

# Optimal Guard-band-aware Channel Assignment with Bonding and Aggregation in Multi-channel Systems

Gulnur Selda Uyanik  
Department of Computer Engineering  
Istanbul Technical University, Istanbul, Turkey  
seldauyanik@itu.edu.tr

Mohammad J. Abdel Rahman and Marwan Krunz  
Department of Electrical and Computer Engineering  
University of Arizona, Tucson, AZ 85721, USA  
{mjabdelrahman, krunz}@email.arizona.edu

**Abstract**—Channel assignment mechanisms in multi-channel wireless networks are often designed without accounting for adjacent-channel interference (ACI). To prevent such interference between different users in a network, guard-bands (GBs) are needed. Introducing GBs has a significant impact on spectrum efficiency. In this paper, we present a channel assignment mechanism that aims at maximizing the spectrum efficiency. More specifically, this mechanism attempts to minimize the amount of additional GB-related spectrum that is needed to accommodate a new link. Furthermore, inline with the IEEE 802.11n and the upcoming IEEE 802.11ac standards, our assignment mechanisms support channel bonding, and more generally, channel aggregation. We formulate the optimal ACI-aware channel assignment that maximizes the spectrum efficiency, while supporting channel bonding and aggregation, as a subset-sum problem. An exact exponential-time dynamic programming (DP) algorithm, a polynomial-time greedy heuristic, and an  $\epsilon$ -approximation are proposed and compared. Results reveal a considerable improvement in spectrum efficiency achieved by the proposed algorithms compared to the recent work in the literature.

**Index Terms**—Channel assignment, spectrum efficiency, subset-sum problem.

## I. INTRODUCTION

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 [1]. Most channel assignment algorithms in the literature do not account for ACI, and assume no power leakage between adjacent channels (see 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 two channels in a practical communication system. As discussed in [2], to mitigate ACI, *guard-bands* (GBs) are needed between adjacent channels that belong to different SUs, as shown in Figure 1(c).

However, introducing GBs constrains the spectrum efficiency. In [2], the authors studied two models for utilizing GBs in a dynamic spectrum access network: “GB reuse” and “no GB reuse”. According to the “GB reuse” model, GBs can be shared by two different (interfering) links. In contrast, in the “no GB reuse” model, two adjacent transmissions require

their own GBs. As explained in [2], the “GB reuse” model is suitable for discontinuous-orthogonal frequency division multiplexing (D-OFDM)-based systems, whereas the “no GB reuse” model is suitable for FDM-based systems. In this paper, we adopt the “GB reuse” model. This model is illustrated in Figure 1(c), where the same amount of GB is used between channels 1 and 2, irrespective of whether link  $B$  is active or not over channel 2. As shown later in this paper, the GB-aware (GBA) channel assignment algorithm in [2] for the “GB reuse” case does not achieve the maximum spectrum efficiency.

To support applications with high rate demands, the IEEE 802.11n and the upcoming IEEE 802.11ac standards have adopted channel bonding [3]. 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 data rates of the individual channels. This concept can be extended to non-adjacent frequency channels, and is referred to as *channel aggregation*. The channel assignment schemes proposed in this paper support both channel bonding and aggregation.

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

- We formulate and obtain the optimal GBA channel assignment for a single link, adopting the “GB reuse” setting. The per-link channel assignment problem is formulated as a subset-sum problem (SSP) [4].
- An exact exponential-time dynamic programming (DP) algorithm, a polynomial-time greedy heuristic, and an  $\epsilon$ -approximation are proposed and compared.
- We evaluate the exponential-time optimal assignment mechanism and compare them with polynomial-time heuristics and  $\epsilon$ -optimal approximation.

**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 optimal channel assignment is explained in Section III. Polynomial-time greedy and  $\epsilon$ -approximate algorithms are also presented in the same section. We evaluate our assignment algorithms in Section IV. Section V gives an overview of related work.

