

Optimizing Channel Selection for Cognitive Radio Networks using a Distributed Bayesian Learning Automata-based Approach*

Lei Jiao,[†] Xuan Zhang,[‡] B. John Oommen[§] and Ole-Christoffer Granmo[¶]

Abstract

Consider a multi-channel Cognitive Radio Network (CRN) with multiple Primary Users (PUs), and with multiple Secondary Users (SUs) competing for the access to the channels. In this scenario, it is essential for SUs to avoid collision among one another while keeping an efficient usage of the available transmission opportunities. We investigate two channel access schemes. In the first model, an SU selects a channel and sends a packet directly without Carrier Sensing (CS) whenever the PU is absent on this channel. In the second model, an SU invokes CS in order to avoid collision among co-channel SUs. For each model, we analyze the channel selection problem and prove that it is a so-called “Exact Potential” game. We also formally state the relationship between the global optimal point and the Nash Equilibrium (NE) point as far as system capacity is concerned. Thereafter, to facilitate the SU to select a proper channel in the game in a distributed manner, we design a Bayesian Learning Automaton (BLA)-based approach. Unlike many other Learning Automata (LA), a key advantage of the BLA is that it is learning-parameter free. The performance of the BLA-based approach is evaluated through rigorous simulations and this has been compared with the competing LA-based solution reported for this application, whence we confirm the superiority of our BLA approach.

Keywords : *Cognitive Radios, Channel Access, Multiple Users, Potential Games, Bayesian Learning Automata.*

*A preliminary version of some of the results of this paper was presented at IEAAIE-2014, the 27th International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems, Kaohsiung, Taiwan, in June 2014. [4].

[†]This author can be contacted at: Department of ICT, University of Agder, Grimstad, Norway. E-mail address: lei.jiao@uia.no .

[‡]This author can be contacted at: Department of ICT, University of Agder, Grimstad, Norway. E-mail address: xuan.zhang@uia.no .

[§]*Chancellor’s Professor, Fellow: IEEE and Fellow: IAPR.* This author can be contacted at: School of Computer Science, Carleton University, Ottawa, Canada : K1S 5B6. This author is also an *Adjunct Professor* with the University of Agder in Grimstad, Norway. E-mail address: oommen@scs.carleton.ca .

[¶]This author can be contacted at: Department of ICT, University of Agder, Grimstad, Norway. E-mail address: ole.granmo@uia.no .

1 Introduction

In Cognitive Radio Networks (CRNs) [7], channels allocated to Primary Users (PUs) can be reused by Secondary Users (SUs) opportunistically whenever the respective channel is not occupied by the PU. Due to the stochastic property of the traffic generated by the PUs, different channels may have distinct probabilities of being idle, as far as the SUs are concerned. Therefore, in order to reuse the channels in an efficient manner, it is of essence for the SUs that they can “learn” the characteristics of the channels and to adjust the channel selection intelligently. Channel access in CRNs can be organized either in a centralized manner or in a distributed manner. When a centralized channel access scheme is adopted, an omniscient central controller is responsible for allocating the traffic among the different channels [3]. On the contrary, in a distributed channel access scheme, there is no central controller, and each individual SU needs to decide for itself whether or not to opt for any specific channel based on the knowledge of the environment.

When distributed channel selection schemes are applied, the SUs are supposed to learn the properties of the channels themselves and to thereafter choose a channel that possesses a higher successful transmission probability. To achieve this goal, Learning Automata (LA) [1, 6, 8] have been widely employed by SUs for channel selection in CRNs [9, 11–13]. The benefit of using LA for channel selection is that the SUs can learn from the environment and adjust themselves appropriately. More importantly, the process of learning and making decisions happens simultaneously without requiring any prior knowledge of the system.

The existing work of channel selection in CRNs in which Learning Automata (LA) is applied can be cataloged into two categories, depending on the number of SU communication pairs in the system. In the first category, one assumes that only one SU communication pair exists in the system [9, 11, 13], and the goal of the LA is to discover and converge to a channel that has the highest idle probability among the channels. In the second category, multiple SU communication pairs exist in the system and they compete with each other for medium access [12]. The latter category is, obviously, more complicated as SUs do not need to merely avoid collisions with PUs in order to protect the services offered by the PU channels. They have to also avoid colliding with *other* SUs. In this category, the scenario when Carrier Sensing (CS) is enabled has been analyzed in depth in [12], where the system is modeled as a potential game. Furthermore, in that paper [12], a Linear Reward-Inaction (L_{R-I}) LA was utilized to play the game.

For the L_{R-I} , in order to achieve the best trade-off between the accuracy and the learning speed, the scheme’s optimal learning speed (achieved by selecting the optimal learning parameter) had to be pre-determined. However, due to the stochastic characteristics of the PUs’ traffic among the different channels, the idle probabilities of the various channels could vary with time. Consequently, it is not an easy task for the user to find the optimal learning parameter *a priori*, or for the scheme to adapt the learning parameter so as to follow a dynamic environment. This has motivated us to design a scheme which does not require the configuration on any learning parameter. Furthermore, as the CS procedure, in and of itself, requires air time in a time slot, it would be an advantage if we could use this time for communication if the learning

algorithm is efficient enough to resolve the collisions among the SUs. Therefore, it is also interesting for us to investigate the performance of the system when a CS procedure is not involved, and to see if learning by itself is sufficient for us to resolve the collisions among the SUs in certain circumstances.

In this work, we consider two channel access strategies, i.e., channel access with and without CS. Based on the strategies and the system models, we formulate the problems to be solved as so-called “Potential Games” and analyze their properties. Thereafter, we propose a Bayesian Learning Automata (BLA)-based learning scheme as a novel solution to this problem, and its efficiency has been validated through detailed, rigorous simulations.

In the interest of scientific ethics and completeness, it is prudent to mention¹ the relationship between this paper and an earlier preliminary version presented at a prestigious AI conference [4]. The latter conference version was published to merely lay a claim on the results, and was written at a very initial stage of this work. The proposed algorithm was not validated completely, and the implementation of the algorithms was only partially done to get the preliminary set of results. Further, it did not include a comprehensive survey of the state-of-the-art. The current journal version was prepared several months after the work matured to consider the cases without CS and also with CS. The conference paper did not contain the formal proofs of the theoretical results for the game models of the two scenarios (with and without the current CS model), and the corresponding utility functions. It is appropriate to further emphasize the differences between the two theoretical analyses more explicitly. In the conference version, we had explained that the optimal point in the case without the current CS model is an Nash Equilibrium (NE) point. This was a “one-directional mapping”, i.e., we showed that a globally optimal solution mapped onto the NE point. In the current journal version, we not only prove that the optimal point is an NE point, but also demonstrate that the NE point is actually a globally optimal solution. This involves the bi-directional mapping between the globally optimal solution and the NE point. Finally, unlike in the conference version, the experimental results have been explained for all the settings and with appropriate detail. Indeed, the current journal version has a larger ensemble of experimental results, and they permit us to submit a broader perspective on the performance for the proposed algorithm.

In brief, the main contributions of this work are outlined as follows:

- First of all, we propose a BLA-based approach to resolving the problem of channel selection for SUs.

The advantages of the BLA-based approach are two-fold:

1. It does not require the configuration of any learning parameters in advance.
2. The performance of the BLA-based approach is superior to the L_{R-I} counterpart in terms of its learning speeds and learning accuracies, which fact is demonstrated in the section that presents the experimental results.

¹We are grateful to the anonymous Referee who requested this write-up to describe the difference between the earlier version [4] and this present version.

- We have analyzed the performance of the system without CS and demonstrated that the system can be modeled as a so-called “Potential Game”. We also prove a fascinating feature that the NE point of the game is actually the global optimal of the system quantified in terms of its capacity, and *vice versa*. Furthermore, we confirm, in the numerical results, that the proposed BLA-based learning approach can converge to the NE point. This not only means that the air time for the CS procedure can be used for purpose of communication, but also that the overall system capacity is optimal after convergence.
- We precisely model the scenario when the system is enabled with CS, and the corresponding procedure of achieving the competition during the CS period is precisely modeled. We have also demonstrated that the system is again a “Potential Game”, and that the BLA-based learning approach can converge to an NE point of this game.

