Les Cahiers du GERAD ISSN: 0711–2440

Cooperative spectrum sensing scheduling optimization in multi-channel dynamic spectrum access networks

A. Azarfar, C.-H. Liu, J.-F. Frigon,
B. Sansò, D. Cabric

G–2014–95

November 2014

GERAD HEC Montréal
3000, chemin de la Côte-Sainte-Catherine
Montréal (Québec) Canada H3T 2A7

Tél. : 514 340-6053
Téléc. : 514 340-5665
info@gerad.ca
www.gerad.ca

Cooperative spectrum sensing scheduling optimization in multi-channel dynamic spectrum access networks

Arash Azarfar[a]
Chun-Hao Liu[b]
Jean-François Frigon[a]
Brunilde Sansò[a]
Danijela Cabric[b]

[a] *GERAD & Department of Electrical Engineering, Polytechnique Montréal, Montréal (Québec) Canada, H3C 3A7*

[b] *Department of Electrical Engineering, University of California, Los Angeles, CA 90095-1594, USA*

arash.azarfar@polymtl.ca
liuch37@ee.ucla.edu
j-f.frigon@polymtl.ca
brunilde.sanso@polymtl.ca
danijela@ee.ucla.edu

November 2014

Les Cahiers du GERAD
G–2014–95

Abstract: Dynamic spectrum access (DSA) for secondary networks improves the spectrum utilization by finding spectrum opportunities and exploiting them efficiently. A key factor to design a DSA network is the spectrum sensing algorithms for multiple channels with multiple users. Multi-user cooperative channel sensing reduces the sensing time, thus increasing the transmission throughput. However, in a multi-channel system, the problem becomes more complex since a sensing schedule, indicating to each user the channel that it must sense at different sensing moments, must be created to optimize system performance. In this paper, we first propose a general sensing strategy to schedule the users according to network parameters. We propose three sensing strategies, and within each one of them several solutions striking a balance between throughput performance, memory usage, and computational complexity are proposed. In addition, we show that the proposed sequential sensing strategy is the one to be preferred when the sensing time is small, the number of channels is large, and the number of users is small. For all the other cases, the parallel sensing strategy is recommended in terms of throughput performance. We also show that a proposed hybrid sequential-parallel sensing strategy achieves the best performance in all scenarios at the cost of extra complexity.

Acknowledgments: This research was supported by the National Science Foundation under CNS grant 1149981 and FQRNT-CREER program for international internship application No. 179776.

1 Introduction

In order to increase current spectrum utilization, it has been proposed that secondary (unlicensed) users (SUs) could efficiently exploit spectrum vacancies that are normally licensed to primary users (PUs) in either temporal, frequency, or space domain in dynamic spectrum access (DSA) networks [1]. The two-stage processing is a well-known DSA strategy for SUs, i.e., sense and transmit [2]. The secondary users first sense a licensed channel, and if the channel is not occupied by primary users, then SUs can transmit on it, otherwise they need to sense other channels to find possible transmission. The sensing strategy is important for the performance of the secondary users since if licensed channels are sensed in a shorter time, the secondary users will have a longer access opportunity on the vacant channels, which results in a larger throughput.

The sensing strategy for secondary users is therefore an important issue that needs to be addressed. That strategy determines how to assign sensing processes to secondary users to optimize an objective metric, such as throughput, delay or energy consumption. In other words, a sensing strategy provides a time schedule (e.g., sensing order) to sense channels as well as a process schedule for SUs, so that a decision is made as at which time instant each SU should sense which channels. Naturally, if multiple SUs are assigned to sense the same channel at the same time, a cooperative sensing is pursued for this channel, which can increase sensing accuracy and may reduce sensing time [3].

Sensing strategies have so far been mostly investigated for sequential sensing order optimization and acquiring the stopping time when channels are sensed one after the other. To the best of our knowledge, [4] is the first to introduce the concept of sensing order. The authors proposed a semi-Markov chain to model channel occupancy, and channels are sensed in the decreasing order of the probability of being idle. [5] and [6] also take channel capacity and sensing time into account to derive the optimal sensing order. A multi-user network is investigated in [7] and [8] where channels are being sensed in parallel, but the only parameter engaged in decision making is channel occupancy in channels with the same capacity without considering the impact of cooperative sensing.

In addition to the sequential sensing order optimization mentioned above, this paper is the first to focus on a more general sensing strategy design, addressing the compromise that exists between assigning less users to sense a large number of channels in parallel versus the benefits of assigning multiple users to cooperatively sense the same channel but sensing multiple channels sequentially. First, we propose several sensing strategies to maximize system throughput, and next we investigate the tradeoff among these strategies under various circumstances. The contribution of this paper is twofold:

1. We introduce the general problem of sensing strategy optimization for optimal allocation of secondary users to maximize system throughput.
2. Three kinds of specific sensing strategies, i.e., sequential, parallel, and sequential-parallel strategies are proposed and compared.

