

# Multi-Channel Selection Maximizing Throughput for Delay-Constrained Multi-Application Secondary Users in Dynamic Cognitive Radio Networks

Luca Zappaterra\*, Hyeong-Ah Choi\*, Xiuzhen Cheng\*, Taieb Znati†

\*Department of Computer Science, George Washington University, Washington DC

†Department of Computer Science, University of Pittsburgh, Pittsburgh PA

Email: {lucaz,hchoi,cheng}@gwu.edu, znati@cs.pitt.edu

**Abstract**—In dynamic cognitive radio networks (CRNs) secondary users (SUs) sense the spectrum bands to find temporal absence of primary users (PUs) and immediately transmit on the identified spectrum holes. SUs sequentially sense the channels, stopping when the available resources are expected to provide the best throughput performance. Following, the selected channels are exploited using multi-channel transmission. In this paper, the multi-channel selection problem for SUs supporting multiple applications generating traffic with different latency requirements is formulated in a CRN with both heterogeneous PU-traffic and channel conditions. We have proposed an optimal solution overcoming the expensive computations and storage requirements typical of optimal stopping problems. Our efficient algorithm only requires linear-time and quadratic-space complexities in the online decision phase, aided by statistical decision values efficiently pre-computed offline. Extensive evaluations validate our solution as a significant improvement over the application of existing solutions, all based on the well-known backward induction technique or its approximations, characterized by either intractable algorithmic complexities or approximate results.

## I. INTRODUCTION

The government regulatory bodies have historically assigned the radio spectrum bands redundantly to applications, resulting in almost all the frequency bands of the usable radio spectrum being allocated. The recent enormous increase of mobile traffic expected to continue in the next years [1], and the proliferation of new wireless applications [2] have created a shortage of bandwidth supply known as the spectrum scarcity problem. At the same time, spectrum measurements [3] highlighted that in different locations and instants in time, significant portions of the radio spectrum (called spectrum holes) are underutilized.

*Cognitive Radio Networks (CRNs)* ease the spectrum scarcity problem by providing opportunistic access for unlicensed secondary users (SUs), to operate on the spectrum holes left by the licensed primary users (PUs). Typically, SUs monitor the channels to detect the presence of PUs, and possibly exploit the resources when PUs are not transmitting. Static CRNs are characterized by channels with stable PU-activity in both space and time domains. Instead, *dynamic CRNs* have rapidly changing PUs' transmission patterns, generating spectrum holes, whose durations are comparable with SUs' communication times. Since large parts of the wireless spectrum are allocated to PUs with rapid and bursty transmission patterns, enabling dynamic CRN operations creates new

exploitable resources, improving the spectrum utilization.

SUs can support multiple applications (such as streaming, file transfer, remote monitoring), each with different requirements of latency and throughput. A time-slotted system is adopted to provide SUs' synchronization and PUs' protection in the dynamic CRN. During each slot the SUs sequentially sense the frequency bands to find the spectrum holes and estimate the channel gains between the SU-transmitter and the SU-receiver. When channels free from PUs' transmissions and with favorable link qualities to meet the application requirements are found, SUs exploit them using *multi-channel transmission* for the remaining part of the time slot. This is accomplished using Discontiguous Orthogonal Frequency Division Multiplex (D-OFDM) [4], which allows to transmit on multiple orthogonal subcarriers, disabling the ones corresponding to the channels in which PU-signals are present.

While sensing more channels could increase the likelihood of finding better resources to support the application requirements, it also incurs the risk of finding channels busy or with poor link qualities, therefore wasting transmission time and so degrading SUs' throughput and delay. The trade-off between these two aspects is the target of the *channel selection* problem, which aims to find the optimal set of channels to satisfy the SU's requirements of *throughput maximization* and *constrained delay* imposed by the application needs. The channel selection problem can be modeled as a finite-horizon optimal stopping problem [5], deciding at any step if to search for additional channels or stop and transmit. The known backward induction method and the  $k$ -stage look-ahead ( $k$ -sla) approximation method suffer of either elevated complexity or imprecision. Also, the multiple traffic types supported by SUs impose to consider separate parameters for each application, significantly aggravating the complexity of the problem.

In this paper, the channel selection problem for SUs opportunistically transmitting over multiple channels through D-OFDM is studied. The delay requirements of the SUs' applications as well as the heterogeneous characteristics of the CRN generate challenging algorithmic complexities, requiring efficient solutions implementable even in hardware-constrained devices. Our main contributions are the following:

- We have devised a low-complexity optimal channel selection algorithm, which maximizes the throughput and

satisfies the delay constraints of SUs' applications, representing an efficient solution for dynamic CRNs.

- We have demonstrated the practicality of our solution which (unlike the backward induction based optimal stopping rule) shows limited execution times and storage needs, even for SUs supporting a large number of applications and the CRN having large number of channels.
- Through extensive performance evaluation, we have shown that our algorithm outperforms the maximum throughput achievable by existing approximate solutions based on the  $k$ -sla method, therefore being the preferred channel selection solution for dynamic CRNs.

