section VII concludes the paper.

II. PROBLEM STATEMENT

We consider an opportunistic DSA environment, with M licensed channels and L SU links. Each channel can be in one of four states: occupied by a PU, occupied by an SU, reserved as a guard-band, or

available for opportunistic communication. All available channels support a common rate of r Mbps. Each link j has a rate demand $d_j \stackrel{\text{def}}{=} \alpha_j r$ Mbps, where α_j is an integer between 1 and M. Given the current spectrum status, i.e., the state of each of the M licensed channels, our objective is to satisfy the demands of the L links while maximizing the spectrum efficiency. Figure 2 shows an example of a spectrum status.



Fig. 2: Example of a spectrum status.

The spectrum efficiency, denoted by SE, associated with a given channel assignment is defined as the fraction of the idle spectrum that can be used for opportunistic communications. Let h_{ij} , $i \in \{1, 2, ..., M\}$, $j \in \{1, 2, ..., M\}$, be a binary variable, which equals one if channel *i* is assigned to link *j* as a data channel, and zero otherwise. Let η_i be a binary variable indicating whether or not the *i*th channel is used as a guard-band channel. Then, the SE of this assignment is defined as follows:

$$SE \stackrel{\text{def}}{=} \frac{\sum_{j=1}^{L} \sum_{i=1}^{M} h_{ij}}{\sum_{j=1}^{L} \sum_{i=1}^{M} h_{ij} + \sum_{i=1}^{M} \eta_i}.$$
 (1)

In this paper, we consider the following two problems.

Problem 1. Given a link with a rate demand of d Mbps and given the current states of the M channels, find the optimal GBA channel assignment that maximizes the spectrum efficiency while satisfying the rate demand d.

Problem 2. Given a set of L links with a rate demand of d_j Mbps for link j, and given the current states of the M channels, find the optimal GBA channel assignment that maximizes the network-wide spectrum efficiency.

III. OPTIMAL GUARD-BAND-AWARE CHANNEL ASSIGNMENT FOR A SINGLE-LINK

In this section, we consider Problem 1. In this case, SE can be expressed as:

$$SE = \frac{\sum_{i=1}^{M} h_i}{\sum_{i=1}^{M} h_i + \sum_{i=1}^{M} \eta_i} = \frac{d}{d + \sum_{i=1}^{M} \eta_i}$$
(2)

where h_i is a binary variable indicating whether channel *i* is assigned for data communication. The equality in (2) holds because we assume the problem is feasible, i.e., there is a feasible assignment that can satisfy the link demand *d*. According to (2), in order to maximize the *SE*, the number of introduced guard bands (i.e., $\sum_{i=1}^{M} \eta_i$) needs to be minimized. Next, we show that in order to minimize this number, channels need to be assigned on a per-block basis. Consider the spectrum status in Figure 2. Each set of consecutive idle channels are grouped into a frequency block, as illustrated in Figure 3, which shows 4 idle blocks. Let \mathcal{N} denote the set of idle frequency blocks, and let $N = |\mathcal{N}|$. Let $R_i \stackrel{\text{def}}{=} \beta_i r$ Mbps denotes the rate supported by block *i*, where β_i is an integer between 1 and *M*. As justified in [3], we assume that one fixed-bandwidth guard-band is sufficient to prevent ACI, irrespective of the block size and transmission power. Note that in a DSA system, the transmission powers for SUs are strictly limited by power masks.



Fig. 3: Set of idle blocks for the spectrum map in Figure 2.

Theorem 1. Assigning channels on a per-block basis achieves the optimal SE.

Proof. We will show that assigning channels on a per-block basis introduces at most one additional guard-band. Consider the set of idle blocks \mathcal{N} . There are two cases to consider:

Case 1: $\exists \mathcal{B} \subseteq \mathcal{N}$ such that $\sum_{i \in \mathcal{B}} R_i = d$. This is shown in Figure 4, where d = 6 Mbps can be met using two blocks of idle channels of rates 1 Mbps and 5 Mbps.



Fig. 4: Channel assignment with no additional guard-bands (d = 6 Mbps).

In this case, the number of introduced guard-bands is zero (recall that we assume the guard-band reuse

model). This is clearly an optimal assignment.

Case 2: $\nexists \mathcal{B} \subseteq \mathcal{N}$ such that $\sum_{i \in \mathcal{B}} R_i = d$.