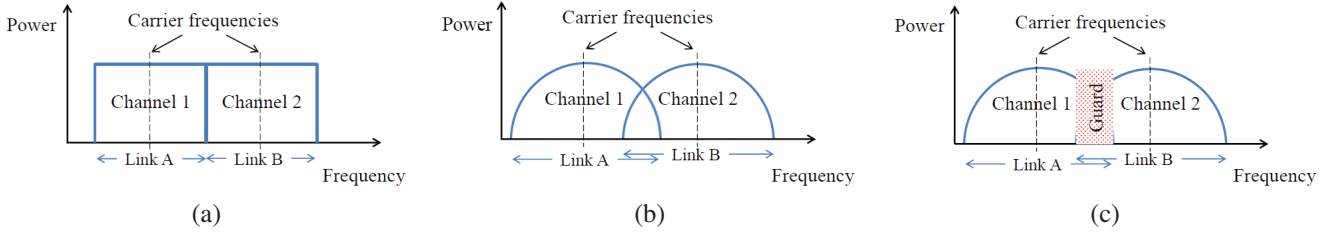


Fig. 1: Guard-band-aware channel assignment.

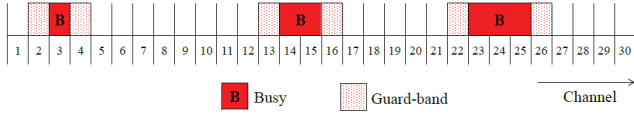


Fig. 2: Spectrum status (channel assignment) at a given time instance.

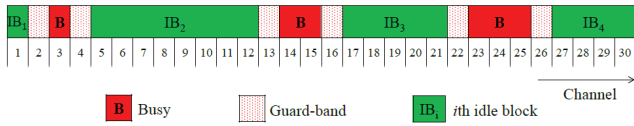


Fig. 3: Set of idle blocks for the spectrum map in Figure 2.

Finally, Section VI concludes the paper.

## II. PROBLEM STATEMENT

We consider a single-link in a single-hop wireless network operating using a set of channels  $\mathcal{M} = \{1, 2, \dots, M\}$ . Without loss of generality, we assume all channels to have the same bandwidth (in Hz). An available channel (i.e., not assigned) can be reserved as a GB, or assigned for data communication. All available channels support a common rate of  $r$  Mbps. The link has a rate demand of  $d_j \stackrel{\text{def}}{=} \alpha_j r$  Mbps, where  $\alpha_j$  is an integer between 1 and  $M$ . Given the current spectrum status, our objective is to satisfy the link demand while maximizing the spectrum efficiency (defined shortly). Figure 2 shows an example of an initial spectrum status having 6 busy channels and related GBs.

The spectrum efficiency ( $SE$ ) associated with a given channel assignment is defined as the fraction of the available spectrum that can be used for data communications. Let  $h_i, i \in \mathcal{M}$  be a binary variable, which equals one if channel  $i$  is assigned as a data channel, and zero otherwise. Let  $\eta_i$  be a binary variable indicating whether or not the  $i$ th channel is to be used as a GB channel. Then, the  $SE$  is defined as follows:

$$SE \stackrel{\text{def}}{=} \frac{\sum_{i=1}^M h_i}{\sum_{i=1}^M h_i + \sum_{i=1}^M \eta_i}. \quad (1)$$

In this paper, we consider the following problem.

**Problem 1.** Given a link with a rate demand of  $d = \alpha r$  Mbps and given the current status of the  $M$  channels, find the optimal GBA channel assignment for this link that maximizes  $SE$  while satisfying the demand  $d$ .

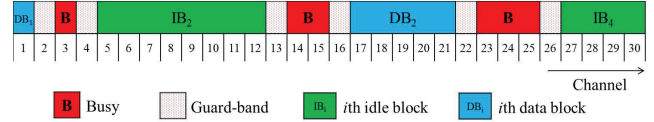


Fig. 4: Channel assignment with no additional GBs ( $d = 6$  Mbps).