Next, Section II presents the related works. Section III details the system model adopted. Section IV- V present the multi-channel selection algorithms and evaluate their performance respectively. Finally, Section VI concludes the paper.

## II. RELATED WORK

The channel selection problem for throughput maximization is formulated in literature as a finite horizon optimal stopping problem [5]. Previous works including [6]–[8] mainly studied the case of SUs transmitting on a single channel, in which the computational complexity represents a minor concern.

Limited works have studied the problem of channel selection with multi-channel transmission. In [9] authors use the backward induction method to obtain the maximum throughput within a transmission slot. The constraints of maximum number of adjacent channels and bandwidth fragments lead to exponential complexity. Taking advantage of the channels' homogeneity, a  $k$ -sla approximation is adopted, achieving almost optimal results. Authors in [10] use a  $k$ -sla approximation stopping rule for the multi-channel selection in a CRN with known channel rates, achieving close to optimal performance by applying first the 1-sla stopping rule, until it tells to stop and transmit, then applying the 2-sla until it also tells to stop, and so on. This process can become computationally expensive. A-priori knowledge of channel transmission rates is assumed, which may not be possible.

In [11] we have devised a low-complexity multi-channel selection algorithm, using only PUs' traffic and channels' data rates statistics available at SUs. In [12] we have extended the low-complexity algorithm to the case when knowledge of channel gains instead of data rates is available, and the channel selection is constrained by the limited power available at SUs.

To the best of our knowledge this work is the first study incorporating the latency requirements of the multiple applications supported by SUs into the channel selection problem.

## III. SYSTEM MODEL

### A. Characteristics of channels

A CRN operates on  $M$  non-overlapping channels, each of bandwidth  $W$ . The channels represent portions of the spectrum potentially available for opportunistic use by SUs, in which PU-traffic is present with different patterns among channels.

A time-slotted system with slots of duration  $T$  is adopted. During each slot, a channel is available/unavailable for SUs to

use, depending on the presence/absence of PUs' transmission on it. A random variable  $X_i(t)$  models the state of channel  $i$  in respect to the presence of PU-activity during a slot  $t$ , which can assume values of 1/0 depending if the channel is available/unavailable, with  $q_i(t)$  channel availability probability of channel  $i$  during  $t$ . Slowly varying channels, with constant channel gains during  $t$  are considered. The gain of channel  $i$  during  $t$ , modeling the degrading effects affecting the wireless channel, is represented by a random variable  $H_i(t)$ , such that  $H_i(t) = g_l$  with probability  $p_{i,l}(t)$ , with  $g_1 < \dots < g_L$ .  $L$  represents the number of channel gain values possible in the CRN, which is assumed finite. Each SU comprises a single D-OFDM transceiver, capable of sending/receiving on multiple discontinuous channels at the same time. The aggregated data rate of  $m$  channels during a transmission slot  $t$  using D-OFDM with uniform power distribution is defined as:

$$b_m(r_1(t), \dots, r_m(t)) = \sum_{i=1}^m r_i(t) \approx W \sum_{i=1}^m x_i(t) \log_2 \frac{Sh_i(t)}{N_0 W z_m(t)} \quad (1)$$

with  $r_i(t)$  observed data rate of channel  $i$  during slot  $t$ ,  $x_i(t)$  and  $h_i(t)$  realizations of  $X_i(t)$  and  $H_i(t)$  respectively, and  $z_m(t) = \sum_{i=1}^m x_i(t)$  number of channels found available. The approximation in equation (1) is possible since the Signal-to-Noise-Ratio (SNR) can be assumed quite larger than 1.

### B. Secondary user characteristics

Although able to transmit or receive on multiple discontinuous channels concurrently, the single D-OFDM transceiver present in each SU can only send, receive or sense at the same time. Consequently, a SU with data to transmit, sequentially search one channel at a time for spectrum holes before transmitting. The SU-transmitter and its intended SU-receiver synchronously hop from channel to channel, sensing the presence of PUs and estimating the link quality. The SU-transmitter senses the presence of PUs on a channel, reporting the availability status of it. Channel sensing is assumed error-free, but including imperfect channel sensing would not alter the validity of our work. When a channel is found free, its gain between the SU-transmitter and the SU-receiver is estimated by sending known pilot symbols and feeding back the obtained result. The sensing and estimating processes including the switching delay incur a constant delay  $\tau$  for each channel.

The order in which channels are sensed is predetermined and it can be derived from techniques such as descending order of availability probabilities [6] or expected data rates [10] or from SU collision avoidance strategies [7]. Unless specified, no assumptions are made on the sequence of channels, which shows random characteristics of PU-traffic and channel gains. It is assumed that the arrays of statistics  $\mathbf{Q}(t) = [q_1(t) \dots q_M]$ , and  $\mathbf{P}_i(t) = [p_{i,1}(t) \dots p_{i,L}(t)]$  are available at SUs.