In this case, let $\mathcal{B} \subset \mathcal{N}$ be the largest set such that $\sum_{i \in \mathcal{B}} R_i < d$. Then, we assign the set of channels in \mathcal{B} to this link, in addition to $d - \sum_{i \in \mathcal{B}} R_i$ channels extracted from the beginning of one of the idle blocks in $\mathcal{N} \setminus \mathcal{B}$, as shown in Figure 5. In this figure, d = 7 Mbps cannot be exactly met by any combination of idle blocks. The demand d is satisfied using two complete blocks of rates 1 Mbps and 5 Mbps, and one channel at the beginning of the 4th idle block that can support a data rate of 4 Mbps. This results in one additional guard-band, which is optimal because any other feasible assignment will introduce at least one guard-band (if there is an assignment with zero new guard-bands, then this contradicts the assumption made in case 2). Hence, the total number of introduced guard-bands is either zero or one.



Fig. 5: Channel assignment with one additional guard-band (d = 7 Mbps).

Having established that assigning channels on a per-block basis results in the optimal SE, Problem 1 can be re-stated as follows: Given a set of idle blocks \mathcal{N} with block *i* supporting a rate demand of R_i Mbps, obtain a combination of idle blocks that either satisfies the link demand *d*, or achieves the nearest rate to *d*. This is exactly the subset sum problem (SSP) [5], with the items being the idle frequency blocks and the weights of the items the rates supported by the idle blocks. Let x_i be a binary variable indicating whether or not idle block *i* is assigned to the link. Then, the optimal GBA channel assignment can be formulated as follows:

$$\begin{array}{l} \underset{x_{i},1\leq i\leq N}{\operatorname{maximize}} \left\{ \mathcal{R}_{s} \stackrel{\text{def}}{=} \sum_{i=1}^{N} R_{i} x_{i} \right\} \\ \text{subject to} \quad \sum_{i=1}^{N} R_{i} x_{i} \leq d \\ x_{i} \in \{0,1\}, \forall i \in \{1,2,\ldots,N\} \end{array} \tag{3}$$

Let \mathcal{R}_s^* denote the optimal solution for the SSP problem. From (3), $\mathcal{R}_s^* \leq d$. When $\mathcal{R}_s^* < d$, we augment the SSP problem with *a post-processing phase*. As it stated in Lemma 1, each of the remaining idle blocks after executing the SSP problem supports a data rate greater than $d - \mathcal{R}_s^*$. In the post-processing phase, we assign a portion of $d - \mathcal{R}_s^*$ channels from any of the remaining idle blocks, starting from the beginning of the block. The assigned channels are followed by a guard-band, as shown in Figure 5.

Lemma 1. Let C be the set of assigned idle blocks after solving the SSP problem, and assume $\mathcal{R}_s^* < d$. Then, $R_i > d - \mathcal{R}_s^*, \forall i \in \mathcal{N} \setminus C$.

Proof. We prove Lemma 1 by contradiction. Suppose $\exists i \in \mathcal{N} \setminus \mathcal{C}$ with $R_i \leq d - \mathcal{R}_s^*$. Then, this block will be selected by the SSP problem, because SSP selects the combination of idle blocks that achieves the nearest rate to d, and by assumption \mathcal{R}_s^* is the optimal solution to the SSP problem. Hence, block $i \in \mathcal{C}$, but we assume that $i \in \mathcal{N} \setminus \mathcal{C} \notin \mathcal{C}$. This leads to a contradiction.

Theorem 2. When augmented with the post-processing phase, SSP attains the optimal GBA channel assignment that achieves the maximum SE.

Proof. There are two cases to consider.

Case 1: $\mathcal{R}_s^* = d$. In this case, $\sum_{i=1}^M \eta_i = 0$ and SE = 1, which is optimal.

Case 2: $\mathcal{R}_s^* < d$. In this case, by Lemma 1 and Theorem 1, $\sum_{i=1}^M \eta_i = 1$ and the $SE = \frac{d}{d+1}$, which is optimal. There is no any other feasible assignment that results in a higher SE. The reason is that by Lemma 1, any feasible assignment will introduce at least one additional guard-band.

SSP is an NP-complete problem [5]–[7]. In the following subsections, we present exact and approximate algorithms for solving the SSP problem.

A. Exhaustive Search Exact Algorithm

The exhaustive search algorithm examines all subsets of set \mathcal{N} , and returns the one whose sum of rates of its elements is closest to d, but does not exceed d. This exhaustive search algorithm runs in $\mathcal{O}(N2^N)$ time, where N is the number of idle blocks [6]. In Section V-A, we refer to this algorithm as exponential exact.

$$R_{s}^{*}(i,\tilde{d}) = \begin{cases} R_{s}^{*}(i-1,\tilde{d}), & \text{if } \tilde{d} < R_{i} \\ \max\left(R_{s}^{*}(i-1,\tilde{d}), R_{s}^{*}(i-1,\tilde{d}-R_{i}) + R_{i}\right), & \text{if } R_{i} \le \tilde{d} \le d. \end{cases}$$
(5)