The rest of this article is organized as follows. In Section 2, the related work is summarized in more detail. In Section 3, we describe and analyze the system model and the channel selection problems. In Section 4, we proceed to present the BLA-based distributed channel access scheme. Section 5 provides extensive simulation results that demonstrate the advantage of the BLA in channel selection. We conclude the paper in Section 6.

2 Related work

In what follows, we briefly summarize the existing work on channel selection in multi-channel CRNs where the AI tool utilized is the field of LA. The reported works are cataloged based on the number of SUs in the system.

In the single-user category, the application of LA in CRNs was first reported in [9], where the authors utilized the Discretized Generalized Pursuit Algorithm (DGPA) to help a single SU determine a channel that had the highest idle probability among the multiple channels. Similar to the L_{R-I} , the optimal learning speed of the DGPA had to be pre-determined in order to achieve the best trade-off between the learning speed and the accuracy. This is a difficult task, especially when this parameter has to be obtained and updated “on the fly”, as is the case in the scenario involving CRNs. In [11], the authors discussed the issue of determining the circumstances under which a new round of learning had to be triggered in CRNs, and the learning-parameter-based DGPA was again adopted as the learning scheme in the SU pairs.

To circumvent the limitations of the DGPA in CRNs, our recent work [13] suggested the incorporation of the BLA [1] into channel selection for the *single user* multiple channel scenario. The advantage of the BLA over the DGPA and other learning-parameter-based LA is that one requires no learning parameters to be pre-defined so as to achieve a reasonable (if not ideal) trade-off between the learning speed and the associated accuracy. Besides, we enabled the channel’s switching functionality in that work, meaning that it is possible for an SU to switch to another channel when the current channel is occupied by a PU, further

facilitating the transmission task of the SU. To sum up, in the single user multiple channel category, LA have been proven to be an efficient approach to solve the problem, and in particular, the BLA is especially suitable, as a learning-parameter-free scheme, for such a scenario.

Although a lot of efforts were invested in the single SU scenario and promising results were illustrated, the applicability of the LA-based approaches was not restricted to the single-user case, and its ability to learn in a more complicated environment had also been studied. In [12], the authors explored the scenario where multiple SUs competed for channel access in multiple channels. In that work, as the number of SUs in the system is assumed to be larger than the number of channels, CS was utilized, by default, in order to avoid co-channel collision among SUs. Based on the system configuration, the channel selection problem was formulated as a game. Thereafter, more mathematical insights were provided from the aspects of both the game itself and its potential solutions. Although the analytical and simulation results in [12] had shown the efficiency of LA in solving problems of this kind in CRNs, there were a few unresolved issues by which the performance could be potentially improved:

1. To allow the SU communication pairs to converge in a distributed manner, the L_{R-I} scheme was utilized to play the game, which, in turn, requires a learning parameter to be configured in advance. As the applicability and efficiency of the learning-parameter-free LA, i.e., the BLA, in game playing were earlier demonstrated for solving the Goore game [2], we were motivated to incorporate the BLA to solve the multi-SU scenario in CRNs, with the ultimate hope that the system's overall performance could be further improved by its inclusion.
2. As mentioned earlier, if the number of channels is greater than the number of SUs, a scheme that did not invoke CS is an interesting option. This option could be considered with the hope that the learning process can successfully resolve the potential collisions among SUs.
3. The CS process in [12] was assumed to be ideal, meaning that a single SU will certainly win the competition among multiple co-channel SUs. In other words, the event of collision among co-channel SUs, which, indeed, exists in reality, was ignored. To model the impact of the collision between potential co-channel SUs, we foresee the need for a more precise function that can describe the CS process. This is because a different model of the CS process will result in a distinct utility function for the game. Consequently, the property of the game under the new model, begs investigation.

Based on the above observations in the state-of-the-art, we are motivated here to investigate the above unresolved issues, and to propose BLA-based distributed approaches to solve the multi-user multi-channel problem in CRNs, and expect to contribute to the state-of-the-art.

In the following sections, we will detail the system configurations, analyze the various problems encountered, design the algorithms, and evaluate their performances by rigorous simulations.

3 System Model and Problem Formulation

In this section, we first present the system model for CRNs. Thereafter, we analyze the associated problems of channel selection.

3.1 System Model and Assumptions

Two types of radios, PUs and SUs, operate in a spectrum band consisting of N channels allocated to the PUs. PUs access the spectrum in a time-slotted fashion and the behavior of the PUs is independent from one channel to another. We assume that the SUs are synchronized with the PUs, and further that the supported data rates for the SUs are the same in all the channels. There are M (where $M > 1$) SU communication pairs in the network, and each of them needs to select, out of N channels, a specific channel for that time slot, for the purpose of communication. Without loss of generality, unless otherwise stated, we utilize the term ‘‘SUs’’ to refer to these SU pairs.

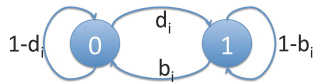


Figure 1: The On/Off channel model for the PU’s behavior.

To model the behavior of the PU in the i^{th} channel, $i \in \{1, \dots, N\}$, we adopt a two-state Markov chain model, as shown in Fig. 1. State 0 represents the condition when the channel is idle. Similarly, State 1 represents the case when the channel is being occupied by a PU. d_i and b_i are the transition probabilities between these two states in channel i . Thus one can verify that the steady state probability of the channel being idle is given by $p_i = \frac{b_i}{b_i + d_i}$.

To avoid collisions with PUs, at the beginning of each time slot, there is a quiet period for the SU to sense the channel. If the channel is determined by the SU to be unoccupied by the PU, the operations of CS (which can reduce the collision probability among the SUs) are carried out before the transmission of packets, if the strategy being utilized permits CS. If the strategy does not permit CS, the SUs will transmit a packet directly after sensing the channel associated with the PUs. The SU packet size is adjusted to be *a single* packet transmission per time slot for the given data rate. We assume that the task of channel sensing is ideal, and that due to the available advanced coding schemes, interruption because of channel fading for SUs will not occur at the given rate. It is assumed that there is a background protocol supporting channel access whose detailed signaling process is outside the scope of this work. We also assume that SUs always have packets ready for transmission.

In what follows, we formulate the problems for channel access with or without CS, as games, and in particular, as ‘‘Exact Potential’’ games. An ‘‘Exact Potential’’ game belongs to the set of ‘‘Ordinal Potential’’ games². It has been demonstrated that a distinguishing feature of a finite Ordinal Potential game is that it has

²More detailed information concerning the families of potential games can be found in [5]. It is omitted here to avoid repetition.

a pure strategy Nash Equilibrium (NE) [5]. We can therefore anticipate that by virtue of this phenomenon, each of the games studied in the respective CR scenarios, has a pure strategy NE.

If we try to put the pieces of the puzzle together, we see that the existence of a pure-strategy NE point in a game is *significant* for an LA-based algorithm. Indeed, at a pure-strategy NE point, each player selects a specific action with a probability value of unity. We can therefore expect that each of the players (i.e., the SUs in our case) can ultimately converge to a single action (channel) which is at the NE point arrived at by utilizing the LA. On the contrary, if for any specific game the solution is merely a *mixed*-strategy equilibrium solution, it implies that multiple actions have to be chosen each with a certain probability, for the players to attain the equilibrium point. Consequently, the property of the SU converging to a single action (channel) using an LA-based algorithm, does not make sense.

3.2 Channel Access Without CS

The structure of a time slot when CS is not enabled is shown in Fig. 2. As one can observe from this figure, there are two time segments in one slot: a quiet period and a period for packet transmission. After detecting an idle channel (the absence of a PU) by using the quiet period, an SU transmits a packet directly. In this case, collisions among the SUs will happen if two or more SUs appear on the same channel. To avoid such a collision, we require that at most one SU can exist in each channel. If $M > N$, i.e., if there are more SUs than channels, collisions among the SUs cannot be avoided because at least a single channel will have more than one SU. Consequently, the first strategy is applicable only when $M \leq N$. The case when $M > N$ will be studied in the next subsection when CS is enabled.

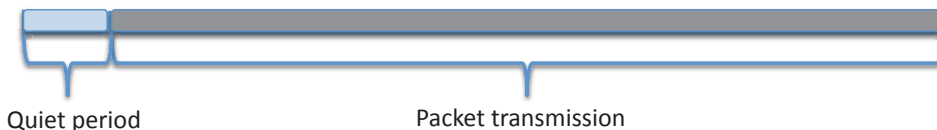


Figure 2: The structure of a time slot when CS is not enabled.