SUs should satisfy the application requirements of throughput (i.e., number of bits sent during a slot) maximization and bounded delay during a transmission slot. Clearly, the process of selecting the CRN's channels affects the delay of packets

$$V_m^{(M_v)}(r_1, \dots, r_m) = \max \left\{ y_m(r_1, \dots, r_m), q_{m+1} \sum_{l=1}^L p_{m+1,l} V_{m+1}^{(M_v)}(r_1, \dots, r_m, W \log_2 \frac{S g_l}{N_0 W z_{m+1}}) + (1 - q_{m+1}) V_{m+1}^{(M_v)}(r_1, \dots, r_m, 0) \right\} \quad (4)$$

$$y_m \geq \left( b_m^{(z_{m+k})} + W \sum_{j=m+1}^{m+k} q_j \sum_{l=1}^L p_{j,l} \log_2 \frac{S g_l}{N_0 W z_{m+k}} \right) [1 - (m+k)\alpha] \quad \text{with } 1 \leq m < M_v \quad (5)$$

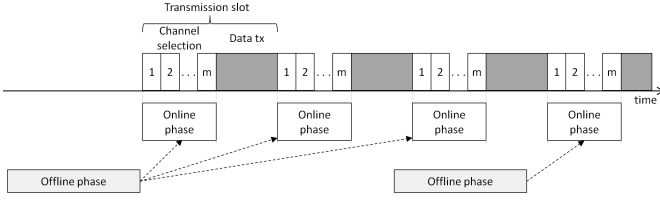


Fig. 1: Algorithmic framework for channel selection.

and may also reduce the throughput, if the increase in data rates does not compensate for the transmission time forgone.

SUs supporting up to  $A$  different types of applications, each characterized by a different maximum tolerable medium access delay requirement  $d_1 < d_2 < \dots < d_A$ , are considered. For an application of type  $v$ , with  $v \in \{1, 2, \dots, A\}$  the throughput during  $t$ , after sensing  $m$  channels is defined as:

$$y_m(r_1(t), \dots, r_m(t)) = b_m(r_1(t), \dots, r_m(t)) (1 - m\alpha) \quad (2)$$

with  $\alpha = \tau/T$ . The delay requirement of the application limits the number of explorable channels during the channel selection process as  $1 \leq m \leq M_v = \min\{M, \lfloor d_v/\tau \rfloor\}$ , with  $M_v \geq 2$ .

### C. Algorithmic framework

Channel information becomes available only after sensing one frequency band at a time, therefore the channel selection constitutes an online algorithm. The process has to be conducted efficiently, minimizing the computation time and memory overhead. The online complexity can be reduced by pre-computing offline the operations based on the statistical values available for the CRN. In fact, these statistics remain valid for multiple consecutive slots, until the CRN conditions significantly change. Figure 1 illustrates the alternation of offline and online phases within the channel selection process.

## IV. CHANNEL SELECTION ALGORITHMS

### A. Backward induction

For an application  $v$  with maximum latency requirement of  $d_v$ , given the current achievable throughput  $y_m(r_1, \dots, r_m)$ , the decision if to stop at  $m$  and transmit for the rest of the slot, or continue the channel selection, is based on the expected maximum throughput obtained from the dynamic programming recurrence relation in equations (3-4) with  $1 \leq m < M_v$ .

$$V_{M_v}^{(M_v)}(r_1, \dots, r_{M_v}) = y_{M_v}(r_1, \dots, r_{M_v}) \quad (3)$$

The base case (3) computes the throughput achievable when  $M_v$  channels are explored. Equation (4) backwardly takes

the best between the throughput after sensing  $m$  channels, and the expected maximum throughput if the next channel is considered. If the throughput  $y_m$  achievable with the current rates corresponds to the values of  $V_m^{(M_v)}$  for the same rates, the SU starts transmitting data, otherwise it explores the next channel. Similarly to the backward induction formulations in [11], [12], computing offline the  $V_m^{(M_v)}$  rewards values requires an asymptotic complexity of  $O\left(M \cdot \binom{M+L}{M} \cdot L\right)$ . Considering  $A$  applications, it approximates to  $O(A \cdot M^L)$ .

When the number of channels  $M$  is elevated, this approach requires expensive computations and large memory requirements, exceeding the resources of commodity devices.