B. Dynamic Programming (DP)-based Exact Algorithm

The idea of the DP-based approach is the following. For each subset of idle blocks, find the maximum achievable rate that is less than or equal to d. A pseudo-code of the DP-based exact channel assignment algorithm is shown in Algorithm 1 [7]. Consider the sub-instance of SSP consisting of idle blocks $1, \ldots, i-$ 1 and demand \tilde{d} . If the rate supported by the *i*th idle block exceeds \tilde{d} (i.e., $R_i > \tilde{d}$), then idle block *i* will not be included in the optimal assignment. If $R_i \leq \tilde{d}$, then idle block *i* will be included in the optimal assignment if this results in better solution value than excluding it. Let $R_s^*(i, \tilde{d})$ be the optimal solution value of the sub-instance of the SSP consisting of idle blocks $1, \ldots, i$ and demand \tilde{d} . Then, the recurrence relation is given by (5) (note that $R_s^*(N, d) \stackrel{\text{def}}{=} R_s^*$).

The DP-based algorithm correctly computes the optimal value of SSP, and runs in $\mathcal{O}(Nd)$ time [7], where N is the number of idle blocks and d is the rate demand.

Algorithm 1 DP-b	ased Exact SSP Algorithm
1: Input: N, d, N by d	+1 array M
2: Initialize: $M[1, \tilde{d}] = 0$	$0, orall ilde d \in \{0, 1, \dots, d\}$
3: for $i = 1 : N$ do	
4: for $\tilde{d} = 0 : d$ do	
5: if $\tilde{d} < R_i$ then	
6: $M[i, \tilde{d}] \leftarrow M[$	$[i-1, ilde{d}]$
7: else	
8: $M[i, \tilde{d}] \leftarrow \max$	$ \left\{ M[i-1,\tilde{d}], R_i + M[i-1,\tilde{d}-R_i] \right\} $

C. ϵ -approximate Algorithm

9:

 $10 \cdot$

end if

end for 11: end for 12: Return: M

The ϵ -approximate algorithm is a fully polynomial-time algorithm [6]. Its running time is polynomial in both $1/\epsilon$ and N. It returns a value that is within a $(1 + \epsilon)$ factor of R_s^* .

A pseudo-code of the ϵ -approximate algorithm is shown in Algorithm 2. The algorithm selects the combination of idle blocks that results in a total rate that is closest to d, and reports the total rate value. At the *i*th iteration of the for loop in line 3 in the pseudo-code, the algorithm considers all combinations of *i* idle blocks. For each combination of *i* blocks, the algorithm stores their total rate in one of the elements of the *i*th list, denoted by L_i . List L_i is obtained by merging lists L_{i-1} and L_{i-1} , augmented with R_i , using the MERGE-LISTS function, which combines the two lists into a one ascendingly ordered list with no duplicate elements. The addition operation in line 4 is a per-element addition operation. The approximate algorithm uses a function called TRIM which trims the lists L_i , i = 1, ..., N to reduce their lengths. TRIM removes an element with value *a* from the list if there is another element with value *b*, such that $|a - b| \le \delta$. In [6], δ is set to $\epsilon/2N$.

Algorithm 2 *e*-approximate SSP Algorithm

```
1: Input: \mathcal{N}, d, \epsilon, and q

2: L_0 \leftarrow \emptyset

3: for i = 1 : N do

4: L_i \leftarrow \text{MERGE-LISTS} (L_{i-1}, L_{i-1} + R_i)

5: L_i \leftarrow \text{TRIM} (L_i, \epsilon/2N)

6: Remove from L_i every element that is greater than q

7: end for

8: Let z^* be the largest element in L_N

9: Return: z^*
```

D. Greedy Scheme

The greedy approach starts with the set of idle blocks, sorted descendingly in their supported data rates. It passes through the sorted list and adds the idle blocks sequentially as long as the total rate will not exceed the demand d. The complexity of the algorithm comes from the sorting phase and the traversal of the sorted array. The complexity of this greedy algorithm is $\Theta(N \log N + N)$ if any sorting algorithm with complexity $\mathcal{O}(N \log N)$ is used. An example of a sorting algorithm with complexity $\mathcal{O}(N \log N)$ is used. An example of a sorting algorithm with complexity $\mathcal{O}(N \log N)$ is the merge sort algorithm.