We formulate the channel selection problem when CS is not enabled as a game denoted by $\mathcal{G} = [\mathcal{M}, \{A_m\}_{m \in \mathcal{M}}, \{u_m\}_{m \in \mathcal{M}}]$, where $\mathcal{M} = \{1, \dots, M\}$ is the set of SUs, $A_m = \{1, \dots, N\}$ is the set of possible action/channel selections of the SU m , and u_m is the utility function of the SU, m . This utility function, associated with the SU m , is defined as

$$u_m(a_m, \mathbf{a}_{-m}) = p_{a_m} f(h(a_m)), \quad (1)$$

where $a_m \in A_m$ is the action/channel selected by the SU m , and $\mathbf{a}_{-m} \in A_1 \times A_2 \times \dots \times A_{m-1} \times A_{m+1} \times \dots \times A_M$ represents the channels selected by all the other SUs, and where the symbol \times represents the Cartesian

However, since this is central to our study, we will briefly outline the definitions and the relationships between the Exact Potential game and the Ordinal Potential game, where we shall also prove that the games encountered in our study are Exact Potential games.

product. As a_m is the index of the channel that is selected by the SU m , p_{a_m} represents the steady state probability of channel a_m being idle. The function $f(k) = 1$ if $k = 1$ and 0 otherwise. We denote $h(k)$ as the number of SUs that have selected channel k . $f(h(a_m))$ represents the event that a successful transmission has occurred, and this happens if and only if exactly a single SU exists in channel a_m .

Based on the above utility function, the NE can be expressed as follows. A channel selection scheme of all the M SUs $(a'_1, a'_2, \dots, a'_M)$, is an NE point of \mathcal{G} if no SU can improve its utility function by deviating unilaterally. Mathematically, this is formalized as $u_m(a'_m, \mathbf{a}'_{-m}) \geq u_m(a_m, \mathbf{a}'_{-m})$, $\forall m \in \mathcal{M}$ and $\forall a_m \in A_m \setminus \{a'_m\}$, where the notation $A \setminus B$ signifies the elimination of the set B from the set A .

We define the potential function for \mathcal{G} as

$$\alpha(a_m, \mathbf{a}_{-m}) = \sum_{i=1}^N \sum_{j=0}^{h(i)} p_i f(j) = \sum_{i, \forall h(i) > 0} p_i. \quad (2)$$

In the middle part of Eq. (2), the inner sum is over the number of SUs that select channel i , i.e., $h(i)$. In this particular case, the result of the inner sum is p_i if $h(i) > 0$, and is 0 if $h(i) = 0$. This leads us to the right hand side of the equation.

By definition, \mathcal{G} is an Exact Potential game if we can illustrate that when an arbitrary SU m changes from channel a_m to channel \tilde{a}_m while the other SUs maintain their respective channel selections unchanged, the change in the value of the potential function equals to the change in the utility of the SU, m .

To briefly explain the relationship between Exact Potential games and Ordinal Potential games, we mention that a game is an Ordinal Potential game if the following condition holds [5]:

$$u_m(\tilde{a}_m, \mathbf{a}_{-m}) - u_m(a_m, \mathbf{a}_{-m}) > 0 \text{ if and only if } \alpha(\tilde{a}_m, \mathbf{a}_{-m}) - \alpha(a_m, \mathbf{a}_{-m}) > 0. \quad (3)$$

Obviously, an Exact Potential game belongs to the class of Ordinal Potential games. Furthermore, in the games that we shall study, the number of SUs and the number of actions are limited, implying that the game is actually a *finite* game. Since every finite Ordinal Potential game possesses a pure-strategy NE point [5], it is true that \mathcal{G} has at least one pure strategy NE point if we can demonstrate that \mathcal{G} is an Exact Potential game.

We now state and prove our first result regarding strategies that do not involve CS.

Theorem 1 *The game \mathcal{G} is an Exact Potential game.*

Proof : We prove Theorem 1 by invoking the definition of an Exact Potential game.

For an arbitrary SU m that changes from channel a_m to channel \tilde{a}_m while the other SUs keep their channel choices unchanged, if it is true that $\alpha(\tilde{a}_m, \mathbf{a}_{-m}) - \alpha(a_m, \mathbf{a}_{-m})$ equals to $u_m(\tilde{a}_m, \mathbf{a}_{-m}) - u_m(a_m, \mathbf{a}_{-m})$, the game is an Exact Potential game [5].

Whenever an arbitrary SU m changes from channel a_m to channel \tilde{a}_m while the others keep their choices unchanged in the game \mathcal{G} , it is clear that *only* the number of SUs in channel a_m and the number of SUs in

channel \tilde{a}_m vary. In other words, everything is unchanged in all the other channels other than a_m and \tilde{a}_m . Therefore, we have:

$$\begin{aligned} & \alpha(\tilde{a}_m, \mathbf{a}_{-m}) - \alpha(a_m, \mathbf{a}_{-m}) \\ &= \left(\sum_{j=0}^{h(\tilde{a}_m)+1} p_{\tilde{a}_m} f(j) + \sum_{j=0}^{h(a_m)-1} p_{a_m} f(j) \right) - \left(\sum_{j=0}^{h(\tilde{a}_m)} p_{\tilde{a}_m} f(j) + \sum_{j=0}^{h(a_m)} p_{a_m} f(j) \right), \end{aligned} \quad (4)$$

where the above follows because the potentials in all the channels other than \tilde{a}_m and a_m get cancelled in the subtraction.

For the bottom half of Eq. (4), it is obvious that

$$\begin{aligned} & \left(\sum_{j=0}^{h(\tilde{a}_m)+1} p_{\tilde{a}_m} f(j) + \sum_{j=0}^{h(a_m)-1} p_{a_m} f(j) \right) - \left(\sum_{j=0}^{h(\tilde{a}_m)} p_{\tilde{a}_m} f(j) + \sum_{j=0}^{h(a_m)} p_{a_m} f(j) \right) \\ &= \left(p_{\tilde{a}_m} f(h(\tilde{a}_m)+1) + \sum_{j=0}^{h(\tilde{a}_m)} p_{\tilde{a}_m} f(j) + \sum_{j=0}^{h(a_m)-1} p_{a_m} f(j) \right) \\ & \quad - \left(\sum_{j=0}^{h(\tilde{a}_m)} p_{\tilde{a}_m} f(j) + \sum_{j=0}^{h(a_m)-1} p_{a_m} f(j) + p_{a_m} f(h(a_m)) \right) \\ &= p_{\tilde{a}_m} f(h(\tilde{a}_m)+1) - p_{a_m} f(h(a_m)) \\ &= u_m(\tilde{a}_m, \mathbf{a}_{-m}) - u_m(a_m, \mathbf{a}_{-m}). \end{aligned}$$

Therefore the condition $\alpha(\tilde{a}_m, \mathbf{a}_{-m}) - \alpha(a_m, \mathbf{a}_{-m}) = u_m(\tilde{a}_m, \mathbf{a}_{-m}) - u_m(a_m, \mathbf{a}_{-m})$ holds. Consequently, \mathcal{G} is, indeed, by definition, an Exact Potential game. Hence the theorem! \square

We now define the overall system capacity $C(a_1, a_2, \dots, a_M)$ as the sum of the utility functions of all the SUs, i.e., $C(a_1, a_2, \dots, a_M) = \sum_{k=1}^M p_{a_k} f(h(a_k))$. Based on the implications of Proposition 1 in [13], it is easy to conclude that the optimal solution for the overall system capacity,

$$(a_1^*, a_2^*, \dots, a_M^*) = \arg \max_{a_i, i \in \mathcal{M}} C(a_1, a_2, \dots, a_M),$$

is to select the channels with $p_i, i \in \{1, \dots, N\}$ from high to low, and with each channel being associated with only a single SU.

We now proceed to demonstrate the properties of the above-mentioned optimal point.

Theorem 2 For the game \mathcal{G} , the point $(a_1^*, a_2^*, \dots, a_M^*)$ is actually an NE point and vice versa.