The reminder of the paper is organized as follows. The system model and problem formulation are provided in Section 2. The general and all the particular sensing strategies are presented and analyzed in Section 3. Numerical results are provided and discussed in Section 4. Finally, Section 5 concludes the paper.

2 System model and problem formulation

We consider a DSA network with N secondary users and M channels. Primary users are assumed to transmit synchronously on the channels in a time-slotted fashion with a slot duration equal to T [5]. At the beginning of each time slot, the central network controller determines the sensing strategy for users to maximize the total expected spectrum opportunities for transmission. A spectrum strategy includes the time schedule (e.g., sensing order) and job schedule (which users sense which channels). After the users finish sensing a channel, sensing results will be merged and sent to the central controller. The final decision for the transmission of the users is made by the central controller depending on whether the channel is available or not.

We adopt a simplified model with non-fading channels where the average received signal-to-noise ratio (SNR) is the same for any potential secondary receivers. We thus define the capacity of the i-th channel as

$C_i = B_i \log_2(1+\Gamma)$, where B_i is the bandwidth of the i-th channel, $1 \le i \le M$, and Γ is the received SNR. The probability for the i-th channel being occupied by primary users is assumed to be known at the central controller as u_i, where u_i can be estimated or measured efficiently [5,9] in the training phase.

In this paper, our focus is on the spectrum opportunity detection part. The transmission part (which users actually transmit in a found spectrum opportunity) is out of the scope of this paper and it remains as our future work. We therefore regard the users as *sensors* and consider detected available channels as spectrum opportunities. The expected throughput from those spectrum opportunities thus represents an upper bound on the actual network throughput when transmission assignment of users will be taken into account.

2.1 Cooperative spectrum sensing

Cooperative sensing is a well-known solution to enhance sensing performance [10]. The reason is that as the collective decision is made with several individual sensing results, the requirement of sensing accuracy for each individual user can be lowered, hence the sensing time can be reduced. In a time slotted DSA network, since the status of the channel does not change during one time slot, minimizing the sensing time for the channel implies increasing the transmission throughput [2] for SUs.

2.1.1 Primary user detection

Consider a secondary receiver that needs to detect primary users sending pilot signals on a particular channel [3]. Let τ be the sensing time and assume that the receiver's sampling frequency is f_s such that $N_s = \tau \times f_s$ samples are gathered to make the decision of whether a channel is occupied by a primary user. The minimum sensing time required to satisfy the given detection quality under additive white Gaussian noise (AWGN) channel by the optimal detector, i.e., the matched filter is equal to [3]:

$$\tau = \frac{\left[Q^{-1}(P_f) - Q^{-1}(P_d)\right]^2}{\gamma f_s}, \tag{1}$$

where $\gamma = \frac{P}{\sigma_w^2}$ is the detected SNR, P is the transmission power of the primary user, σ_w^2 is the noise variance, and P_d is the probability of detection, defined as the probability of detecting the primary user when it is present. Finally, P_f is the probability of false-alarm defined as the probability of wrongly finding the channel occupied when it is actually vacant.

2.1.2 Fusion rules

Sensing results reported by different users may be combined in different manners, known as fusion rules [10]. In what follows, we discuss *OR* and *AND* fusion rules because they are commonly used in the literature and also they provide bounds for the more general k-*out-of*-N fusion rule. Assume all N users are homogeneous, i.e., they have the same P_d and P_f. Thus, the cumulative probability of detection and false-alarm are given as $Q_d = 1-(1-P_d)^N$, and $Q_f = 1-(1-P_f)^N$ for the *OR* fusion rule respectively, and as $Q_d = P_d^N$, and $Q_f = P_f^N$ for the *AND* fusion rule respectively. By plugging the above P_f and P_d equations into (1), we can derive the minimum cooperatively sensing time by N homogeneous users. Throughout the paper, we also define $\tau_{m,n}$ as the cooperative sensing time of channel m by n users.