In contrast to the other algorithms, in the ϵ -approximate algorithm, there is a chance after executing the algorithm to find idle blocks with rates less than or equal to the remaining unsatisfied demand, i.e., with probability p > 0, \exists an unassigned block i such that $R_i \leq d - \sum_{j=1}^N R_j \eta_j$. If \exists an unassigned block i such that $R_i = d - \sum_{j=1}^N R_j \eta_j$, then the ϵ -approximate algorithm can be turned into optimal by searching for such blocks and including them in the assignment.

It is also to be noted that the input size of the above algorithms is the number of idle blocks N which is typically much smaller than the total number of idle channels M, i.e., $N \ll M$ (recall that N depends not only on M, but also on p_{busy}). Therefore, the exponential-time exact algorithms can be used to retrieve the optimal single-link assignment within a reasonable amount of time.

IV. OPTIMAL GUARD-BAND-AWARE CHANNEL ASSIGNMENT FOR MULTIPLE-LINKS

When multiple-links are considered, there are two approaches for assigning channels to links. The first approach is the sequential assignment approach, in which the demands of various links in the network are satisfied sequentially according to some order; one link is considered at each step. Each link can be assigned following one of the algorithms discussed in Section III. It is clear that the sequential assignment does not necessarily result in the network-wide optimal spectrum efficiency. In order to obtain the network-wide optimal assignment, the alternative approach is the *batch assignment* approach. In the batch approach, all links are assigned jointly such that the network-wide spectrum efficiency is maximized.

In order to attain the network-wide optimal assignment in a distributed network, we borrow the *access window (AW)* concept proposed in [8], [9], where each link broadcasts its rate demand throughout the network. Each link waits for a certain amount of time to collect the demands of other links in the network before executing the joint assignment problem. This time duration is called the access window, and is denoted by AW.

An intuitive way of modeling the optimal GBA channel assignment problem for multiple-links is to use the multiple subset sum problem (MSSP) [10], [11]. MSSP is a variant of the multiple knapsack problem (MKP), in which the price of an item is equal to its weight. More specifically, since we assume that different links have different rate demands (demand d_j for link j), the MSSP version with different capacities is the most attractive model. Let $x_{ij}, i \in \{1, 2, ..., N\}, j \in \{1, 2, ..., L\}$ be a binary variable, which equals 1 if idle block i is assigned to link j and zero otherwise. Then, using the MSSP with different capacities model, our channel assignment for multiple-links can be modeled as follow.

$$\max_{x_{ij}, 1 \le i \le N, 1 \le j \le L} \left\{ \mathcal{R}_m \stackrel{\text{def}}{=} \sum_{j=1}^L \sum_{i=1}^N R_i x_{ij} \right\}$$

subject to
$$\sum_{i=1}^N R_i x_{ij} \le d_j, \forall j \in \{1, 2, \dots, L\}$$
(6)

$$\sum_{j=1}^{L} x_{ij} \le 1, \forall i \in \{1, 2, \dots, N\}$$
(7)

$$x_{ij} \in \{0, 1\}, i \in \{1, 2, \dots, N\}, j \in \{1, 2, \dots, L\}.$$
 (8)

Several approximate and heuristic algorithms for the MKP and MSSP problems have been proposed in the literature. Examples include [12], [13].

In the case of a single-link, SSP augmented with the post-processing phase achieves the maximum SE, as proved in Theorem 2. SSP maximizes the rate assigned to the link, while keeping it less than or equal to the link demand. In the case of multiple-links, maximizing the total rate assigned to the L links, as the MSSP does, does not necessarily achieve the maximum SE. Recall from (1) that the SE depends not only on the total rate assigned to the L links (i.e., $r \sum_{j=1}^{L} \sum_{i=1}^{M} h_{ij}$), but also on the number of introduced guard-bands (i.e., $\sum_{i=1}^{M} \eta_i$). Therefore, the optimal channel assignment for multiple-links should consider the total rate assigned to the links, as well as the number of introduced guard-bands.

Moreover, similar to SSP, MSSP needs to be augmented with a post-processing phase. The postprocessing phase of MSSP is more complicated than that of SSP; because the result of the post-processing phase in MSSP depends on the order of serving unsatisfied links after executing the MSSP problem. To obtain the optimal assignment, we need to consider all combinations of unsatisfied link orders and remaining idle blocks. Even with the optimal assignment in the post-processing phase, MSSP augmented with the post-processing phase does not result in the optimal assignment that maximizes the spectrum efficiency. To illustrate this, consider the following example of two links with demands $d_1 = 3$ Mbps and $d_2 = 7$ Mbps. There exists two idle blocks of sizes $\beta_1 = 2$ and $\beta_2 = 11$. MSSP will assign the first idle block to one of the links. Then, in the post-processing phase, either one channel will be assigned to the first link and seven channels to the second link, both are taken from the second idle block, or three channels to the first link and five channels to the second link, both are taken from the second idle block. In both cases, two additional guard-bands will be introduced. However, there exists a better assignment with higher *SE*, that is, assign three channels to the first link and seven channels to the second link, both from the second idle block, without using the first idle block. In this case, only one additional guard-band is introduced.