Proof: It is not difficult to show that the point $(a_1^*, a_2^*, \dots, a_M^*)$ satisfying the above property is actually an NE point of \mathcal{G} , because any unilateral change of any SU will result in either a collision, i.e., if it tunes onto a channel where another SU resides, or the usage of a channel with a lower idle probability, i.e., if it tunes onto a channel where there is no SU. Thus, the global optimal solution is, indeed, an NE point of \mathcal{G} . Similarly, for an NE point, if it is not a global optimum, it implies that we can bring an SU from a lower-ranking channel (ranked in terms of the p_i 's) to an idle higher-ranking channel, leading to a positive payoff that can

be perceived to be the consequence of a unilateral change. Such an event cannot occur because it contradicts the fundamental definition of the NE point. Consequently, the NE point is a global optimal point in \mathcal{G} . \square

3.3 Channel Access with CS

In this strategy, SUs compete for channel accesses. The structure of a time slot when CS is enabled is demonstrated in Fig. 3. After the quiet period, when the channel is determined as being idle, an SU will perform CS in the contention period. In each round of contention, an SU randomly picks an integer from a contention window that has a fixed size, and then initiates a counting down phase. An SU will transmit a packet when the integer counts down to zero. During the counting down period, SUs will listen to the channel. If it detects the transmission of another SU, it will give up the transmission, meaning that it loses the contention for this round. Obviously, the SU that selects the minimum integer among all the integers selected by the co-channel SUs will reach zero first and consequently win the contention. If two or more contending SUs select the same integer that happens to be the minimum one, a collision will happen. The SUs which lose in the contention or which collide with one another will give up their right to transmit for this round. Statistically, the co-channel SUs will share the channel access opportunities equally.



Figure 3: The structure of a time slot when CS is enabled.

We formulate the channel selection problem when CS is enabled as a game denoted by $\mathcal{G}' = [\mathcal{M}, \{A_m\}_{m \in \mathcal{M}}, \{u'_m\}_{m \in \mathcal{M}}]$, where \mathcal{M} and A_m have the same definition as in the previous subsection. In this case, the utility function of SU m , u'_m , can be defined as

$$u'_m(a_m, \mathbf{a}_{-m}) = p_{a_m} f'(h(a_m), c), \quad (5)$$

with a_m , \mathbf{a}_{-m} , p_{a_m} and $h(k)$ being the same as defined in Eq. (1).

The function $f'(h(a_m), c)$ represents the probability of a successful transmission of SU m given $h(a_m)$ co-channel SUs that have a contention window of size $c > 1$.

This function can be expressed as:

$$f'(h(a_m), c) = \begin{cases} 0, & \text{for } h(a_m) = 0, \\ 1, & \text{for } h(a_m) = 1, \\ \frac{\sum_{i=1}^{c-1} (c-i)^{h(a_m)-1}}{c^{h(a_m)}}, & \text{for } 1 < h(a_m) \leq M. \end{cases} \quad (6)$$

The expression for $f'(h(a_m), c)$ with $1 < h(a_m) \leq M$ can be justified as follows. Given $h(a_m)$ SUs and a contention window with size c , the number of possible combinations of the integers that the SUs can

choose from so as to initiate BO is $c^{h(a_m)}$. Among those $c^{h(a_m)}$ combinations, we would like to calculate the number of combinations that results in a successful contention without collision. Whenever an SU wins a channel contention competition, it implies that it is the only SU that selects the least number among all numbers selected by the co-channel SUs. If the number selected by the successful SU is the smallest possible number within the contention window, the number of possible combinations is $h(a_m)(c-1)^{(h(a_m)-1)}$. Similarly, if the number selected by the successful SU is the second least possible number within the contention window, the number of possible combinations is $h(a_m)(c-2)^{(h(a_m)-1)}$, and so on. By summing all the possible combinations by which an SU can win a channel contention, we see that this quantity is $h(a_m)\sum_{i=1}^{c-1}(c-i)^{(h(a_m)-1)}$. We can therefore compute the probability of attaining an overall successful contention as $h(a_m)\sum_{i=1}^{c-1}(c-i)^{(h(a_m)-1)}/c^{h(a_m)}$. For each contending SU, the probability of success is the overall probability of succeeding averaged by the number of available co-channel SUs, $h(a_m)$. Consequently, we can see that the probability of success for an SU is $\sum_{i=1}^{c-1}(c-i)^{(h(a_m)-1)}/c^{h(a_m)}$.

As $f'(h(a_m), c)$ describes the procedure of CS with the consideration of the events involving potential collision, the utility function $u'_m(a_m, \mathbf{a}_{-m})$ is different from the one derived in [12] for this aspect.

We define the potential function for \mathcal{G}' as

$$\alpha'(a_m, \mathbf{a}_{-m}) = \sum_{i=1}^N \sum_{j=0}^{h(i)} p_i f'(j, c). \quad (7)$$

Similar to the case when we did not permit CS, we can demonstrate that the game \mathcal{G}' is an Exact Potential game.

Theorem 3 *The game \mathcal{G}' is an Exact Potential game.*

Proof: Similar to the proof of Theorem 1, we prove this result by invoking the definition of an Exact Potential game. Whenever an arbitrary SU m changes from channel a_m to channel \tilde{a}_m while the other SUs keep their channel choices unchanged, the following equation holds:

$$\begin{aligned} & \alpha'(\tilde{a}_m, \mathbf{a}_{-m}) - \alpha'(a_m, \mathbf{a}_{-m}) \\ &= \left(\sum_{j=0}^{h(\tilde{a}_m)+1} p_{\tilde{a}_m} f'(j, c) + \sum_{j=0}^{h(a_m)-1} p_{a_m} f'(j, c) \right) - \left(\sum_{j=0}^{h(\tilde{a}_m)} p_{\tilde{a}_m} f'(j, c) + \sum_{j=0}^{h(a_m)} p_{a_m} f'(j, c) \right) \\ &= p_{\tilde{a}_m} f'(h(\tilde{a}_m+1), c) - p_{a_m} f'(h(a_m), c) \\ &= u'_m(\tilde{a}_m, \mathbf{a}_{-m}) - u'_m(a_m, \mathbf{a}_{-m}). \end{aligned} \quad (8)$$

Therefore, \mathcal{G}' is, indeed, an Exact Potential game that has at least one pure strategy NE point, and the result follows. \square

Theorem 4 *The NE point of the game \mathcal{G}' is not necessarily a global optimal point in terms of its capacity and vice versa.*

Proof: We prove Theorem 4 by using a counterexample. Consider a scenario that has two SUs communicating in a network in which there are two channels with idle probabilities $p_1 = 0.3$ and $p_2 = 0.9$, respectively. The optimal system capacity is $p_1 + p_2 = 1.2$, and this is achieved when one SU utilizes Channel 1 and the other utilizes Channel 2. However, the selection is not an NE when CS is enabled because the one that selects Channel 1 can move to Channel 2 to achieve a higher capacity for itself, meaning that the point of optimal system capacity is not an NE point in this case. On the contrary, the NE point in this example is the selection that requires both of the SUs to adopt Channel 2, indicating that the NE point cannot provide optimal system capacity. \square

4 BLA-based Distributed Channel Access Scheme

In the previous section, we analyzed the games encountered in our problem domain, and examined their properties. In what follows, we propose a BLA-based learning scheme to solve the multi-SU multi-channel access problem. The aim of the scheme is to guide the SUs to learn, in a distributed manner, the stochastic property of each channel and to thus converge to an NE point of the game. It should be noted that we assume that *no* prior information about the system is known to the SUs, and that additionally, information exchange among the SUs is not permitted.

Bayesian Learning Automata: Although analytically intractable in many cases, Bayesian approaches provide a standard for optimal decision making. The BLA is a LA which is inherently Bayesian in nature and its features, explained in detail in [1], are omitted here to avoid repetition. Briefly speaking, however, the BLA works in an environment with multiple actions, each of which has a different reward probability. The BLA possesses no prior information about the reward probabilities of the different actions, and any knowledge related to the reward probabilities are “learned” and accumulated through trials. The aim of the BLA is to find out the action that corresponds to the highest reward probability, and it aims to achieve this as efficiently as possible. To attain this goal, the BLA resorts to Bayesian reasoning by maintaining hyperparameter pairs to count the rewards and penalties that each action has received from the random environment. These hyperparameter pairs determine the shape of each of the estimated *a posteriori* reward probability distributions, which, in turn, affects the way the actions are selected in the next iteration. The hyperparameter pairs get updated as the iterations proceed. By virtue of the Bayesian reasoning that is based on the conjugate priors, i.e., by means of the hyperparameters, the BLA achieves computational efficiency, which, combined with Thompson sampling [10], provides convenience in implementation and application.