## III. OPTIMAL GUARD-BAND-AWARE CHANNEL ASSIGNMENT

Consider Problem 1,  $SE$  can also be expressed as:

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

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 GBs (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 is grouped into a “idle frequency block”, as illustrated in Figure 3. 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 denote the rate supported by the  $i$ th block ( $IB_i$ ), where  $\beta_i$  is an integer between 1 and  $M$ . As justified in [5], we assume that one fixed-bandwidth GB channel on each side of a data transmission block is sufficient to prevent ACI, irrespective of the block size (it has been shown in [5] that the nearest subcarriers of a neighboring frequency block that is assigned to another transmission are the main source of interference to any demodulated subcarrier). We remark that, in general, the difference in the transmission powers of two frequency-adjacent links impacts the required amount of GB between them. However, in this paper, we assume that the power difference between adjacent links is small, and one GB on each side of the frequency block is sufficient to prevent ACI (for example, as in the case when a channel represents one OFDM subcarrier).

**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 GB. 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 case is exemplified in Figure 4, where  $d = 6$  Mbps can be met using

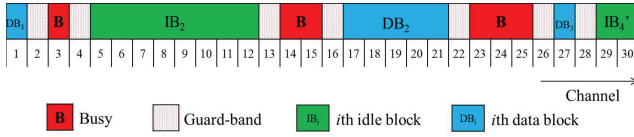


Fig. 5: Channel assignment with one additional GB ( $d = 7$  Mbps).

$B = \{IB_1, IB_3\}$  since  $R_1 = 1$  Mbps and  $R_3 = 5$  Mbps. In this case, the number of introduced GBs is zero (recall that we assume the GB reuse model). This is clearly an optimal assignment.

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

In this case, let  $B \subset \mathcal{N}$  be the largest set such that  $\sum_{i \in B} R_i < d$ . We assign the channels in  $B$  to this link. The unfulfilled  $d - \sum_{i \in B} R_i$  demand is then assigned to channels extracted from the beginning of one of the idle blocks in  $\mathcal{N} \setminus B$ . Consider, for example, the spectrum status in Figure 2. Suppose that we need to assign channels to a new link with  $d = 7$  Mbps. This demand cannot be exactly met by any combination of idle blocks. It can be satisfied using blocks  $IB_1$  and  $IB_3$ , of rates 1 Mbps and 5 Mbps, and one channel (channel 27) taken from the 4th idle block. As shown in Figure 5, this results in one additional GB, which is optimal because any other feasible assignment will introduce at least one GB (if there is an assignment with zero new GBs, then this contradicts the assumption made in case 2). Hence, the total number of introduced GBs is either zero or one. ■

Having established that assigning channels on a per-block basis results in the optimal  $SE$ , Problem 1 can be re-stated as follows: Given the set of idle blocks  $\mathcal{N}$ , 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) [4], where “items” correspond to idle frequency blocks and the weights of these items correspond to the rates supported by the idle blocks. Let  $x_i$  be a binary variable, which indicates whether or not idle block  $i$  is to be assigned to the underlying link. Then, the optimal GBA channel assignment can be stated as follows:

$$\underset{x_i, i \in \mathcal{N}}{\text{maximize}} \left\{ \mathcal{R}_s \stackrel{\text{def}}{=} \sum_{i=1}^N R_i x_i \right\} \quad (3)$$

$$\text{subject to } \sum_{i=1}^N R_i x_i \leq d \quad (4)$$

$$x_i \in \{0, 1\}, \forall i \in \mathcal{N}. \quad (5)$$

Let  $\mathcal{R}_s^*$  denote the optimal solution for the SSP problem. From (4),  $\mathcal{R}_s^* \leq d$ . When  $\mathcal{R}_s^* < d$ , we augment the SSP problem with a *post-processing phase* to make up for the demand “deficit”. As stated in Lemma 1 below, after executing the SSP problem each of the remaining idle blocks must support a data rate greater than  $d - \mathcal{R}_s^*$ . In the post-processing phase, we assign a portion of  $(d - \mathcal{R}_s^*)/r$  channels from any of the remaining idle blocks, starting from the beginning of the block. The assigned channels are followed by a GB, as shown in Figure 5.

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

**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 a 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,  $i \in \mathcal{C}$ , which 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, no additional GBs will be introduced, which is optimal.

Case 2:  $\mathcal{R}_s^* < d$ . In this case, by Lemma 1 and Theorem 1, one new GB will be introduced, which is also 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 GB. ■

SSP is an NP-complete problem [4], [6], [7]. In the following subsections, we present exact and approximate algorithms for solving it.