### B. $k$ -stage look-ahead ( $k$ -sla)

This approximated solution decides at each stage if to stop or continue, depending on whether the rule optimal among those truncated  $k$  stages ahead stops or continues [5]. A simplified  $k$ -sla stopping rule is obtained in (5) similarly to the multi-channel case in [10]. For an application  $v$ , with a maximum tolerable delay  $d_v$  translating into maximum number of explorable channels  $M_v$ , it finds when the throughput attainable using the  $z_m$  available channels among  $1, \dots, m$  is no less than the expected throughput after adding the available channels among the next  $k$ . The term  $b_m^{(z_{m+k})}$  adapts  $b_m$  to consider the effect on the SNR of transmitting over  $z_{m+k}$  channels:

$$b_m^{(z_{m+k})} = W \sum_{i=1}^m x_i \log_2 \frac{S h_i}{N_0 W z_{m+k}} = b_m + W Z_{m,m+k} \quad (6)$$

with  $Z_{m,n} = \{0 \text{ if } z_m = 0; z_m \log_2 \frac{z_m}{z_n} \text{ otherwise}\}$ . The number of channels available in  $m+k$  steps is estimated as  $z_{m+k} = z_m + \sum_{i=m+1}^{m+k} q_i$ . The double summation in (5) computes the expected aggregate rate from the next  $k$  channels.

The  $k$ -sla method achieves close to optimal results by applying incremental  $k$ -sla rules, starting from 1-sla, increasing if the current rule suggests to stop, up to the  $k$ -sla. The process leads to an online complexity of  $O((M+k) \cdot k \cdot L)$ . Although, pre-computation may reduce the online complexity, this solution becomes either imprecise or computationally expensive. In fact, when the next  $k$  channels do not constitute an accurate representation of the all selectable CRN's channels, the result would be imprecise. The probability of this event decreases by increasing  $k$ , which aggravates the online computations.

### C. Delta algorithm

In the proposed algorithm, the channel selection for an application  $v$  stops if the throughput achievable after visiting  $m$

channels is no less than the largest expected throughput that can be possibly achieved using future channels  $m + 1$  to  $M_v$ . The stopping rule at step  $m$  is defined as:

$$y_m \geq \max_{m < n \leq M_v} \{(b_m^{(z_n)} + B_{m,n})(1 - n\alpha)\} \quad 1 \leq m < M_v \quad (7)$$

with  $b_m^{(z_n)}$  current attainable aggregate data transmission rate  $b_m$  adapted to consider the effect on the SNR of transmitting over  $z_n$  channels (i.e.,  $b_m^{(z_n)} = b_m + WZ_{m,n}$ ) and  $B_{m,n}$  expected aggregate data transmission rate supported by the channels  $m + 1$  to  $n$  derived as follows:

$$\begin{aligned} B_{m,n} &= W \sum_{j=m+1}^n q_j \sum_{l=1}^L p_{j,l} \log_2 \frac{S g_l}{N_0 W z_n} \\ &= W \sum_{j=m+1}^n q_j \left( \log_2 \frac{S}{N_0 W z_n} + \sum_{l=1}^L p_{j,l} \log_2 g_l \right) \end{aligned} \quad (8)$$

For each step  $m$ , the formula in (7) can be rewritten as:

$$y_m \geq (b_m^{(z_n)} + B_{m,n})(1 - n\alpha) \quad \forall n, m < n \leq M_v \quad (9)$$

and translated into a condition on the data transmission rate:

$$b_m \geq (WZ_{m,n} + B_{m,n}) \frac{1 - n\alpha}{(n - m)\alpha} = \Delta_{m,n}(v) \quad \forall n, m < n \leq M_v \quad (10)$$

The stopping rule for an application  $v$  stated in (7), is equivalent to the stopping rule in (11), based on Theorem 1.

$$b_m \geq \Delta_m(v) = \max_{m < n \leq M_v} \{\Delta_{m,n}(v)\} \quad 1 \leq m < M_v \quad (11)$$

**Theorem 1.** For an application  $v$  seeking for throughput maximization constrained by the maximum number of explorable channels  $M_v$ , the two statements are equivalent at any channel selection step  $m$ : (i) The rate  $b_m$  is no less than  $\Delta_{m,n}(v)$ . (ii) The throughput  $y_m$  is no less than the maximum expected throughput if exploration continues until any step  $n$ , i.e.

$$b_m \geq \Delta_{m,n}(v) \iff y_m \geq (b_m^{(z_n)} + B_{m,n})(1 - n\alpha) \quad \forall m, n, 1 \leq m < n \leq M_v \quad (12)$$

The proof of Theorem 1 is derived similarly to the proof in [11]. The algorithm implementation results in the following operations. At every step  $m$ , the rate achievable exploiting the channels found idle (i.e.,  $z_m$ ), is compared with the corresponding threshold, denoted as  $\Delta_m^{(z_m)}(v)$ . The threshold values are pre-computed and stored in memory, reducing the online phase to a comparison operation for each step. Algorithm 1 illustrates the offline computations. Lines 1-8 compute the partial sums  $Q_i$  and  $G_i$ . Lines 9-25 compute the  $\Delta_m^{(z_m)}(v)$  values with  $O(M^3)$  time-complexity. Particularly, the outer loops consider all the possible steps and available channels for the channel selection process. The inner loop iterates through the remaining channels, switching the considered application when the delay limit of the current application is passed. The

pre-computed values are accessed during the online phase, requiring only linear-time and  $O(A \cdot M^2)$  space complexities.