Reasons of utilization of the BLA: Apart from its convenience in implementation, the reason why we have chosen the BLA as our scheme, is that it can be adapted to the channel selection scenario quite naturally. Firstly, the property of the environment is stochastic. Each channel is either idle or occupied, with a certain probability. Secondly, the question of whether or not a packet is successfully transmitted by an SU can be precisely mapped as a reward or a penalty. Each SU counts the number of rewards and the number of

penalties to learn which channel provides the best chance for a packet to be transmitted.

Reasons for utilizing the Beta distribution as a conjugate prior: In the BLA-based channel access scheme that we shall propose, the Beta distribution is utilized as a conjugate prior. This is determined by the Bernoulli property of the reward probability in the environment. From an SU's perspective, the feedback from a channel would be either a successful transmission or an unsuccessful one, which is clearly, Bernoulli distributed, with the parameter of the Bernoulli distribution possibly changing according to the dynamics of the environment. Based on the theory of Bayesian estimation, the Beta distribution is the conjugate prior for the Bernoulli distribution, i.e., the distribution which renders the Bayesian posterior probability distribution to be of the same form as the prior, and which renders Bayesian reasoning to be tractable in an iterative way.

Definition of a reward and a penalty in different scenarios: For the packet transmission of each SU, in the quiet period at the beginning of each slot, the SU selects a channel and senses its status. If the channel is occupied by its PU, it receives a penalty. Otherwise, the SU will proceed with one of the following two distinct steps depending on whether CS is supported or not.

1. When CS is not supported, the SU will start transmitting after the quiet period on this idle channel. If the transmitting SU is the only SU that selects this channel, the transmission succeeds and a reward is received. Otherwise, the transmission fails due to the occurrence of a collision, and each of the colliding SUs receives a penalty.
2. When CS is supported, if there is only a single SU that wins the contention, the transmission succeeds and the SU receives a reward. When an SU experiences a collision or loses in the channel contention, the transmission fails, and it receives a penalty.

Because of the randomness of the activities of the PUs and the unpredictable behaviors of the other SUs, it is challenging to determine the best channel for any specific SU. However, by equipping each of the SUs with a BLA described as follows, the problem can be solved both elegantly and efficiently.

The formal algorithm is given below.

Algorithm: BLA – Channel Access

Initialization:

- $a_i := b_i := 1$, where $i = 1, 2, \dots, N$, and N is the number of channels. These hyperparameters, a_i and b_i , count the respective rewards and penalties that each channel has received.
- $c := 0$. c counts the number of time slots.

BeginRepeat

1. $c := c + 1$. Draw a value x_i randomly from the following Beta distributions:

$$f(x_i; a_i, b_i) = \frac{\int_0^{x_i} v^{(a_i-1)}(1-v)^{(b_i-1)} dv}{\int_0^1 u^{(a_i-1)}(1-u)^{(b_i-1)} du}.$$

2. List $x_i, i \in [1, N]$ in their descending orders.
3. Select the channel associated with the largest element in the list of $\{x_i\}$. Suppose the selected channel is channel j .
4. Try to transmit in channel j .
If transmission succeeds, it receive a reward; Update ($a_j := a_j + 1$).
If transmission fails, a penalty occurs; Update ($b_j := b_j + 1$).

EndRepeat(ForEver)

End Algorithm: BLA – Channel Access

The beauty of Bayesian learning is that after trying different options, the SU can learn from the environment and adjust its actions accordingly in order to maximize the rewards received. The efficiency of this algorithm will be demonstrated in the next section.

5 Simulation Results

In this section, we present the results we have obtained by simulating the above solution. In the first subsection, we mainly focus on the convergence of the BLA in the Potential games. In the second subsection, we compare the convergence speed of the BLA and the L_{R-I} scheme. In the third subsection, we study the performance of the BLA and the L_{R-I} in terms of the normalized system capacities in different system configurations, where the normalized capacity is the result of dividing the total number of packets that are transmitted for all SUs in the system by the total number of simulated time slots.

Extensive simulations were conducted based on various configurations. Table 1 shows three groups (Conf. 1-3) of transition probabilities, each of which has nine channels, i.e., from Channel 1 (c1) to Channel 9 (c9). We will use below the configuration number and the channel number to indicate a specific simulation

setting. For example, we use Conf. 1 (c1-c3) to represent that the simulation environment consists of Channel 1, Channel 2 and Channel 3 from configuration identified as Conf. 1. When CS is utilized, the contention window size was set to be 16 in our simulations unless otherwise stated.

Table 1: The transition probabilities used in different channels and the corresponding configurations.

Indx	Configurations	c1	c2	c3	c4	c5	c6	c7	c8	c9
d_i	Conf. 1	.9	.8	.7	.6	.5	.4	.3	.2	.1
b_i		.1	.2	.3	.4	.5	.6	.7	.8	.9
d_i	Conf. 2	.4	.4	.4	.4	.5	.5	.5	.5	.5
b_i		.6	.6	.6	.6	.5	.5	.5	.5	.5
d_i	Conf. 3	.2	.1	.8	.2	.2	.5	.1	.3	.1
b_i		.8	.9	.2	.3	.6	.5	.4	.9	.3

5.1 Convergence of the BLA in the Potential Games

We tested the BLA-based algorithm in various environment settings, in each of which we conducted an ensemble of 100 independent experiments, and where each of the experiments had 80,000 iterations/time slots. The results show that the BLA is able to converge to an NE point with a high probability. As more results will be presented in Section 5.3, to avoid repetition, we highlight the cases in Conf. 1 (c1, c2 and c9) to demonstrate that the BLA-equipped SUs converge to an NE point of the game.

Consider Conf. 1, i.e., (c1, c2, and c9) with 3 SUs. When CS is utilized, the NE point of the game is that all SUs converge to Channel 9, while the NE point of the game when CS is not adopted is that each of the SUs converges to a different channel, which is also the global optimal point. The values of the theoretical system capacity at the NE point with CS and without CS (which is also the global optimal point) are 0.9 and 1.2 respectively. The theoretical capacity value is calculated by summing up the static idle probabilities of the channels at a specific point, which, in reality, can be considered to be the upper bound of the capacity at this point.

For an SU, if its probability of selecting one channel surpasses 95%, we say that the SU has converged to this specific channel. If all the SUs converge, we consider that the game has converged. Further, if the game converges to an NE point, we say that the game has converged correctly.

In all the 100 experiments conducted, the game converged correctly. For example, all the SUs converged to Channel 9, which is an NE point, before 80,000 iterations for the configuration when CS is enabled. The normalized capacity achieved was 0.8203, which is fairly close to the theoretical value of 0.9. The gap between the normalized and the theoretical capacities is mainly due to the deviation from the NE point during the learning process before all the SUs converged.

To illustrate, in more detail, the convergence procedure of the algorithm in Conf. 1 (c1, c2, and c9) with 3 SUs and with CS enabled, Fig. 4 depicts a snapshot of the probability of selecting the correct channel (in this case, Channel 9) as the iteration runs up to 600 steps in one experiment. The probability of selecting

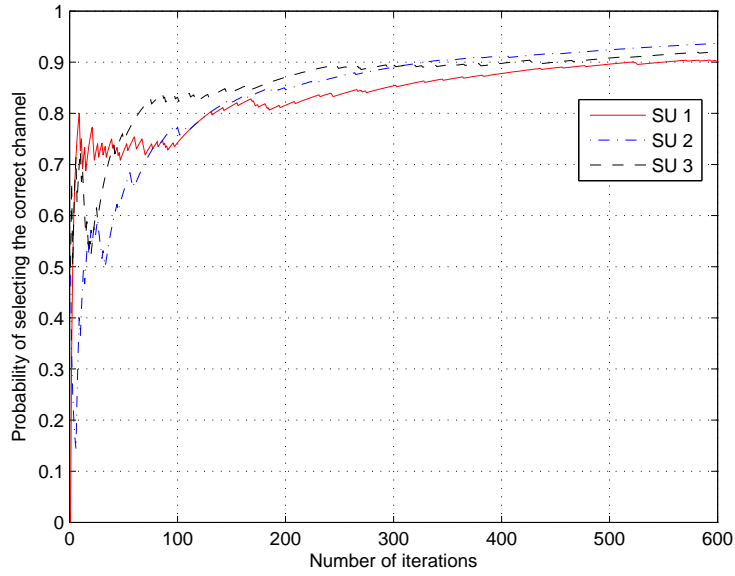


Figure 4: The probability of selecting the correct channel as a function of the number of iterations when CS is enabled for Conf. 1 (c1, c2, and c9).

the correct channel is the quotient of dividing the total number of times that Channel 9 is selected by the total number of iterations for a specific SU. As can be observed from this figure, after the initial exploration, the proposed algorithm displays a trend to converge to the correct channel. Indeed, when we observe the scenario for more iterations, this probability will increase, meaning that the channel selection for the SUs will converge to Channel 9.