#### A. Dynamic Programming (DP)-based Exact Algorithm

The idea behind the DP-based approach is as follows. 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 a sub-instance of SSP, consisting of idle blocks  $IB_1, \dots, IB_{i-1}$  and rate demand  $\tilde{d}$ . If the rate supported by  $IB_i$  exceeds  $\tilde{d}$  (i.e.,  $R_i > \tilde{d}$ ), then  $IB_i$  will not be included in the optimal assignment. Otherwise,  $IB_i$  will be included in the optimal assignment if this results in a better solution 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  $IB_1, \dots, IB_i$  and demand  $\tilde{d}$ . Then, the recurrence relation is given by (6) (note that  $R_s^*(N, d) \stackrel{\text{def}}{=} R_s^*$ ).

The DP-based algorithm correctly computes the optimal SSP solution. It runs in  $\mathcal{O}(Nd)$  time [7].

---

#### Algorithm 1 DP-based Exact SSP Algorithm

---

- 1: **Input:**  $N, d, N$  by  $d + 1$  array  $M$
  - 2: **Initialize:**  $M[1, \tilde{d}] = 0, \forall \tilde{d} \in \{0, 1, \dots, d\}$
  - 3: **for**  $i = 1 : N$  **do**
  - 4:   **for**  $\tilde{d} = 0 : d$  **do**
  - 5:     **if**  $\tilde{d} < R_i$  **then**
  - 6:        $M[i, \tilde{d}] \leftarrow M[i - 1, \tilde{d}]$
  - 7:     **else**
  - 8:        $M[i, \tilde{d}] \leftarrow \max \left\{ M[i - 1, \tilde{d}], R_i + M[i - 1, \tilde{d} - R_i] \right\}$
  - 9:     **end if**
  - 10:   **end for**
  - 11: **end for**
  - 12: **Return:**  $M$
-

$$R_s^*(i, \tilde{d}) = \begin{cases} R_s^*(i-1, \tilde{d}), & \text{if } \tilde{d} < R_i \\ \max\left(R_s^*(i-1, \tilde{d}), R_s^*(i-1, \tilde{d} - R_i) + R_i\right), & \text{if } R_i \leq \tilde{d} \leq d. \end{cases} \quad (6)$$

### B. $\epsilon$ -approximate Algorithm

A pseudo-polynomial  $\epsilon$ -approximate algorithm for SSP was developed in [6] (see Algorithm 2). This algorithm selects the combination of idle blocks that results in a total rate that is closest to  $d$ . In the  $i$ th iteration (see the ‘for loop’ in line 3 of Algorithm 2), 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 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  $l_i, i = 1, \dots, N$  to reduce their lengths. TRIM removes an element with value  $a$  from the list if there is another element with value  $b$ , such that  $|a - b| \leq \delta$ . In [6],  $\delta$  is set to  $\epsilon/2N$ .

---

#### Algorithm 2 $\epsilon$ -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:**  $z^*$
- 

Note that the  $\epsilon$ -approximate algorithm may return idle blocks with rates less than or equal to the remaining unsatisfied demand, i.e., there is some probability that  $\exists$  an unassigned block  $i$  such that  $R_i \leq d - \sum_{j=1}^N R_j \eta_j$ . If  $R_i = d - \sum_{j=1}^N R_j \eta_j$ , then the  $\epsilon$ -approximate algorithm can be turned into optimal by searching for such blocks and including them in the assignment.

### C. Greedy Scheme

The greedy approach starts with the set of idle blocks, sorted in a descending order of their data rates. It passes through the sorted list and adds the idle blocks sequentially as long as the total rate does 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 one uses a sorting algorithm with complexity  $\mathcal{O}(N \log N)$  (e.g., merge sort algorithm).

In contrast to the other algorithms, in the  $\epsilon$ -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  $p > 0$ ,  $\exists$

TABLE I: Simulation parameters.

Parameter	Value
$d$	10 Mbps
$\epsilon$	0.2
$p_{\text{busy}}$	0.25
$L$	1
$M$	50

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  $\epsilon$ -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  $p_{\text{busy}}$ ). Therefore, the exponential-time exact algorithms can be used to retrieve the optimal assignment within a reasonable amount of time.