2.2 Problem formulation

We define the beginning of a time slot as a reference point $t = 0$, and the elapsed time when the sensing process for channel i is finished as $T_I^{(i)}$. Note that $T_I^{(i)}$ depends on the sensing order and user allocation schemes, and the sensing time for channel i depends on the number of users allocated to it. As illustrated in Figure 1(a), if channel i is found available, it is a spectrum opportunity with duration $T - T_I^{(i)}$. The expected throughput obtainable from the spectrum opportunity of channel i is thus equal to $C_i(1-u_i)(T - T_I^{(i)})$. The elapsed time for a channel which is not sensed can be assumed to be T (no throughput gain). Our objective is to maximize the total expected normalized throughput R from all channel spectrum opportunities by deciding the optimal sensing strategy, i.e.,

$$\max_{\mathcal{A}} \mathbb{E}\{R(\mathcal{A})\} = \sum_{i=1}^{M} \frac{(T - T_I^{(i)}(\mathcal{A}))C_i(1-u_i)}{T}, \tag{2}$$

where $\mathbb{E}\{\cdot\}$ is the expectation operation and $\mathcal{A}$ is a sensing strategy. Note for any channel i, $T_I^{(i)}(.)$ is a function of the sensing strategy $\mathcal{A}$.

3 Spectrum sensing strategies

A sensing strategy determines the order in which the channels are sensed and the number of users which sense a channel cooperatively. As discussed, $\tau_{m,n}$ is defined as the sensing time of channel m to satisfy a desired sensing performance when the channel is sensed cooperatively by n users. In addition, the sensing strategy should also give the timing schedule of sensing different channels. The optimal sensing strategy, which is called *general strategy* in this paper, includes any possible strategy to sense a set of channels. For instance, consider the scenario in Figure 1(b) with 4 channels and 3 users. At the slot start, channel 1, 2, and 4 are sensed respectively by users 1, 2, and 3. To sense channel 3, there are 3 possibilities: i) User 1 solely senses channel 3 when it finishes its job sooner than user 2 and 3. Hence sensing channel 3 is finished at $T_I^{(3)} = \tau_{1,1} + \tau_{3,1}$; ii) User 1 waits for user 3 to finish its job and then they cooperatively sense channel 3 and $T_I^{(3)} = \tau_{4,1} + \tau_{3,2}$; iii) Both users 1 and 3 wait also for user 2 and then sense channel 3 cooperatively (by three users) and $T_I^{(3)} = \tau_{2,1} + \tau_{3,3}$. As shown in the figure, it is assumed that option (ii) is the optimal solution. However, due to the large number of possible solutions, targeting the general strategy is highly cumbersome and it can not be solved efficiently in time. Therefore, in the following sections, we propose three specific strategies with particular sensing strategy structures. Each strategy can be considered as a sub-optimal scheme for the general strategy.

The three proposed strategies are: i) a *sequential strategy* where channels are sensed cooperatively by all N users in a sequential manner, ii) a *parallel strategy* where channels are cooperatively sensed in parallel with a subset of users, and iii) a mixture of sequential and parallel called *sequential-parallel strategy* where different sets of channels are sensed in parallel, but channels in each set are sensed in a sequential manner. An example for each strategy is provided in Figure 1.

3.1 Sequential sensing strategy

The sequential strategy with cooperative sensing was first discussed in [5]. We briefly review this strategy and show an example in Figure 1(d). Given a list of users and channels, it is assumed that all users cooperate to sense the channels one by one. Therefore, the sensing time of any channel m by N users is given by $\tau_{m,N}$. The sensing order for all users is defined as $\mathcal{A} = (a_1, a_2, \ldots, a_M)$ which is a permutation of $(1, 2, \ldots, M)$. The expected spectrum opportunity throughput can then be re-written from (2) as

$$\max_{\mathcal{A}} \mathbb{E}\{R(\mathcal{A})\} = \sum_{i=1}^{M} \frac{(T - T_I^{(a_i)})C_{a_i}(1-u_{a_i})}{T}, \tag{3}$$

where $T_I^{(a_i)} = \sum_{j=1}^{i} \tau_{a_j,N}$. Naturally if $T_I^{(a_i)} > T$, the throughput is zero for this channel. In [5], it is proved that the optimal sensing order is found by sorting the channels in decreasing order of $\frac{C_j(1-u_j)}{\tau_{j,N}}, j = 1, \ldots, M$.

3.2 Parallel sensing strategy

In this strategy, channels are sensed in parallel and the central controller makes the decision on the number of users who should sense each channel. Intuitively, when no user is assigned to a channel, the channel is not sensed and no spectrum opportunity throughput is available for this channel. Each user is assumed to