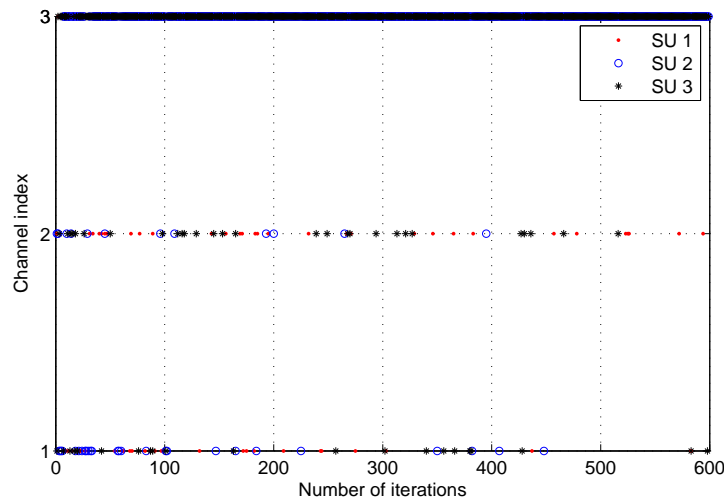


Figure 5: Channels selected by different SUs as a function of the number of iterations when CS is enabled for Conf. 1 (c1, c2, and c9).

Similarly, Fig. 5 describes the convergence procedure from another perspective. In this figure, the selected channel for each SU in each iteration is plotted, where 1, 2, and 3 in the y-axis represent Channel

1, Channel 2 and Channel 9 respectively. Although Channel 1 and Channel 2 are selected relatively often at the initial stage of learning, the number of times that each SU selects Channel 1 and Channel 2 becomes increasingly smaller as the iterations proceed, implying that the SUs learn that Channel 9 is the best channel by using the proposed algorithm.

5.2 Convergence Speed of the BLA and the L_{R-I}

In this subsection, we compare the convergence speed of the BLA and the L_{R-I} in playing the games. We conducted two sets of simulations. The first had two SUs which competed for channels without CS. The other had three SUs and the CS functionality is incorporated. Each simulation set included four different configuration settings, as shown in Tables 2 and 3. In each of the configuration settings, an ensemble of 200 experiments were conducted to reduce the variance of the result.

The simulation results are summarized in the tables, where γ represents the game convergence accuracy probability, and λ denotes the learning parameter for the L_{R-I} scheme. In other words, a game converges correctly with probability γ , where by tuning the parameter λ , the L_{R-I} achieves a certain convergence accuracy.

Table 2: The convergence accuracy probability and the average number of steps for convergence in different configuration settings when CS is disabled and there are two SUs.

LA type	Parameters	Conf. 1 (c1, c2, c9)	Conf. 1 (c1-c4)	Conf. 3 (c1-c4)	Conf. 3 (c1-c5)
BLA	γ	100%	100%	100%	96.5%
	No. of steps	3,466.4	4,296.3	1,632.5	9,372.7
L_{R-I}	γ	99.5%	98.5%	99%	91%
	No. of steps	4,015.7	6,274.2	15,510.4	25,932.6
	λ	0.08	0.05	0.01	0.01

Table 2 shows that the BLA converges faster than the L_{R-I} given the same or even higher accuracy probabilities. For example, in Conf. 1 (c1-c4), the L_{R-I} achieves a convergence accuracy of 98.5% when λ is tuned to be 0.05. The average number of steps it takes for the game to converge is 6,274.2. In the same configuration, the convergence accuracy of the BLA is 100% and the average number of steps for convergence is 4,296.3, which represents an advantage of about 32%. Similarly, in Conf. 3 (c1-c4), the BLA converges with an accuracy of 100% within, on average, 1,632.5 steps, while the L_{R-I} has a convergence accuracy of 99%, for which it consumes on average 15,510.4 steps with the learning parameter $\lambda = 0.01$. In this case, the BLA required only about 10% of the number of iterations when compared to the L_{R-I} scheme! In a more complicated configuration setting, i.e., Conf. 3 (c1-c5), the learning became more difficult as the second best channel and the third best channel possessed similar idle probabilities. Thus the convergence accuracy of the BLA degraded to 96.5%, which, however, is still higher than the accuracy achieved by the L_{R-I} when λ was set to be 0.01. Besides, the average number of steps required by the BLA was again much

less than that required by the L_{R-I} (only 36%), as shown in the table.

It is worth mentioning that the values of λ as assigned in Table 2 are not the optimal learning parameters, i.e., the ones that yield the fastest convergence for a given convergence accuracy of 100%. We did not feel that it was worth our while to determine the optimal parameter in this suite of simulations because the current results have already shown that the L_{R-I} takes more steps than the BLA to converge with an even lower accuracy than the BLA. In other words, if the L_{R-I} has to achieve the same convergence accuracy as the BLA, a decreased value of λ needs to be used, which, in turn, will lower the convergence speed further.

Table 3 shows the results of three SUs competing for channels with CS enabled. As can be seen from the table, the BLA is able to converge with an accuracy of 100% in all these four different configuration settings. Also, the L_{R-I} converged with 100% accuracy when λ was set to the optimal value. Under the optimal learning parameter λ , there are certain cases where the L_{R-I} scheme outperformed the BLA in terms of speed of convergence. For example, in Conf. 3 (c1-c4), the L_{R-I} took on average 1,128.5 steps to converge while the BLA needed an average of 3,937.6 steps. However, when the learning parameter was set to be the same (0.02 in Table 3) for all these configuration settings, the advantage of the BLA over the L_{R-I} becomes clear, as the former achieved a much better trade off between the convergence accuracy and the corresponding speed, than the latter scheme.

Table 3: The convergence accuracy probability and the average number of steps for convergence in different configuration settings when CS is enabled, and when there are three SUs.

LA type	Parameters	Conf. 1 (c1, c2, c9)	Conf. 1 (c2, c3, c5, c8)	Conf. 3 (c1-c4)	Conf. 3 (c1-c5)
BLA	γ	100%	100%	100%	100%
	No. of steps	5,155.3	9,908.8	3,937.6	3,527.1
L_{R-I}	γ	100%	100%	100%	100%
	No. of steps	3,106.6	10,690.7	1,128.5	34,534.3
	λ (optimal)	0.06	0.03	0.15	0.01
	γ	100%	100%	100%	98%
	No. of steps	6,285.7	15,717.7	18,563.3	1,5122.9
	λ (uniform)	0.02	0.02	0.02	0.02

We can summarize these results as follows: When the L_{R-I} is used to play the game, different values of λ yield various values of the convergence accuracy, γ . Equivalently, to achieve a certain convergence accuracy, various system configurations have to adopt different values of λ , as the reader can observe from Table 3. Moreover, the convergence accuracy, in one sense, conflicts with the scheme's convergence speed. A larger learning parameter accelerates the convergence but may decrease the accuracy, while an unnecessarily small learning parameter gains more accuracy but simultaneously leads to a loss in the rate of convergence. In CRNs, it is quite challenging to determine a reasonable or universal learning parameter that compromises well between the accuracy and the speed for all different scenarios. We conclude, therefore, that the L_{R-I} is not totally suitable for CRNs. On the contrary, the Bayesian nature of the BLA provides a much better tradeoff between the learning speed and its accuracy, and more importantly, these tradeoff are achieved

automatically, i.e., without any parameter tuning.