To obtain the network wide optimal assignment for multiple-links, the objective function in MSSP needs to be replaced with the following objective function.

$$\underset{\substack{x_{ij}, 1 \leq i \leq N, 1 \leq j \leq L\\\eta_k, 1 \leq k \leq M}}{\text{maximize}} \left\{ \sum_{j=1}^{L} \sum_{i=1}^{N} R_i x_{ij} + \frac{1}{M} \sum_{k=1}^{M} \eta_k \right\}$$

where $\eta_k, k \in \{1, 2, ..., M\}$, is a binary variable, which equals one if channel k is reserved as a guard-band and zero otherwise.

Assume that r > 1 and hence $R_i = \beta_i r > 1, \forall i \in \mathcal{N}$. Then, the first term in the objective function dominates the second term; because the second term is always less than or equal to one. In other words, if there are two channel assignments to the above problem with the first assignment has a higher value of the first term than the second assignment, then the first assignment will be always selected irrespective of the values of the second term for the two assignments.

The above maximization problem is subject to constraints (6), (7), and (8), in addition to the constraints that relate the selection of data channels to the required guard-bands to be reserved. The latter constraints are complicated and require defining several auxiliary variables, so they removed from the paper for brevity.

In the joint assignment of multiple-links (i.e., the batch approach), each idle channel before the assignment will end up being in one of L + 2 states after the assignment: assigned to one of the L

links, reserved as a guard-band, or left unassigned. Therefore, obtaining the optimal solution by following the exhaustive search approach will yield an exponential complexity of $(L+2)^{I}$, where $I = \sum_{i=1}^{N} \beta_{i}$ and β_{i} , as defined in Section III, is equal to R_{i}/r . In the following subsection, we present an exponential-time exact algorithm, followed by an approximate sequential assignment algorithm.

A. Exponential-time Exact Algorithm

We implemented the optimal assignment of multiple-links that results in the maximum assigned rate with the minimum number of introduced guard-bands following an exhaustive search approach.

In the designed search tree, each node represents the state of each idle channel. Node i, denoted by $n_i = \{s_1, s_2, \ldots, s_I\}$, where $s_i \in \{D, G, 1, 2, \ldots, L\}$ with D means channel i is left idle, G means that is is reserved as a guard-band, and $k, k = 1, 2, \ldots, L$, means that it is assigned to link k. The depth of a node represents the number of determined variables in that node, i.e., the states of the first i channels, s_1, \ldots, s_i , for all nodes of depth i are determined.

To decrease the search space while ensuring the feasibility conditions, such as the required guard-bands for the assigned channels, we introduce the following pruning rules. Let us denote the current set of partially served links with \mathcal{P} . Then,

- If idle channel i is at the beginning of an idle block. Then, $s_i \in \{D, u\}$, where $u \in \mathcal{P}$.
- If idle channel i is not at the beginning of an idle block. Then,
 - If idle channel i 1 has been assigned to link y (i.e., $s_{i-1} = y$). Then,

If $y \in \mathcal{P}$, $s_i \in \{G, y\}$.

```
If y \notin \mathcal{P}, s_i = G.
```

- If idle channel i 1 has been reserved as a guard-band G (i.e., $s_{i-1} = G$). Then, $s_i \in \{D, u\}$, where $u \in \mathcal{P}$.
- If idle channel i 1 has not been assigned (i.e., $s_{i-1} = D$). Then, $s_i = D$.
- If the total number of assigned channels in node i located at depth t in the tree, denoted by A_i , is less

than $A_{\text{best}} + t - I$, where A_{best} is the total number of assigned channels in the current best solution.

Then, we do not branch further from node *i*, because this will not improve the current best solution.

Adding the above pruning rules reduces the running time of the brute force procedure significantly. However, the running time is still long, and we limit our simulations in Section V-B to small numbers of idle channels and links.

B. Approximate Sequential Assignment Algorithm

To avoid the exponential-time complexity of the exact algorithm, we propose assigning channels to links sequentially instead of jointly. Each link can be assigned using any of the algorithms proposed in Section III. We propose using the fast greedy algorithm for SSP in assigning channels to each individual link. The links can be assigned in different orders. In here, we implement three different ordering approaches: start with the link with smallest demand (denoted by SEQ_{ASC}), start with the link with largest demand (denoted by SEQ_{DSC}), and follow a random ordering of links (denoted by SEQ_{RND}). For comparisons in Section V-B, we have also implemented a version of the sequential assignment that uses the algorithm proposed in [3] for each individual link assignment. The algorithm proposed in [3] selects existing guard-bands and minimizes the number of assigned frequency blocks.