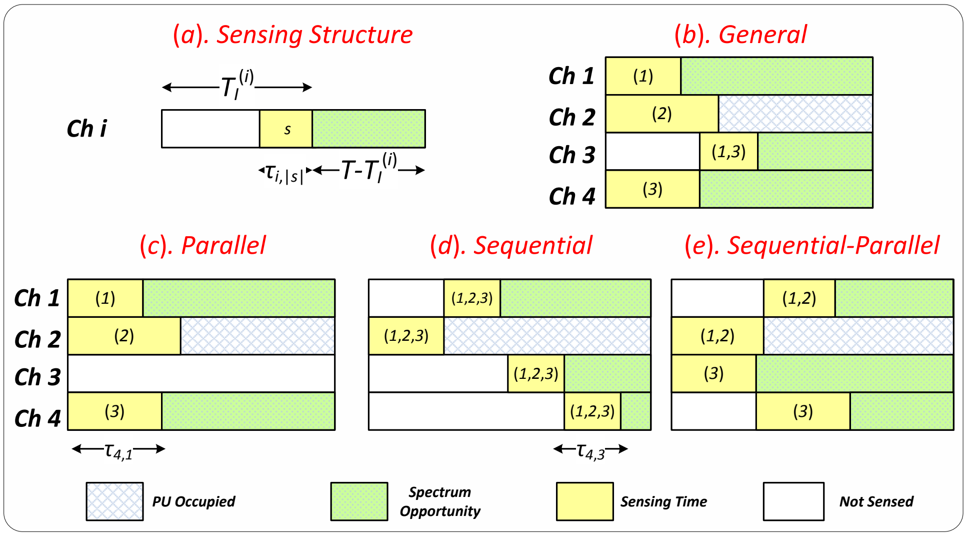


Figure 1: (a) Sensing structure when channel i is sensed by a subset s with $|s|$ users. (b) A general sensing strategy where user 1 waits for user 3 to finish its job and then they sense channel 3 cooperatively. (c) An example of parallel strategy where channel 3 is not sensed. (d) An example of sequential strategy with the channel (Ch) sensing order (Ch 2,Ch 1,Ch 3,Ch 4) by all users. (e) Sequential-parallel strategy where (Ch 2, Ch 1) are sensed sequentially by users 1 and 2 and in parallel for (Ch 3, Ch 4) by user 3.

sense one channel. An example of this strategy is illustrated in Figure 1(c). The optimization problem can be represented as

$$\max_{\mathcal{A}} \sum_{i=1}^{M} \frac{(T-\tau_{i,k_i})C_i(1-u_i)}{T}, \text{ s.t. } \sum_{i=1}^{M} k_i = N, \tag{4}$$

where k_i is the number of users assigned to channel i and $\mathcal{A} = (k_1, k_2, \ldots, k_M)$. This is a classical integer programming problem. In the following, we first discuss a dynamic programming (DP) solution, and then a heuristic solution. At the end, the condition to have an integer assignment is relaxed and a relaxed optimization problem is discussed.

3.2.1 Dynamic programming

As a resource allocation problem, we propose the following dynamic programming (DP) solution to find the optimal assignment [11]. The *stage* of the DP is the channel number. Thus, starting from channel 1, we must decide at each stage, how many of the remaining users should be assigned to the particular channel considered. The decision variable is the number of users, the instantaneous payoff is the throughput which may be obtained from this channel, and the value function $v_k(n)$ is the total expected throughput which can be obtained from the optimal assignment from now on when k channels and n users remain. Transition possibilities naturally depend on the remaining number of users. Then, the Bellman equation can be written as

$$v_k(n) = \max_{0\le j\le n} \left\{ \frac{(T-\tau_{k,j})C_k(1-u_k)}{T} + v_{k-1}(n-j) \right\}. \tag{5}$$

The terminal conditions are when no users remain to be assigned, i.e., $v_k(0) = 0$, $\forall k$. We thus have $v_1(n) = \frac{(T-\tau_{M,n})C_M(1-u_M)}{T}$, meaning that in the last stage, any remaining users should be assigned to the last channel (channel M). The DP is finite, so it is solved by backward induction [11], and the maximal throughput is equal to $v_M(N)$. Note that since the channels are sensed in parallel, sensing order and the order of channels in the DP are irrelevant.

In the proposed DP solution, we choose the users as resource to be assigned to channels since the DP has a lower runtime complexity compared to the case where channels are assigned to users. Consider an optimal assignment $\mathcal{A}^o = (k_1, \ldots, k_M)$ with a given number of users. Assume one new user is assigned to channel i to achieve an optimal allocation, for any other channel j, we should thus have the condition $\mathbb{E}\{R(k_1, \ldots, k_i + 1, \ldots, k_M)\} \geq \mathbb{E}\{R(k_1, \ldots, k_j + 1, \ldots, k_M)\}$, $\forall j \neq i$, and it can be simplified as $(\tau_{i,k_i} - \tau_{i,k_i+1})C_i(1 - u_i) \geq (\tau_{i,k_j} - \tau_{i,k_j+1})C_j(1 - u_j)$. Therefore, each new user is added to a channel i with currently k_i assigned users which has the highest $(\tau_{i,k_i} - \tau_{i,k_i+1})C_i(1 - u_i)$ value. The DP algorithm for the parallel sensing strategy is presented in Algorithm 1.