## IV. PERFORMANCE EVALUATION

The exact algorithm explained in Section III-A, the  $\epsilon$ -approximate algorithm described in Section III-B, and the greedy algorithm described in Section III-C are all implemented in C++. In addition to these algorithms, we implement the channel assignment scheme proposed in [2] in MATLAB. We refer to this scheme as ‘‘Choose all existing GBs’’ in the legends of the simulation 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 initially selects all existing GBs.

All algorithms are simulated in a common setup (shown in Table I) and using a common spectrum status.  $p_{\text{busy}}$  in Table I is the probability that a given channel is already assigned to another link (i.e., the probability that a channel is in state ‘busy’ in Figure 2). Our results are averaged over 50 runs, and the 95% confidence intervals are indicated in the figures.

Figure 6 depicts the spectrum efficiency vs.  $p_{\text{busy}}$  for various assignment schemes. As shown in this figure, SSP algorithms achieve higher  $SE$  than the scheme in [2]. This results from the fact that SSP-based assignment schemes are per-block schemes that inherently try to use existing GBs and avoid introducing any new guard-band, hence maximizing the  $SE$ . As  $p_{\text{busy}}$  increases, the number of existing GBs 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 [2] also improves with increasing  $p_{\text{busy}}$ ; because of the

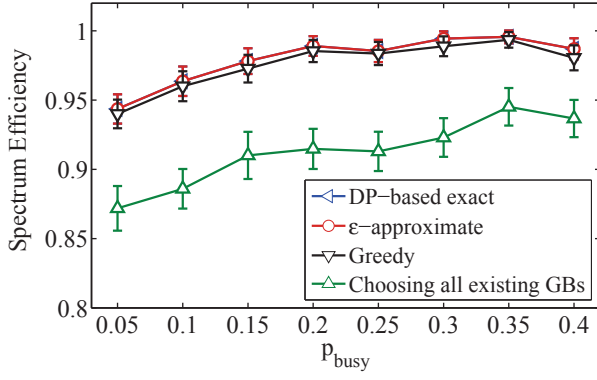


Fig. 6: Spectrum efficiency vs.  $p_{\text{busy}}$ .

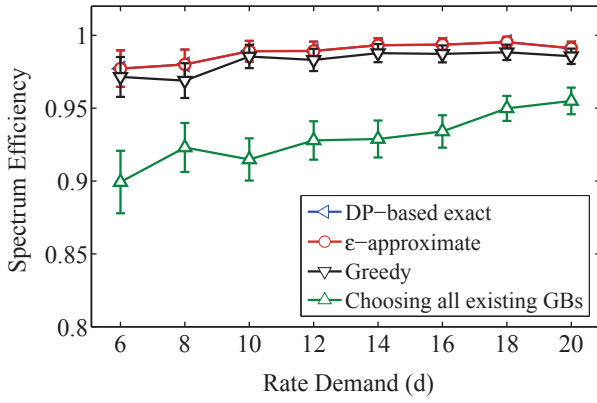


Fig. 7: Spectrum efficiency vs.  $d$ .

reduction in the sizes of idle blocks. The idle blocks selected by this scheme may not change with increasing  $p_{\text{busy}}$ , but the probability that the first and last channels of these blocks are existing GBs increases, which increases the  $SE$ . As shown in Figure 6, the  $\epsilon$ -approximate and greedy algorithms achieve a close  $SE$  to the optimal DP algorithm. Figure 7 shows the  $SE$  vs. the rate demand  $d$ . SSP-based assignment algorithms outperform the one in [2] for all values of  $d$ .

The number of introduced GBs is depicted in Figure 8 for different values of  $p_{\text{busy}}$ . The SSP-based algorithms introduce smaller numbers of GBs compared to [2]. As shown in Figure 8, the number of introduced GBs in the SSP-based algorithms is always less than or equal to one, which is consistent with the result in Theorem 2. As  $p_{\text{busy}}$  increases, the number of existing GBs increases because the number of busy blocks also increases. Since the SSP-based algorithms try to maximize the  $SE$ , they introduce a smaller number of GBs as  $p_{\text{busy}}$  increases, due to the increase in the number of existing GBs.