V. PERFORMANCE EVALUATION

In the following subsections, we evaluate the single-link and multiple-links assignment schemes, discussed in Sections III and IV.

A. Performance Evaluation of the Single-Link Assignment Algorithms

The exact algorithms explained in Sections III-A and III-B, the ϵ -approximate algorithm described in Section III-C, and the greedy algorithm described in Section III-D are all implemented in C++. In addition to these four algorithm, we implement the channel assignment scheme proposed in [3] in MATLAB. We refer to this scheme as *Choose all existing GBs* in the legends of the numerical figures. In this scheme,

the objective function is to minimize the number of assigned idle blocks that is required to meet a certain rate demand. This scheme selects all existing guard-bands.

All algorithms are simulated in a common setup specified by the values shown in Table I for various parameters, and using a common status of the spectrum. Simulation results are obtained for different rate demands d and different values of p_{busy} . Our numerical results are averaged over 50 runs, and the 95% confidence intervals are indicated in the figures.

Parameter	Value	
d	10 Mbps	
М	50	
p_{busy}	0.25	
ϵ	0.2	

TABLE I: Simulation parameters for the single-link assignment algorithms.

Figure 6 depicts the spectrum efficiency vs. p_{busy} for all discussed single-link assignment schemes. SE is computed as in (1). As shown in Figure 6, SSP algorithms achieve higher SE than previously proposed assignment scheme in [3]. This results from the fact that SSP-based assignment schemes are per-block schemes that inherently try to use existing guard-bands and avoid introducing any new guard-band, hence maximizing the SE. As p_{busy} increases, the number of existing guard-bands increases. This improves the performance of the SSP-based schemes; because the sizes of idle blocks become smaller, which increases the chance to find a subset of idle blocks whose sum rate is equal to the rate demand d. The performance of the scheme proposed in [3] also improves with increasing p_{busy} ; because of the reduction in the sizes of idle blocks. The idle blocks selected by this scheme may not change with increasing p_{busy} , but the probability that the first and last channels of these blocks are existing guard-bands increases, which increases the SE. As shown in Figure 6, the ϵ -approximate and greedy algorithms achieve a close SE to the optimal exponential and DP algorithms. ϵ -approximate outperforms the greedy algorithm. Figure 7 shows the SE vs. the rate demand d. SSP-based assignment algorithms outperform the one in [3] for all values of d.

The number of introduced guard-bands is depicted in Figure 8 for different values of p_{busy} . The SSPbased algorithms introduce smaller numbers of guard-bands compared to the one in [3]. Because of this,



Fig. 6: Spectrum efficiency vs. p_{busy}.



Fig. 7: Spectrum efficiency vs. d.

the SE of the SSP-based algorithms is higher. As shown in Figure 8, the number of introduced guardbands in the SSP-based algorithms is always less than or equal to one, which is consistent with the result in Theorem 2. Figure 9 shows the number of introduced guard-bands for different values of d. SSP-based assignment algorithms outperform the one in [3] for all values of d.

When channel availability decreases with increasing p_{busy} , the chance of not meeting the link demand increases. Figure 10 shows the fraction of the 50 runs that report infeasibility for different values of p_{busy} . We call this fraction the infeasibility ratio. As shown in Figure 10, the infeasibility ratio can reach up



Fig. 8: Number of introduced guard-bands vs. p_{busy}.



Fig. 9: Number of introduced guard-bands vs. d.

to 0.45 when $p_{\text{busy}} = 0.4$. The infeasibility ratio is also shown for various values of d in Figure 11. As shown in Figures 10 and 11, all algorithms result in very close infeasibility ratios.

B. Performance Evaluation of the Multiple-Links Assignment Algorithms

The exact algorithm in Section IV-A, the MSSP algorithm (augmented with the post-processing phase), and the sequential assignment algorithms in Section IV-B are all implemented in C++. In the post-processing phase of MSSP, partially satisfied links are considered in a random order. In the sequential assignment algorithms (SEQ_{ASC}, SEQ_{DSC}, and SEQ_{RND}), greedy SSP is adopted for each single-link



Fig. 10: Infeasibility ratio vs. p_{busy}.