Algorithm 1 Pseudo Algorithm for Parallel DP Solution

```
for m = 1 : M do
    for n = 1 : N do
        R(m, n) = (T − τ_{m,n}) C_m (1 − u_m)
    end for
end for
R(:, 0) = 0
A = 0
while N > 0 do
    m* = arg max_m ΔR = R(m, k_m + 1) − R(m, k_m), 1 ≤ m ≤ M
    k_{m*} = k_{m*} + 1
    N = N − 1
end while
```

3.2.2 Greedy heuristic

The high execution complexity of the DP solution prompts the need to have a low-complexity heuristic. A simple yet efficient solution, as shown in Section 4, is a greedy heuristic that puts more users on a channel with a higher probability of availability $1 - u_i$. We thus propose

$$k_i = \left[\frac{1 - u_i}{\sum_{j=1}^{M} (1 - u_j)} \right], \tag{6}$$

where $[\cdot]$ is the rounding operation. Since the sum of k_i values derived from (6) is not necessarily M, if $N - \sum_{j=1}^{M} k_j > 0$, the remaining $N - \sum_{j=1}^{M} k_j$ users are assigned to the channel with the maximum $(1 - u_j)$, otherwise $\sum_{j=1}^{M} k_j - N$ additional users are eliminated from the channel with the highest probability of availability $1 - u_j$.

3.2.3 Constraint relaxation

In this section, we propose to relax the constraint of the optimization problem in (4) where k_i values are not necessary integer. This helps us to derive a bound for the parallel strategy. It can be easily shown that the objective function $\mathbb{E}\{R(\mathcal{A})\}$ is not a simple concave function, yielding to a non-convex optimization programming solution that, given the reduced size, it can still be optimally solved by brute-force search.

3.3 Sequential-parallel strategy

We propose in this section a mixed strategy named sequential-parallel. As can be seen in the example provided in Figure 1(e), both users and channels are clustered into disjoint groups. The decision to be made is which user subset should be assigned to which channel subset to perform sequential cooperative sensing on this channel subset. In other words, within each channel subset, a sequential strategy is followed while different channel subsets are sensed in parallel.

We define a function $R_s(\mathcal{S}_m, n)$ which is the maximum expected throughput obtainable from sequentially sensing subset $\mathcal{S}_m$ of channels cooperatively by n users. From Section 3.1, we already have the optimal sequential strategy within one channel subset. With this type of structure, the throughput maximization problem is indeed a Knapsack problem [12] where we are looking for the best 2-tuples $(\mathcal{S}_m, n)$ to put in the knapsack. In the following, a dynamic programming model and a greedy heuristic are proposed to solve this problem.

3.3.1 Dynamic programming

Given the function R_s, the state variable in the DP equation is represented by $(\mathcal{S}, n)$, where $\mathcal{S}$ is a subset of channels, not sensed yet, and n is the number of remaining users, not assigned to any channel set. The decision is one of the subsets of $\mathcal{S}$ and the number of users assigned to it. Therefore, the total number of possible actions is equal to $2^{|\mathcal{S}|}(n+1)$. The Bellman equation can be given by

$$v(\mathcal{S}, n) = \max_{0 \le j \le n, \mathcal{X} \subseteq \mathcal{S}} \left\{ R_s(\mathcal{X}, j) + v(\mathcal{S} - \mathcal{X}, n - j) \right\}, \tag{7}$$

where $\mathcal{X}$ is the decision variable which is a subset of $\mathcal{S}$. The DP model is of infinite-horizon, so it can be solved by value iteration [11]. As soon as we reach any state with $v(\emptyset, n)$ or $v(\mathcal{S}, 0)$, the ongoing payoff is zero and the solution is terminated.

3.3.2 Greedy heuristic

Similar to the classical Knapsack problem, the greedy approach starts with the 2-tuple whose ratio of throughput versus the number of users is maximum. When a channel subset and the number of users to sense it are decided, the algorithm is continued for the remaining users and channels. The greedy algorithm can be found in Algorithm 2.