## V. PERFORMANCE EVALUATION

### A. Evaluation of algorithmic complexities

The execution times and space requirements of the backward induction and Delta algorithms are studied. Instead, the  $k$ -sla method is not considered now, since it is an approximation. Different cases of number of applications supported by the SUs and number of channels in the CRN are considered. Three scenarios with varying number of channel gains supported are examined. The values of maximum explorable channels  $M_v$  for each application  $v$  are chosen equally spaced in  $\{2, \dots, M\}$ .

Figure 2 shows the execution times of the offline phases for the backward induction and Delta algorithms, running on a virtual machine with 3.0 GHz clock speed and 2.4 GB of RAM, having two to three times more resources than typical portable devices. The backward induction requires several seconds which is not tolerable, whereas the Delta algorithm has very constrained execution times, between hundreds of microseconds to few milliseconds. All variables significantly affect the backward induction algorithm, whereas our solution is only minimally affected by the number of channels.

Figure 3 shows the storage requirements of the backward induction and Delta algorithms to hold in memory the values pre-computed offline and needed for the online decisions. The

---

#### Algorithm 1 $\Delta_m^{(z_m)}(v)$ Pre-computing

---

**Require:**  $P_i$  for  $1 \leq i \leq M$ ,  $Q, \alpha, g_l$  for  $1 \leq l \leq L$ ,  $M_v$  for  $1 \leq v \leq A$

**Ensure:**  $\Delta_m^{(z_m)}(v)$

```

1:  $Q_0 = 0; G_0 = 0; \bar{v} = 1$ 
2: for  $i = 1 \rightarrow M_A$  do
3:    $\gamma = 0; Q_i = Q_{i-1} + q_i$ 
4:   for  $l = 1 \rightarrow L$  do
5:      $\gamma = \gamma + p_{i,l} \log_2 g_l$ 
6:   end for
7:    $G_i = G_{i-1} + q_i \gamma$ 
8: end for
9: for  $m = 1 \rightarrow M_{\bar{v}} - 1$  do
10:  for  $z_m = 0 \rightarrow m$  do
11:     $v = \bar{v}; \Delta_m^{(z_m)}(v) = 0$ 
12:    for  $n = m + 1 \rightarrow M_A$  do
13:       $z_n = z_m + Q_n - Q_m$ 
14:      if  $n > M_v$  then
15:         $v = v + 1$ 
16:         $\Delta_m^{(z_m)}(v) = \Delta_m^{(z_m)}(v - 1)$ 
17:      end if
18:       $\Delta_m^{(z_m)}(v) = \max \left\{ \Delta_m^{(z_m)}(v), W \frac{1 - n\alpha}{(n - m)\alpha} \left[ Z_{m,n} \right. \right. \right.$ 
19:         $\left. \left. + (G_n - G_m) + (Q_n - Q_m) \log_2 \left( \frac{S}{N_0 W z_n} \right) \right] \right\}$ 
20:    end for
21:  end for
22:  if  $m == M_{\bar{v}} - 1$  then
23:     $\bar{v} = \min\{\bar{v} + 1, A\}$ 
24:  end if
25: end for
26: return  $\Delta_m^{(z_m)}(v)$ 