Fig. 11: Infeasibility ratio vs. d.

assignment. A sequential assignment algorithm with a random order where each link is assigned according to the scheme in [3] is also implemented in MATLAB. This scheme is called *Choose all existing GBs* in the legends of the numerical figures. Because of the exponential-time complexity of the exact algorithm in Section IV-A, it is evaluated for small values of the system parameters, given in Table II, where p_{busy} is fixed and *L* is varying. In addition to the setup in Table II, the sequential assignment algorithms are also evaluated using the parameters listed in Table III, where *L* is fixed and p_{busy} is varying. MSSP and the Choose all existing GBs algorithms are evaluated for the setup in Table III. Our numerical results are averaged over 50 runs, and the 95% confidence intervals are indicated in the figures.

TABLE II: Simulation parameters for the exponential-time exact multiple-links assignment algorithm.

Parameter	Value		
M	50		
p_{busy}	0.4		
d_{\min}	1 Mbps		
d_{\max}	5 Mbps		

TABLE III: Simulation parameters for the approximate multiple-links assignment algorithms.

Parameter	Value		
M	150		
L	10		
d_{\min}	2 Mbps		
d_{\max}	10 Mbps		

Table IV shows the fraction of runs that SEQ_{ASC} , SEQ_{DSC} , and SEQ_{RND} result in a sub-optimal solution, for different values of *L*.

SEQ Alg.	L=2	L = 4	L = 6	L = 8	L = 10
SEQ _{ASC}	0.04	0.28	0.6	0.78	0.84
SEQ _{DSC}	0.20	0.34	0.18	0.22	0.20
SEQ _{RND}	0.08	0.34	0.46	0.48	0.48

TABLE IV: Fraction of runs with sub-optimal results.

Let us define the service ratio, denoted by SR, as follows.

$$SR = \frac{\sum_{j=1}^{L} \sum_{i=1}^{N} R_i x_{ij}}{\sum_{j=1}^{L} d_j}.$$
(9)

Figure 12 depicts the SR vs. L for the optimal and sequential algorithms. As shown in Figure 12, the sequential greedy approaches achieve very close to optimal SR, even if the number of sub-optimal results in Table IV is large. The number of introduced guard-bands and the SE are plotted in Figures 13 and 14, respectively. The relative performance of the three sequential algorithms depends on the states of the channels and the demands of the links. This is the reason for the large intersecting confidence intervals

for the sequential algorithms. The average behavior shows that the SEQ_{DSC} outperforms the SEQ_{ASC} and SEQ_{RND} in terms of SE and SR, especially for large L.



Fig. 12: Service ratio vs. L for multiple-links.



Fig. 13: Number of introduced guard-bands vs. L for multiple-links.

Figure 15 depicts the SR vs. p_{busy} for the exact MSSP, the sequential algorithms, and the algorithm in [3]. SR decreases with p_{busy} . All algorithms achieve very close SRs, but they achieve different performance in terms of the number of introduced guard-bands and the SE, as shown in Figures 16 and 17, respectively. MSSP achieves a better average performance than the sequential algorithms; because, even though it is not optimal, it assigns channels to links jointly considering all links demands. MSSP and the sequential



Fig. 14: Spectrum efficiency vs. L for multiple-links.

algorithms SEQ_{DSC}, SEQ_{ASC}, and SEQ_{RND} outperform the scheme in [3]. As shown in Figure 16, the inefficient performance of the scheme in [3] is more noticeable when p_{busy} is small, which leads to idle blocks of large sizes. Since the algorithm in [3] aims at minimizing the the number of assigned blocks, larger blocks will be preferable over smaller blocks, which introduces more guard-bands and reduces the *SE*. The increase in the number of introduced guard-bands also reduces the *SR* given in (9).



Fig. 15: Service ratio vs. p_{busy} for multiple links.



Fig. 16: Number of introduced guard-bands vs. p_{busy} for multiple-links.



Fig. 17: Spectrum efficiency vs. p_{busy} for multiple-links.

VI. RELATED WORK

Most of the existing channel assignment schemes in DSA networks do not account for ACI. PU traffic is directly affected by the usage of guard-bands, as discussed in [14]. In [14], independently developed DSA testbeds were used in a practical coexistence setup to explore the effect of reducing the guard-bands between the PU and the coexisting SUs. Guard-bands are needed not only to protect PUs, but also other SUs. The GBA channel assignment problem is considered in [3] under two guard-bands sharing paradigms: guard-band reuse and no guard-band reuse. In this paper, we have considered the guard-band

reuse paradigm. The proposed assignment scheme in [3] for the guard-band reuse case is not optimal in terms of the spectrum efficiency. The scheme in [3] was designed to minimize the number of idle blocks assigned to a given link, aiming by that to minimize the number of guard-bands. The motivation behind this was that each idle block requires two guard-bands. However, under the guard-band reuse paradigm, minimizing the number of blocks does not necessarily result in the optimal spectrum efficiency, as we had shown in the paper.