Algorithm 2 Pseudo Algorithm for Sequential-Parallel Greedy Heuristic Solution

```
while N > 0 do
    Select 2-tuple (S*, n*) with maximum R_s(S_m, n)/n
    Remove all entries (S_m, n) if S_m ∩ S* ≠ ∅
    Remove all entries (S_m, n) if n > N − n*
    N = N − n*
end while
```

3.4 Memory usage and computational complexity discussion

It is not possible to solve the throughput optimization problem in the general mixed sensing strategy in polynomial time, since all permutations of M channels along with all ways to divide N users among M channels need to be considered. In terms of the memory space complexity, the maximum memory space required for it is $\mathcal{O}(2^M 2^N)$ to keep the sensing time of any subset of channels by any subset of users, where $\mathcal{O}(\cdot)$ is the big O notation. For the other proposed strategies, the memory space and computational complexity are summarized in Table 1. Note that for the sequential-parallel strategy, DP has more lower order terms than heuristic in computation. In addition, in calculating the computation complexity, the execution time to fill the required data structures is not considered to avoid repetition.

Table 1: Memory space and computational complexity of sensing strategies.

Strategy	Memory	Computation
Sequential	$\mathcal{O}(M)$	$\mathcal{O}(M^2)$
Parallel-DP	$\mathcal{O}(MN)$	$\mathcal{O}(MN^2)/\mathcal{O}(M^2N)$
Parallel-Heuristic	$\mathcal{O}(1)$	$\mathcal{O}(M)$
Sequential-Parallel-DP	$\mathcal{O}(2^M N)$	$\mathcal{O}(2^{2M} N^2)$
Sequential-Parallel-Heuristic	$\mathcal{O}(2^M N)$	$\mathcal{O}(2^{2M} N^2)$

4 Numerical results

In this section, the numerical results for the throughput under different strategies are presented and discussed. In what follows, we choose the sampling frequency as the Nyquist frequency which equals to two times the corresponding channel bandwidth for cooperative sensing.

4.1 Throughput comparisons for all strategies

In Figure 2 it can be observed that, among the parallel strategies, the constraint relaxation achieves the largest throughput because the solution in (4) is not necessarily integer hence it provides an upper bound for the parallel strategies. Second, the proposed greedy heuristic strategy performs almost as good as the others when the number of users is high. However, since we have the rounding operation, this strategy is unstable, which is shown by a performance gap on $N = 9$. Third, the sequential strategy outperforms the parallel strategies since the sensing time is small for each channel and the number of channels is large compared to the number of users so that the sequential strategy would not waste as many spectrum opportunities as the parallel strategies. Finally, the sequential-parallel strategy performs better than the others while our proposed greedy heuristic can be a good candidate solution to achieve the optimal solution. Note that in Figure 2 we consider the OR rule, but for the AND rule we also have similar curves.

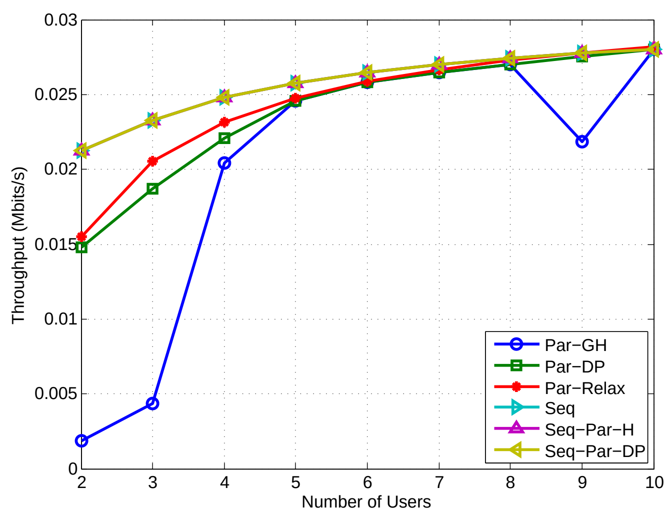
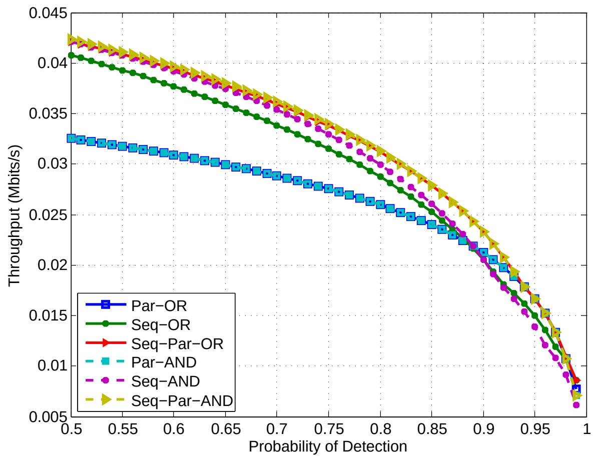


Figure 2: Performance comparison for parallel strategy with greedy heuristic (Par-GH), dynamic programming (Par-DP), constraint relaxation (Par-Relax), sequential strategy (Seq), sequential-parallel strategy with greedy heuristic (Seq-Par-GH), and dynamic programming (Seq-Par-DP) under OR fusion rule versus the number of users. Simulation parameters are $T = 1$ ms, $\Gamma = 10$ dB, $Q_d = 0.9$, $Q_f = 0.25$, $M = 6$, $\gamma = 5$ dB, $B = 0.5 f_s = (1, 1.5, 2, 2.5, 3, 5)$ kHz, and $u = (0.1, 0.2, 0.3, 0.4, 0.5, 0.3)$ for each channel.