```

---

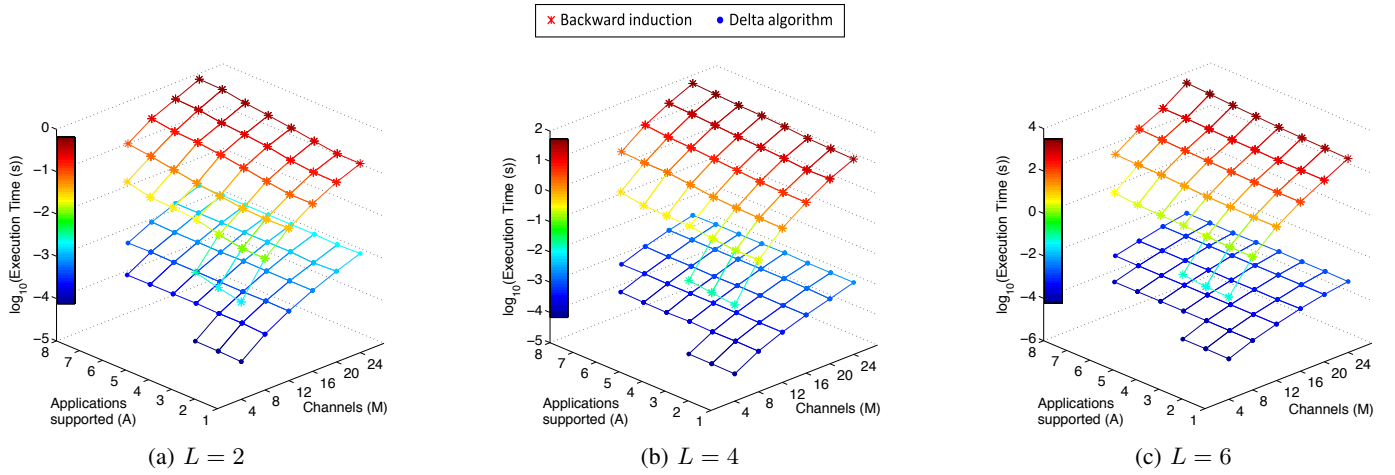


Fig. 2: Offline execution times when varying number of: applications supported ( $A$ ), channels ( $M$ ), and channel gains ( $L$ ).

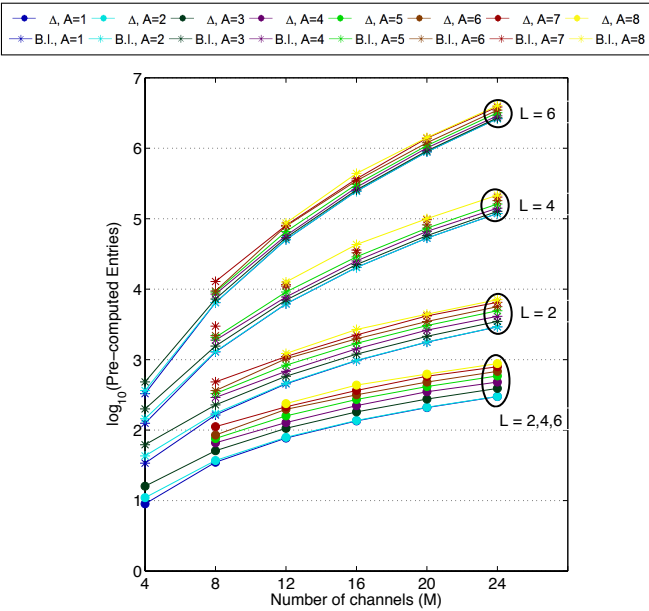


Fig. 3: Number of offline pre-computed entries for the backward induction (B.I.) and Delta ( $\Delta$ ) algorithms.

parameters considered are number of channels, applications supported, and channel gains. The Delta algorithm needs much less storage than the backward induction to run its operations. The memory savings range from 4-10 times for  $L = 2$  up to several thousands for  $L = 6$ , confirming the feasibility of our solution and the intractability of the backward induction.

### B. Network performance

Computer simulations are used with these settings: bandwidth  $W = 1$  MHz, signal power at the SU-receiver  $S = -80$  dBm, thermal noise spectral density  $N_0 = 290 \cdot 1.38e - 23$  W/Hz,  $T = 10$  ms, and  $\tau = 300 \mu s$ . The CRN considered has  $M = 25$  channels, with gains  $\{0.2, 0.7\}$  and three different probability mass functions distributed among channels:  $P_i = [0.8 \ 0.2]$ ,  $P_i = [0.5 \ 0.5]$ , and  $P_i = [0.2 \ 0.8]$ . The PU-traffic is

generated by a Poisson process with exponentially distributed durations multiple of  $T$ . SU-traffic is considered saturated. The *maximum throughput* achievable by knowing in advance the PU-occupancies and the channel gains is used for comparisons.

The average throughput of a SU application in a CRN with heterogeneous PU-traffic characteristics among channels is shown in Figure 4. Three scenarios are considered with four, three and two channel availability probability values possible, equally spaced in the range  $0.1 - 0.9$ . The Delta algorithm reaches 96–98% of the maximum throughput in all cases. The scenarios show similar behaviors, with our proposed solution performing better than the backward induction and the  $k$ -sla approximation solution. The  $k$ -sla approximation necessitates incremental values of  $k$  to achieve comparable performance.

Although the  $k$ -sla solution performs reasonably well in the scenario considered, we have identified that it suffers when the next  $k$ -channels are not be good indicators of all next channels. We have considered channel sequences alternating consecutive groups of very likely available channels with groups of very likely unavailable channels. Figure 5 shows the average throughput results when varying the  $k$  parameter. When the lengths of groups are large, the  $k$ -sla approximation performs poorly compared to the Delta and the maximum throughput values, slightly improving when the  $k$  parameter exceeds the group’s length. Performances improve for shorter groups of similar channels, where the  $k$ -sla becomes comparable to the

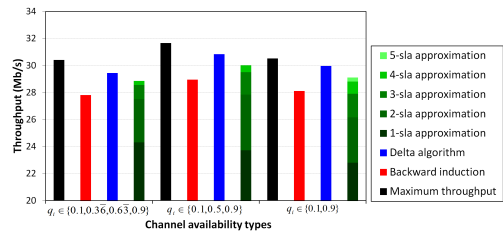


Fig. 4: Average throughput under three CRN configurations with heterogeneous characteristics of channel availabilities.