In [15], the amount of required guard-bands is determined based on the differences in the capacity limits of the used spectrum. A designated spectrum broker is used to manage spectrum sharing among different users with different priorities. In [16], a centralized adaptive guard-band configuration, called *Ganache* was proposed to account for ACI. Ganache requires a central server for frequency planning and does not support channel aggregation. In our proposed channel assignment schemes we consider channel bonding and aggregation.

VII. CONCLUSION

In this paper, we proposed GBA channel assignment algorithms to account for ACI, for a single-link as well as multiple links, in a DSA network. We adopted the guard-band reuse paradigm. The optimal assignment problem for a single-link was formulated as an SSP problem, and several exact and approximate algorithms were adapted to solve the channel assignment problem. We also obtained the optimal assignment for multiple-links. The MSSP problem was used as an initial step to derive the optimal joint assignment. To avoid the high complexity of the exact multiple-links assignment algorithm, a polynomial-time sequential assignment was used, where the greedy algorithm was adopted for each link. Our numerical results showed that the greedy sequential assignment achieves a comparable performance to optimal. The approximate greedy approach is still better than a previously proposed approach in [3].

REFERENCES

^[1] G. S. Uyanik, M. J. Abdel-Rahman, and M. Krunz, "Optimal guard-band-aware channel assignment with bonding and aggregation in multi-channel systems," in *Proceedings of the IEEE GLOBECOM Conference*, 2013.

- [2] V. Angelakis, S. Papadakis, V. Siris, and A. Traganitis, "Adjacent channel interference in 802.11a is harmful: Testbed validation of a simple quantification model," *IEEE Communications Magazine*, vol. 49, no. 3, pp. 160–166, March 2011.
- [3] H. Bany Salameh, M. Krunz, and D. Manzi, "An efficient guard-band-aware multi-channel spectrum sharing mechanism for dynamic access networks," in *Proc. of the IEEE GLOBECOM Conference*, Dec. 2011, pp. 1–5.
- [4] D. Skordoulis, Q. Ni, H.-H. Chen, A. Stephens, C. Liu, and A. Jamalipour, "IEEE 802.11n MAC frame aggregation mechanisms for next-generation high-throughput WLANs," *IEEE Transactions on Wireless Communications*, vol. 15, no. 1, pp. 40–47, Feb. 2008.
- [5] S. Martello and P. Toth, Knapsack problems: algorithms and computer implementations. John Wiley & Sons, 1990.
- [6] T. H. Cormen, C. Stein, R. L. Rivest, and C. E. Leiserson, Introduction to Algorithms, 2nd ed. McGraw-Hill Higher Education, 2001.
- [7] J. Kleinberg and E. Tardos, Algorithm Design. Addison-Wesley, 2005.
- [8] A. Muqattash and M. Krunz, "POWMAC: a single-channel power-control protocol for throughput enhancement in wireless ad hoc networks," *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 5, pp. 1067–1084, May 2005.
- [9] H. Bany Salameh and M. Krunz, "Adaptive power-controlled MAC protocols for improved throughput in hardware-constrained cognitive radio networks," *Ad Hoc Networks Journal*, vol. 9, no. 7, pp. 1127–1139, Sep. 2011.
- [10] S. Martello and P. Toth, "A bound and bound algorithm for the zero-one multiple knapsack problem," *Discrete Applied Mathematics*, vol. 3, no. 4, pp. 275–288, 1981.
- [11] D. Pisinger, "An exact algorithm for large multiple knapsack problems," *European Journal of Operational Research*, vol. 114, pp. 528–541, 1999.
- [12] D. Bertsimas and R. Demir, "An approximate dynamic programming approach to multidimensional knapsack problems," *Management Science*, vol. 48, no. 4, pp. 550–565, April 2002.
- [13] A. Caprara, H. Kellerer, and U. Pferschy, "A 3/4-approximation algorithm for multiple subset sum," *Journal of Heuristics*, vol. 9, pp. 99–111, 2003.
- [14] K. Nolan, P. Sutton, L. Doyle, T. Rondeau, B. Le, and C. Bostian, "Dynamic spectrum access and coexistence experiences involving two independently developed cognitive radio testbeds," in *Proc. of the IEEE DySPAN Conference*, April 2007, pp. 270–275.
- [15] O. Yu, E. Saric, and A. Li, "Dynamic control of open spectrum management," in *Proc. of the IEEE WCNC Conference*, March 2007, pp. 127–132.
- [16] L. Yang, B. Y. Zhao, and H. Zheng, "The spaces between us: Setting and maintaining boundaries in wireless spectrum access," in Proc. of the ACM MobiCom Conference, 2010, pp. 37–48.