Figure 9 shows the number of introduced GBs for different values of  $d$ . SSP-based assignment algorithms again outperform the one in [2] for all values of  $d$ . The number of introduced GBs is not directly effected with the increase in  $d$ , because this depends more on the combination of the

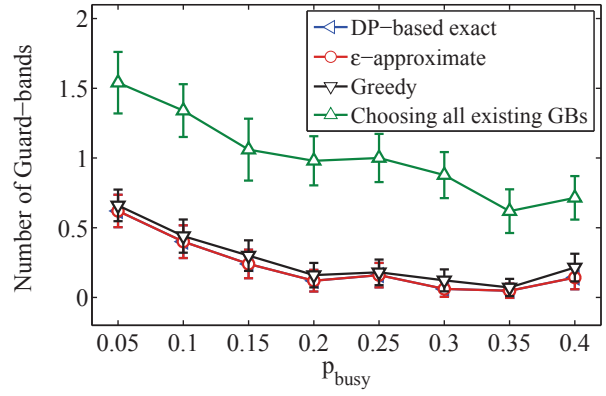


Fig. 8: Number of introduced GBs vs.  $p_{\text{busy}}$ .

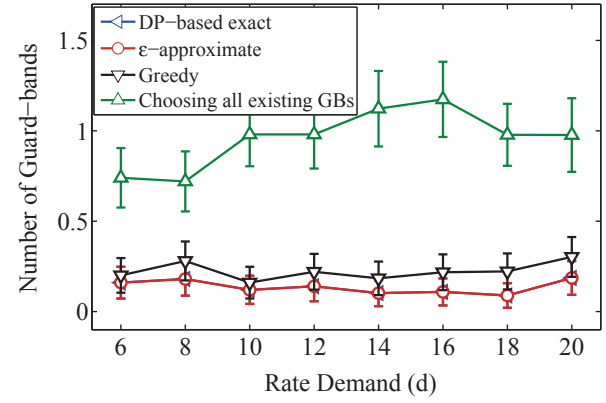


Fig. 9: Number of introduced GBs vs.  $d$ .

selected blocks.

Figure 10 shows the fraction of the 50 runs that report infeasibility for different values of  $p_{\text{busy}}$ . As shown in Figure 10, the infeasibility ratio of all considered schemes can reach up to 0.45 when  $p_{\text{busy}} = 0.4$ . When the channel becomes less available (i.e.,  $p_{\text{busy}}$  increases), the chance of not meeting the link demand increases, irrespective of the used assignment scheme. The infeasibility ratio is also shown for various values of  $d$  in Figure 11. As shown in the figure, for a given value of  $p_{\text{busy}}$ , as  $d$  increases, the infeasibility ratio increases.

It is to be noted that the greedy algorithms performs very close to the exact algorithm (which have an exponential-time complexity).

## V. RELATED WORK

Most of the existing channel assignment schemes in wireless networks do not account for ACI. The GBA channel assignment problem is considered in [2]. The GBA channel assignment problem is considered in [2] under two GBs sharing paradigms: GB reuse and no GB reuse. In this paper, we have considered the GB reuse paradigm. The proposed assignment scheme in [2] for the GB reuse case is not optimal in terms of the spectrum efficiency. The scheme in [2] was

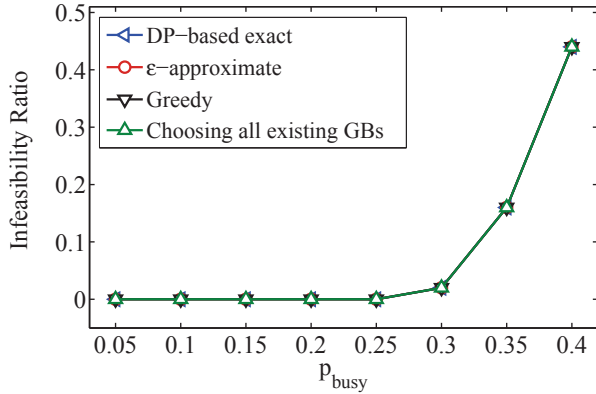


Fig. 10: Infeasibility ratio vs.  $p_{\text{busy}}$ .