It is also worth mentioning that the L_{R-I} can always achieve a higher convergence accuracy by reducing its learning parameter, λ , at the expense of a slower speed of convergence. For instance, if we consider Table 3, Conf. 3 (c1-c5) with 3 SUs and where CS was permitted, we see that $\lambda = 0.02$ yielded in a convergence accuracy of 98%. However, by tuning λ to be 0.01, the L_{R-I} was able to achieve a convergence accuracy of 100%, with the average number of steps for convergence being significantly larger, i.e., 34,534.3. In other words, if an extremely high accuracy of convergence is required, the L_{R-I} can always achieve this goal, because there is a provision for the learning parameter to be tuned to be smaller while it is still positive³. However, on the other hand, the cost can also be expensive, as a smaller learning parameter leads to a much slower convergence!

While the above-mentioned arbitrarily high convergence accuracy can be achieved by the L_{R-I} , it cannot be expected from the BLA as the latter has no tunable learning parameter. Fortunately, though, the advantage of the BLA is not degraded, as it can, for almost all the scenarios, yield a competitive learning accuracy and speed.

5.3 Comparison of the Capacities of the BLA and the L_{R-I}

In this section, we compare the performance of the BLA and the L_{R-I} by examining their normalized capacities. We organize the numerical results in two parts depending on whether CS was utilized or not, as shown in Tables 4 and 5 respectively. All the results presented are the averaged values obtained from 100 independent experiments each of which had 80,000 iterations. The learning rate for the L_{R-I} is listed in the second column, in parenthesis.

In Tables 4 and 5, GO and NE stand for Global Optimal and the Nash Equilibrium values, respectively. The values in the rows for GO/NE represent the theoretical capacities at those points. Note that when CS is disabled, the GO point is an NE point; but when they are enabled, the GO point does not necessarily have to be an NE point.

As can be seen from Table 4, the capacity achieved by the L_{R-I} is generally lower than that obtained by the BLA, showing that the BLA approach is almost always superior to the L_{R-I} . Besides, the capacity of the BLA is quite close to the theoretical upper bound, which means that in the scenarios where CS is disabled, the BLA is quite efficient in terms of the transmission of the packets.

Table 5 illustrates the simulation results when CS is enabled. Note that in this table, the theoretical capacity at the GO point does not necessarily equal to that at an NE point, and all the normalized capacities achieved by both algorithms tend to approach the NE capacity instead of the GO capacity. This is because both the BLA and the L_{R-I} tend to converge to an NE point, even if there exists a GO point that may yield a superior capacity. Again, the BLA almost uniformly outperforms the L_{R-I} as the normalized capacity

³We refer the reader to [12] for a proof of this statement.

Table 4: The capacity of the BLA with different number of SUs in 9-channel configurations, where CS is not permitted.

Conf.	Alg.	$M=2$	$M=4$	$M=6$	$M=8$
Conf.1	BLA	1.6982	2.9952	3.8891	4.3743
	L_{R-I} (0.005)	1.6729	2.9134	3.7150	4.0400
	L_{R-I} (0.01)	1.6844	2.9565	3.8203	4.2260
	L_{R-I} (0.02)	1.6879	2.9739	3.8593	4.2848
	L_{R-I} (0.05)	1.6676	2.9655	3.8617	4.3377
	L_{R-I} (0.1)	1.6223	2.9216	3.8643	4.3542
(c1-c9)	NE/GO	1.7	3	3.9	4.4
Conf.2	BLA	1.1957	2.3828	3.3859	4.3835
	L_{R-I} (0.005)	1.1868	2.3081	3.2547	4.1339
	L_{R-I} (0.01)	1.1943	2.3207	3.3276	4.2817
	L_{R-I} (0.02)	1.1977	2.3201	3.3622	4.3507
	L_{R-I} (0.05)	1.1870	2.3025	3.3547	4.3802
	L_{R-I} (0.1)	1.1546	2.2666	3.3361	4.3771
(c1-c9)	NE/GO	1.2	2.4	3.4	4.4
Conf.3	BLA	1.6976	3.2441	4.7383	5.8346
	L_{R-I} (0.005)	1.6809	3.1725	4.5634	5.5444
	L_{R-I} (0.01)	1.6865	3.2067	4.6405	5.7234
	L_{R-I} (0.02)	1.6900	3.2177	4.6675	5.7972
	L_{R-I} (0.05)	1.6549	3.2102	4.6362	5.8228
	L_{R-I} (0.1)	1.6262	3.1710	4.6216	5.8156
(c1-c9)	NE/GO	1.7	3.25	4.75	5.85

achieved by the former is higher than that of the L_{R-I} in most cases. Admittedly, with specific M and λ , we can observe capacity values for the L_{R-I} that are larger than the corresponding ones obtained using the BLA, e.g., $M = 16$ and $\lambda = 0.1$. However, as illustrated in this table, such λ values result in inferior performances with other values of M . Again, we emphasize that it is not possible to determine a single value of λ for the L_{R-I} which can, in general, offer a superior performance over BLA. Further, tuning the value of λ on a system configuration basis is unrealistic in the domain of CRNs.

It is worth mentioning that the reason for the higher capacity yielded by the L_{R-I} when $M = 16$ with $\lambda = 0.05$ or 0.1 is because there is *absolutely no* strategy by which the SUs can converge to an NE point with a reasonably high probability. For example, in Conf. 2 with $M = 16$ and $\lambda = 0.1$, in nearly half of the experiments, the SUs cannot converge to the NE point. However, at the non-NE points that the SUs converge to, the capacity is higher than that yielded at the NE point because of the underlying contention. This phenomenon can be explained as follows. At an NE point, a channel that has a higher idle probability will be shared by more SUs, resulting in a correspondingly higher collision probability. Thus, if an SU opts to deviate from an NE point by changing from a channel possessing a higher idle probability to one characterized by a lower idle probability, the corresponding collision probability for the channel with the higher idle probability decreases while that of the channel with the lower idle probability increases. Whenever the capacity improvement due to the reduced collision probability in a higher-idle-probability channel is larger than the decrement due to the increased collision probability in a lower-idle-probability channel, the overall

Table 5: The capacity of the BLA with different number of SUs in 9-channel configurations where CS is permitted. The contention window size is 16.

Conf.	Alg.	$M=4$	$M=8$	$M=12$	$M=16$
Conf.1	BLA	2.9926	3.7788	3.9582	3.9146
	L_{R-I} (0.005)	2.9157	3.6377	3.8116	3.8047
	L_{R-I} (0.01)	2.9592	3.6754	3.8980	3.8602
	L_{R-I} (0.02)	2.9651	3.6806	3.8907	3.8826
	L_{R-I} (0.05)	2.9270	3.6752	3.8864	3.9137
	L_{R-I} (0.1)	2.8061	3.6544	3.8775	3.9379
(c1-c9)	NE/GO	3/3	3.9/4.4	4.2/4.5	4.4/4.5
Conf.2	BLA	2.3872	4.3725	4.7637	4.6432
	L_{R-I} (0.005)	2.3202	4.1873	4.5938	4.5260
	L_{R-I} (0.01)	2.3534	4.3153	4.7107	4.6058
	L_{R-I} (0.02)	2.3541	4.3633	4.7576	4.6383
	L_{R-I} (0.05)	2.3133	4.3658	4.7808	4.6532
	L_{R-I} (0.1)	2.2833	4.2799	4.7677	4.6606
(c1-c9)	NE/GO	2.4/2.4	4.4/4.4	4.9/4.9	4.9/4.9
Conf.3	BLA	3.2438	5.8188	5.6333	5.4860
	L_{R-I} (0.005)	3.1854	5.4397	5.4933	5.3993
	L_{R-I} (0.01)	3.2165	5.5121	5.5868	5.4582
	L_{R-I} (0.02)	3.2245	5.5250	5.6252	5.4789
	L_{R-I} (0.05)	3.2179	5.4869	5.6374	5.4913
	L_{R-I} (0.1)	3.1793	5.2647	5.6006	5.5018
(c1-c9)	NE/GO	3.25/3.25	5.85/5.85	5.85/6.05	5.85/6.05

system capacity increases.