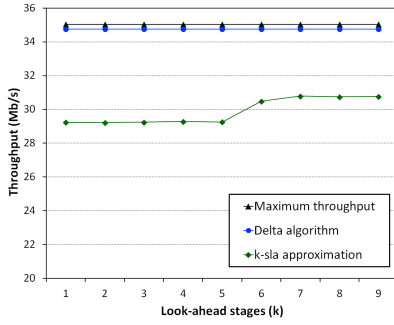
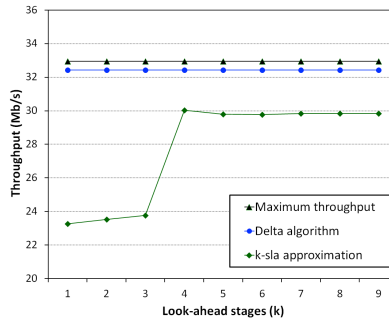
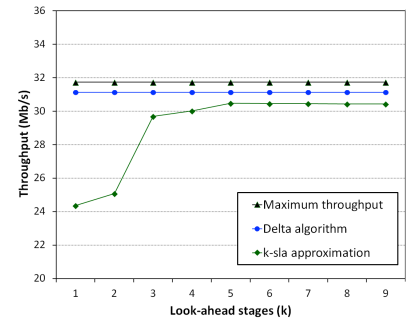
(a)  $Q = [0.9 \ 0.9 \ 0.9 \ 0.9 \ 0.9 \ 0.1 \ \dots]$ (b)  $Q = [0.9 \ 0.9 \ 0.9 \ 0.1 \ 0.1 \ 0.1 \ \dots]$ (c)  $Q = [0.9 \ 0.9 \ 0.1 \ 0.1 \ \dots]$ Fig. 5: Average throughput of the  $k$ -sla algorithm varying  $k$ , compared to the Delta algorithm and the maximum throughput.

TABLE I: Throughput and delay in realistic scenarios.

PU-traffic →		Heavy		Light		Heavy and light		Heterogeneous	
↓ Channels quality		$q_i = 0.1$		$q_i = 0.9$		$q_i = 0.1/0.9$		$q_i = 0.1 + \frac{0.9-i}{M-1}(i-1)$	
	$M_v = 24$	Th(Mb/s)	Delay( $\tau$ )	Th(Mb/s)	Delay( $\tau$ )	Th(Mb/s)	Delay( $\tau$ )	Th(Mb/s)	Delay( $\tau$ )
Poor/Avg/Best		8.11	16.26	46.68	14.42	31.06	14.17	30.33	14.51
	Max. Th.	9.89	13.71	47.21	14.34	31.58	14.05	31.6	13.9
Poor/Avg		7.9	16.22	44.83	14.34	30.00	13.99	29.11	14.46
	Max. Th.	9.6	13.73	45.31	14.22	30.52	13.9	30.44	13.9
Avg/Best		8.29	16.48	48.85	14.54	32.24	14.22	31.43	14.59
	Max. Th.	10.24	13.67	49.4	14.43	32.97	14.1	32.82	14.04
	$M_v = 20$	Th(Mb/s)	Delay( $\tau$ )	Th(Mb/s)	Delay( $\tau$ )	Th(Mb/s)	Delay( $\tau$ )	Th(Mb/s)	Delay( $\tau$ )
Poor/Avg/Best		7.84	15.53	46.65	14.42	30.81	14.08	30.24	14.43
	Max. Th.	9.61	12.99	47.16	14.33	31.36	13.98	31.56	13.9
Poor/Avg		7.59	15.60	44.83	14.34	29.97	14.07	29.15	14.44
	Max. Th.	9.33	12.9	45.34	14.25	30.58	13.85	30.44	13.9
Avg/Best		8.14	15.57	48.82	14.55	32.36	14.18	31.43	14.53
	Max. Th.	9.95	12.92	49.42	14.44	33.00	14.01	32.8	14.0
	$M_v = 16$	Th(Mb/s)	Delay( $\tau$ )	Th(Mb/s)	Delay( $\tau$ )	Th(Mb/s)	Delay( $\tau$ )	Th(Mb/s)	Delay( $\tau$ )
Poor/Avg/Best		7.33	13.64	46.4	14.41	30.05	13.38	29.62	13.84
	Max. Th.	9.07	11.29	47.26	14.29	31.17	13.10	31.42	13.12
Poor/Avg		7.19	13.57	44.61	14.32	29.27	13.35	28.49	13.78
	Max. Th.	8.86	11.23	45.36	14.15	30.34	13.10	30.17	13.09
Avg/Best		7.65	13.62	48.55	14.53	31.61	14.5	30.64	13.9
	Max. Th.	9.34	11.31	49.38	14.38	32.68	13.17	32.49	13.18

Delta solution as soon as  $k$  exceeds the length value.