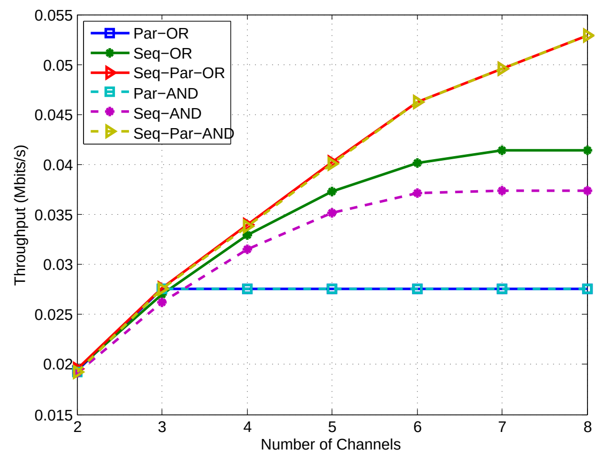
4.2 Throughput comparisons for parallel and sequential strategies

To have a better insight on all the strategies, we vary different parameters to show the impact on the optimal throughput in Figure 3. We assume that all channels have the same parameters to simplify the results interpretation.

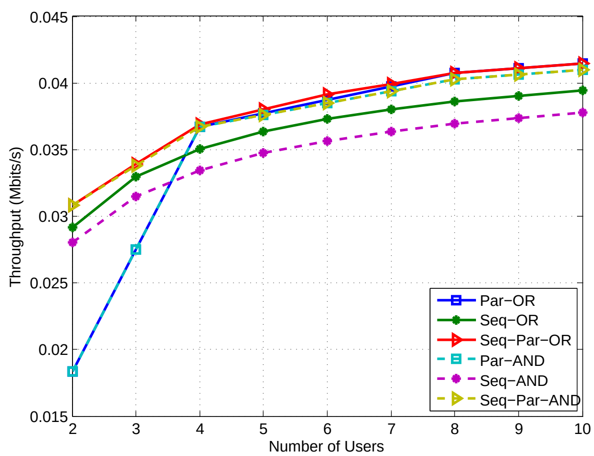
In Figure 3(a), we first adapt various Q_d. To support a higher target Q_d, the sensing time must be increased. We thus observe a performance degradation for all strategies. When the sensing time is short, the sequential strategy guarantees that all channels will be sensed. On the other hand, the parallel strategy senses a maximum of N channels which is less than M. Hence, it can be seen that the parallel strategy performs better than the sequential when the sensing time is long becuase the spectrum opportunities in the last channels of the sequential schedule will be very short or null. The sequential-parallel strategy naturally



(a) Performance comparison versus probability of detection Q_d. Simulation parameters are $Q_f = 0.1$, $N = 3$, $M = 4$, $\gamma = 5$ dB, $f_s = 2B = 5$ kHz, $u = 0.3$, $T = 1$ ms, and $\Gamma = 10$ dB.



(b) Performance comparison versus number of channels. Simulation parameters are $Q_d = 0.9$, $Q_f = 0.25$, $N = 3$, $\gamma = 5$ dB, $f_s = 2B = 5$ kHz, $u = 0.3$, $T = 1$ ms, and $\Gamma = 10$ dB.



(c) Performance comparison versus number of users. Simulation parameters are $Q_d = 0.9$, $Q_f = 0.25$, $M = 4$, $\gamma = 5$ dB, $f_s = 2B = 5$ kHz, $u = 0.3$, $T = 1$ ms, and $\Gamma = 10$ dB.

Figure 3: Throughput comparisons for using the *OR* fusion rule of parallel (Par-OR), sequential (Seq-OR), sequential-parallel (Seq-Par-OR), and the *AND* fusion rule of parallel (Par-AND), sequential (Seq-AND), sequential-parallel (Seq-Par-AND) strategies with homogeneous channels.

outperforms the sequential and parallel strategies. It can be seen that when the sensing time is very short, the performance of the sequential strategy and the sequential-parallel is the same. This means that the optimal decision by the sequential-parallel strategy also senses all channels by all users. When sensing time is very long, the sequential-parallel results are close to the parallel strategy.