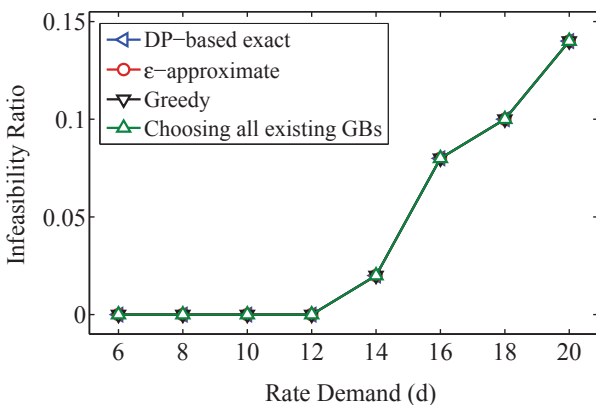


Fig. 11: Infeasibility ratio vs.  $d$ .

designed to minimize the number of idle blocks assigned to a given link, aiming by that to minimize the number of GBs. The motivation behind this was that each idle block requires two GBs. 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 [8], the amount of required GBs 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 [9], a centralized adaptive GB 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.

Resource allocation schemes with channel bonding have been considered in [10]–[12] for OFDMA systems. However, these schemes do not account for ACI by introducing additional GBs.

## VI. CONCLUSION

In this paper, we proposed GBA channel assignment algorithms that account for ACI in multi-channel wireless networks

with channel bonding/aggregation. Single-link assignments were considered under guard band reuse paradigm. The optimal assignment problem was formulated as an SSP problem with a post processing phase. Exact, approximate, and heuristic solutions were presented. Our numerical results showed that the greedy assignment achieves a near-optimal performance. Moreover, the approximate greedy approach is better than a previously proposed approach in [2], [13].

## VII. ACKNOWLEDGMENT

This research was supported in part by NSF (under grants CNS-1016943 and CNS-0904681, IIP-0832238, IIP-1231043), Raytheon, and the “Connection One” center. Any opinions, findings, conclusions, or recommendations expressed in this paper are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

## REFERENCES

- [1] 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.
- [2] H. Bany Salameh, M. Krunz, and D. Manzi, “Spectrum bonding and aggregation with guard-band awareness in cognitive radio networks,” *to appear in the IEEE Transactions on Mobile Computing*.
- [3] 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.
- [4] S. Martello and P. Toth, *Knapsack problems: algorithms and computer implementations*. John Wiley & Sons, 1990.
- [5] I. Trigui, M. Siala, and H. Boujemaa, “Optimized pulse shaping for OFDM multi-user communications over doubly dispersive channels,” in *Proceedings of the International Symposium on Signal Processing and Its Applications*, 2007, pp. 1–4.
- [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] O. Yu, E. Saric, and A. Li, “Dynamic control of open spectrum management,” in *Proceedings of the IEEE WCNC Conference*, March 2007, pp. 127–132.
- [9] L. Yang, B. Y. Zhao, and H. Zheng, “The spaces between us: Setting and maintaining boundaries in wireless spectrum access,” in *Proceedings of the ACM MobiCom Conference*, 2010, pp. 37–48.
- [10] H. Zhu and J. Wang, “Chunk-based resource allocation in OFDMA systems - part I: chunk allocation,” *IEEE Transactions on Communications*, vol. 57, no. 9, pp. 2734–2744, 2009.
- [11] H. Zhu and J. Wang, “Chunk-based resource allocation in OFDMA systems - Part II: Joint chunk, power and bit allocation,” *IEEE Transactions on Communications*, vol. 60, no. 2, pp. 499–509, 2012.
- [12] L. Lei, V. Angelakis, and D. Yuan, “Performance analysis of chunk-based resource allocation in wireless OFDMA systems,” in *IEEE International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*, 2012, pp. 90–94.
- [13] H. Bany Salameh, M. Krunz, and D. Manzi, “An efficient guard-band-aware multi-channel spectrum sharing mechanism for dynamic access networks,” in *Proceedings of the IEEE GLOBECOM Conference*, Dec. 2011, pp. 1–5.