Table I reports the average throughput and channel access delay of the Delta algorithm compared with the maximum throughput, for three applications with different delay requirements, translating into  $M_v = 24, 20, 16$  respectively, under varied realistic conditions of PU-traffic and link quality of channels. Four cases of PU-traffic are considered: (1) all channels with heavy PU-traffic (i.e.,  $q_i = 0.1$ ); (2) all channels with light PU-traffic (i.e.,  $q_i = 0.9$ ); (3) half of the channels with light and half with heavy PU-traffic; (4) heterogeneous PU-traffic, with channel characterized by different availability probabilities, equally spaced in the interval  $0.1 - 0.9$ . Channel gain values are chosen among  $[0.2 \ 0.45 \ 0.7]$  with different probabilities, modeling *poor* ( $P_i = [0.7 \ 0.2 \ 0.1]$ ), *average* ( $P_i = [0.1 \ 0.8 \ 0.1]$ ), and *best* ( $P_i = [0.1 \ 0.2 \ 0.7]$ ) link quality channels. The following CRN cases considered: (1) poor, average, and best channels; (2) poor and average channels; (3) average and best channels. In all cases, the Delta algorithm performs greatly, reaching 81–98% of the maximum throughput, and channel access delays from equal to a maximum  $2.8\tau$  increase, compared to the maximum throughput solution.

## VI. CONCLUSIONS

We have presented the multi-channel selection problem for SUs supporting multiple applications with different latency

requirements, in dynamic CRNs with heterogeneous channels. We have proposed an efficient algorithm, with very limited execution times and storage requirements, which make it very appropriate for implementations in hardware-constrained SUs. The validity of our solution has been demonstrated through extensive experimental and simulation results, confirming the algorithmic efficiency and network performance quality, in comparison with existing solutions and theoretical results.

## REFERENCES

- [1] “Cisco visual networking index: Global mobile data traffic forecast update, 20122017,” *Cisco, White Paper*, Feb 2013.
- [2] D. Raychaudhuri and M. Gerla, *Emerging Wireless Technologies and the Future Mobile Internet*, vol. 1.
- [3] M. A. McHenry, “NSF Spectrum Occupancy Measurements Project Summary,” *Shared Spectrum Company, Tech. Report*, Aug 2005.
- [4] J. Poston and W. Horne, “Discontiguous OFDM considerations for dynamic spectrum access in idle TV channels,” in *New Frontiers in Dynamic Spectrum Access Networks, 2005. DySPAN 2005. 2005 First IEEE International Symposium on*, 2005, pp. 607–610.
- [5] T. Ferguson, *Optimal stopping time and applications*, <http://www.math.ucla.edu/~tom/Stopping/Contents.html>.
- [6] H. Jiang, L. Lai, R. Fan, and H. Poor, “Optimal selection of channel sensing order in cognitive radio,” *Wireless Communications, IEEE Transactions on*, vol. 8, no. 1, pp. 297–307, 2009.
- [7] Z. Khan, J. Lehtomki, L. DaSilva, and M. Latva-aho, “Autonomous sensing order selection strategies exploiting channel access information,” *Mobile Computing, IEEE Transactions on*, vol. 12, no. 2, pp. 274–288, 2013.
- [8] T. Shu and M. Krunz, “Throughput-efficient sequential channel sensing and probing in cognitive radio networks under sensing errors,” in *Proceedings of the 15th annual international conference on Mobile computing and networking*, ser. *MobiCom '09*. New York, NY, USA: ACM, 2009, pp. 37–48.
- [9] J. Jia, Q. Zhang, and X. Shen, “HC-MAC: A Hardware-Constrained Cognitive MAC for Efficient Spectrum Management,” *Selected Areas in Communications, IEEE Journal on*, vol. 26, no. 1, pp. 106–117, 2008.
- [10] H. T. Cheng and W. Zhuang, “Simple Channel Sensing Order in Cognitive Radio Networks,” *Selected Areas in Communications, IEEE Journal on*, vol. 29, no. 4, pp. 676–688, Apr. 2011.
- [11] L. Zappaterra, J. S. Gomes, A. Arora, and H.-A. Choi, “A linear-time algorithm for optimal multi-channel access in cognitive radio networks,” in *World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2013 IEEE 14th International Symposium and Workshops on a. IEEE*, 2013.
- [12] L. Zappaterra and H.-A. Choi, “Efficient Power-Aware Multi-Channel selection for dynamic cognitive radio networks,” in *IEEE WCNC'14 Track 2 (MAC and Cross-Layer Design) (IEEE WCNC'14 Track 2 : MAC)*, Istanbul, Turkey, Apr. 2014.