Comparing the *AND* and *OR* rules, the performance is similar to the parallel strategy. The reason is that it is better to distribute the users more evenly in terms of sensing time, implying that each user will sense one channel, and no cooperation takes place. Therefore, the fusion model is irrelevant. For the sequential strategy, we could see that the *AND* rule outperforms the *OR* rule when the sensing time is short. It can be explained by the fact that the sensing time is shorter for the *AND* rule. However for $Q_d = 0.9$, they have the same performance and after that the *OR* rule performs better since the sensing time becomes shorter for the *OR* rule. It is worth noting that decreasing the Q_f or decreasing the sampling frequency has a similar impact in both rules, as the sensing time increases.

In Figure 3(b), the number of users is fixed to three. When M increases, it is expected to have some performance gain. However, the parallel strategy can not sense more than 3 channels and thus its throughput saturates at 3 channels. The sequential strategy has some gain until no more channels can be sensed as M increases. Given the sensing time equal to τ_N, maximum $\lfloor \frac{T}{\tau_N} \rfloor$ channels may be sensed in one time slot, which is $\lfloor \frac{1ms}{0.128ms} \rfloor = 7$ channels in this figure. The sequential-parallel strategy may have three user subsets and in each subset, maximum $\lfloor \frac{T}{\tau_1} \rfloor = \lfloor \frac{1ms}{0.24ms} \rfloor = 4$ channels can be sensed. The saturation thus occurs at $M = 12$. Discussions related to comparing AND and OR fusion rules are similar to the previous figure.

In Figure 3(c), we can observe that when $N < M$, the parallel strategy can not sense all channels. Therefore, the sequential strategy outperforms the parallel strategy. When the number of users increases, the parallel strategy is able to sense all channels, so it becomes superior. The other interesting point to observe is that as the number of users increases, OR outperforms AND because cooperative sensing becomes more likely and for the given Q_d and Q_f, the OR fusion model has shorter cooperative sensing times.

5 Conclusions

In this paper, we propose and compare several cooperative spectrum sensing strategies, i.e., sequential, parallel, and sequential-parallel to schedule all the users to sense multiple channels in order to achieve the optimal throughput. For each strategy, we propose several solutions including low-complexity heuristic and dynamic programming methods. In terms of throughput performance, we show that with longer sensing time, such as when we have a stringent constraint on probability of detection, smaller number of channels, or larger number of users, the parallel sensing strategy is recommended. Otherwise the sequential sensing strategy should be adopted.

References

[1] Q. Zhao and B. Sadler, A survey of dynamic spectrum access, IEEE Signal Processing Magazine, 24(3), 79–89, May 2007.

[2] Y.-C. Liang, Y. Zeng, E.C.Y. Peh, and A.T. Hoang, Sensing-throughput tradeoff for cognitive radio networks, IEEE Trans. Wireless Commun., 7(4), 1326–1337, Apr. 2008.

[3] D. Cabric, A. Tkachenko, and R.W. Brodersen, Spectrum sensing measurements of pilot, energy, and collaborative detection, in IEEE Military Commun. Conf. (MILCOM), Oct. 2006.

[4] H. Kim and K.G. Shin, Efficient discovery of spectrum opportunities with mac-layer sensing in cognitive radio networks, IEEE Trans. Mobile Comput., 7(5), 553–545, May 2008.

[5] C.-H. Liu, J. Tran, P. Pawelczak, and D. Cabric, Traffic-aware channel sensing order in dynamic spectrum access networks, IEEE JSAC, 31(11), 2312–2323, Nov. 2013.

[6] H. Kim and K.G. Shin, Fast discovery of spectrum opportunities in cognitive radio networks, in IEEE DySPAN, Oct. 2008.

[7] Z. Khan, J. Lehtomaki, L. DaSilva, and M. Latva-Aho, Autonomous sensing order selection strategies exploiting channel access information, IEEE TMC, 12(2), 274–288, Feb. 2013.

[8] A. Azarfar, J. Frigon, and B. Sansò, User-differentiated channel recovery in multi-channel cognitive radio networks, in International Conference on DRCN, March 2013, 242–249.

[9] M. Lopez-Benitez and F. Casadevall, Time-dimension models of spectrum usage for the analysis, design and simulation of cognitive radio networks, IEEE TVT, 62(5), 2091–2104, Jun. 2013.

[10] A. Ghasemi and E. Sousa, Collaborative spectrum sensing for opportunistic access in fading environments, in IEEE DySPAN, Nov. 2005, 131–136.

[11] D.P. Bertsekas, Dynamic Programming and Optimal Control. Athena Scientific, 2005, vol. I, 3rd Edition.

[12] S. Martello and P. Toth, Knapsack Problems: Algorithms and Computer Interpretations. Wiley-Interscience, 1990.