One can also see from Table 5 that in the cases where there are more SUs, the achieved capacity is relatively further away from the theoretical capacity guaranteed by the NE point. The reasons for this are twofold. Firstly, a larger number of SUs implies a more complicated environment requiring more steps for the SUs to converge. Secondly, the processes that invoke CS cannot be considered to be the ideal strategies when the number of SUs increases. Indeed, understandably, more collisions could occur if there are more SUs contending for transmission over the same number of channels. To further explain the impact of different contention window sizes, we illustrate in Table 6 the system capacity with a larger contention window size of 32. As expected, the capacity in Table 6 increases in most cases compared with the results in Table 5 as the collision probability is reduced due to the larger contention window. Again, we can observe a few cases that L_{R-I} with larger learning rates has higher capacity values than those of the BLA approach due to the deviation from the NE point.

We finally investigate the issue of fairness in CRNs. In cases in which CS is not enabled, the SUs tend to stay at or near the NE point after the game has converged. In other words, the SU that has converged to a channel with a higher static idle probability, will always have a better chance for communication, resulting in an unfairness among the SUs. This can be resolved by re-initiating the learning process after a pre-specified time interval. In this way, SUs can take turns to use the different channels and the fairness can be achieved

Table 6: The capacity of the BLA with different number of SUs in 9-channel configurations where CS is permitted. The contention window size is 32.

Conf.	Alg.	$M=4$	$M=8$	$M=12$	$M=16$
Conf.1	BLA	2.9929	3.8254	4.0564	4.0448
	L_{R-I} (0.005)	2.9101	3.6599	3.8984	3.9503
	L_{R-I} (0.01)	2.9498	3.7276	3.9654	4.0053
	L_{R-I} (0.02)	2.9295	3.6896	3.9831	4.0369
	L_{R-I} (0.05)	2.8969	3.6730	3.9770	4.0536
	L_{R-I} (0.1)	2.8310	3.6694	3.9619	4.0635
(c1-c9)	NE/GO	3/3	3.9/4.4	4.2/4.5	4.4/4.5
Conf.2	BLA	2.3879	4.3726	4.8175	4.7607
	L_{R-I} (0.005)	2.3304	4.1915	4.6449	4.6431
	L_{R-I} (0.01)	2.3595	4.3098	4.7671	4.7251
	L_{R-I} (0.02)	2.3455	4.3555	4.8132	4.7581
	L_{R-I} (0.05)	2.3245	4.3474	4.8358	4.7749
	L_{R-I} (0.1)	2.2774	4.2341	4.8335	4.7790
(c1-c9)	NE/GO	2.4/2.4	4.4/4.4	4.9/4.9	4.9/4.9
Conf.3	BLA	3.2441	5.8145	5.7333	5.6618
	L_{R-I} (0.005)	3.1842	5.4242	5.5926	5.5715
	L_{R-I} (0.01)	3.2189	5.5027	5.6855	5.6318
	L_{R-I} (0.02)	3.2271	5.5276	5.7254	5.6565
	L_{R-I} (0.05)	3.2227	5.3856	5.7269	5.6686
	L_{R-I} (0.1)	3.1843	5.3204	5.7106	5.6755
(c1-c9)	NE/GO	3.25/3.25	5.85/5.85	5.85/6.05	5.85/6.05

statistically. On the contrary, when CS is enabled, the fairness among the SUs is improved because the co-channel SUs can share the channel access opportunities.

Before we move onto the conclusions, we would like to discuss the limitations of our work and the potential avenues for future work. First of all, the size of the contention window is currently fixed for the CS-enabled scheme. A smaller collision probability can be achieved by each SU if the size of the contention window can be further extended when collision occurs more often. Secondly, although described as a *stationary* stochastic process, the parameters in this model that describe the behavior of the PUs are not time-variant. In reality, the behavior of PUs can be time-varying. For example, the channel may be occupied more often in business hours than at midnight. Therefore, it would also be interesting to study the performance of the proposed approach and other existing schemes in time-varying (i.e., non-stationary) environments. Thirdly, throughout this study, we have assumed that the behaviors of the PUs are homogeneous to all SUs. It would be a very interesting and non-trivial task to study the case where one channel occupied by a PU is considered to be truly occupied by a specific SU, while it is reckoned to be idle by another SU due to, for example, the geographic differences between SUs.

6 Conclusions

This paper studies the channel selection problem in CRNs when multiple SUs exist. The problem includes two channel access strategies, i.e., when CS is enabled, and when it is not. Each of the strategies has been formulated as an Exact Potential game, and a BLA-based approach is presented to play these games. Simulation results show the advantages of the BLA scheme from four aspects. Firstly, the BLA is able to converge to the NE point of the game with a high accuracy. Secondly, when CS is disabled, the cost paid by the BLA before converging to the NE point is less than what the L_{R-I} algorithm demands. Thirdly, when CS is permitted, the BLA, though it does not necessarily converge faster than the L_{R-I} under its optimal learning parameter, has a significant advantage that it does not have to tune any parameter to achieve a good tradeoff between its learning speed and accuracy. Therefore, under the condition of playing on a fair field, i.e., when neither the BLA nor the L_{R-I} has any prior knowledge of the environment for parameter tuning, the BLA is superior to the L_{R-I} in the sense that the former balances learning accuracy and speed automatically. Finally, the BLA has been shown to be able to achieve a normalized system capacity close to the theoretical capacity value attainable at the NE point.

References

- [1] O. C. Granmo, "Solving two-armed Bernoulli bandit problems using a Bayesian learning automaton," *International Journal of Intelligent Computing and Cybernetics*, Vol. 3, No. 2, 2010, pp. 207-234.
- [2] O. C. Granmo and S. Glimsdal, "Accelerated Bayesian learning for decentralized two-armed bandit based decision making with applications to the Goore game," *Applied Intelligence*, Vol. 38, No. 4, 2013, pp. 479-488.
- [3] IEEE 802.22 WG, "IEEE standard for wireless regional area networks—Part 22: Cognitive wireless RAN medium access control (MAC) and physical layer (PHY) specifications: Policies and procedures for operation in the TV bands," IEEE Std., Jul. 2011.
- [4] L. Jiao, X. Zhang, O. C. Granmo and B. J. Oommen, A Bayesian Learning Automata-based distributed channel selection scheme for cognitive radio networks, *Proceedings of IEA/AIE'14, the 2014 International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems*, Kaohsiung, Taiwan, June 2014, pp. 48-57.
- [5] D. Monderer and L. S. Shapley, "Potential games," *Games and Economic Behavior*, Vol. 14, 1996, pp. 124-143.
- [6] S. Lakshmivarahan, *Learning Algorithms Theory and Applications*. New York: Springer-Verlag, 1981.

- [7] Y. C. Liang, K. C. Chen, G. Y. Li, and P. Mahönen, “Cognitive radio networking and communications: An overview,” *IEEE Trans. Veh. Technol.*, Vol. 60, No. 7, Sept. 2011, pp. 3386-3407.
- [8] K. S. Narendra and M. A. L. Thathachar, *Learning Automata: An Introduction*. Prentice Hall, 1989.
- [9] Y. Song, Y. Fang, and Y. Zhang, “Stochastic channel selection in cognitive radio networks,” in *IEEE Global Telecommunications Conference*, Washington DC, USA, Nov. 2007, pp. 4878-4882.
- [10] W. R. Thompson, “On the likelihood that one unknown probability exceeds another in view of the evidence of two samples,” *Biometrika*, vol. 25, pp. 285–294, 1933.
- [11] T. A. Tuan, L. C. Tong, and A. B. Premkumar, “An adaptive learning automata algorithm for channel selection in cognitive radio network,” *Proceedings of the IEEE International Conference on Communications and Mobile Computing*, Shenzhen, China, Apr. 2010, pp. 159-163.
- [12] Y. Xu, J. Wang, Q. Wu, A. Anpalagan, and Y. -D. Yao, “Opportunistic spectrum access in unknown dynamic environment: A game-theoretic stochastic learning solution,” *IEEE Trans. on Wirel. Commun.*, Vol. 11, No. 4, 2012, pp. 1380-1391.
- [13] X. Zhang, L. Jiao, O. C. Granmo, and B. J. Oommen, “Channel selection in cognitive radio networks: A switchable Bayesian Learning Automata approach,” *Proceedings of PIMRC'13, the 2013 IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, London, UK, Sept. 2013, pp. 2372-2377.