Q-CE: Self-Organized Cognitive Engine based on Q-Learning

Ali H. Mahdi, Zeeshan Ansar, Stephen Mwanje, Oleksandr Artemenko, Andreas Mitschele-Thiel
Integrated Communication Systems Group
Technische Universität Ilmenau
Ilmenau, Germany
Email: ali.mahdi@tu-ilmenau.de

***Abstract*—One of the main challenges in Cognitive Radio Networks (CRNs) is that the dynamic radio environment affects the Quality of Service (QoS) requirements of Secondary Users (SUs). Another challenge is how to predict Primary Users (PUs) activities over licensed channels, to avoid interfering with PUs. So, there is a need to implement a Cognitive Engine (CE) as a self-organized entity in a CR that overcomes those challenges. In our previous studies, an algorithm called Adaptive Discrete Particle Swarm Optimization (ADPSO) combined with Case-Based Reasoning (CBR) has been proposed for CE. ADPSO selects the optimal configurations when an unknown environment is countered while CBR allows usage of previous knowledge in an environment that has been previously observed. CBR however depends on a single observation of the previous state and gives inaccurate results where an individual states' performance changes. Another problem is how to find the best action when the environment is changing dynamically. In this paper, we propose a self-organized Q-Learning-based CE (Q-CE) which: 1) autonomously adapts the link configuration; 2) applies the previous action under similar environments and 3) where the environment changes, Q-CE learns from radio environment behavior and PU activities the best action to apply, in order to achieve QoS requirements and avoid interfering with PUs. The proposed CE combines following methods: ADPSO for link configuration; CBR for fast reasoning under similar environment; and Q-Learning to learn the environmental behavior. The results show improvements of about 67% in the achieved throughput, about 50% in signaling overhead when compared with the previous solutions that use only ADPSO and CBR.**

keywords - Cognitive Radio; Optimization; Learning; Self-Organization; Simulation

I. Introduction

A Cognitive Radio (CR) is a smart wireless communication entity that can be adapted in response to its environment using spectrum awareness and radio reconfigurability, resulting in high spectral efficiency. While the intelligence is the main feature of CR, the Cognitive Engine (CE) is the key for implementing intelligence in CR. The main task of the CE is to learn the spectrum behavior and reconfigure the radio system parameters according to spectrum behavior. The radio system parameters include waveform, operating frequency, etc. According to [1], the CE can be defined as an intelligent agent that manages the cognition tasks in a CR, where intelligence denotes behavior that is consistent with a specified goal. The CE continuously monitors the performance of the unlicensed users also called Secondary Users (SUs), in addition to reading the radio's outputs. It then uses this information to deduce the observations of radio environment, e.g. channel conditions, link performance, etc. that can be used to provide the required Quality of Service (QoS) subject to an appropriate combination of user requirements, operational limitations, and regulatory constraints. Another task of CE is to learn from radio environment behavior the characteristics of the spectrum bands so as to determine the best action, which would avoid interfering with licensed users, also called Primary Users (PUs) and achieve the QoS requirements.

In the past few years, some work has been carried out in designing the CE based on Artificial Intelligence (AI) approaches. In [2], the author proposed CE based on the use of Genetic Algorithm (GA) as the optimization algorithm and Case Based Decision Theory (CBDT) for reducing search space for GA. In [3], the author implemented GA and Rule-Based System (RBS) for optimizing link configuration and fast reasoning in order to overcome the inter-dependency of objective function in physical layer. The weakness of [2] and [3] is the long convergence time in addition to no learning process from radio environment behavior. The work in [4] presented CE based on reinforcement learning for decision making. It however can not achieve fast reasoning due to convergence time required to find the best link configuration. In [5], the authors applied Case-Based Reasoning (CBR) to use past experience in the form of state-action pairs for future actions to speed up the convergence. Nevertheless, Reinforcement Learning (RL) presents an opportunity/good solution for learning from the environment. For example, in [6], the authors used reinforcement learning for adapting modulation scheme and code rate in Orthogonal Frequency Division Multiplexing (OFDM) wireless systems, while [7] applied Q-Learning (QL) for managing aggregated interference generated by multiple Wireless Rural Area Network (WRAN) systems. However, in [6] and [7], applying RL/QL for adapting link configuration in CRs leads to slow down the convergence due to proposing different link configuration until it reaches the state of convergence, especially under new observations from radio environment.

The goal of this work is to design a self-organized QL based CE (Q-CE) which can autonomously adapt the link configuration according to any change in the surrounding radio environment, in order to achieve QoS requirements and to avoid interfering with PUs. We propose a new design for CE which consists of three entities: Cross-Layer Optimization (CLO), Reasoning and Learning. We use Adaptive Discrete Particle Swarm Optimization (ADPSO) algorithm [8], [9] for CLO. For reasoning, we use CBR [10]. For learning entity, we use QL algorithm. The results show how the Q-CE

978-1-4799-3086-9/14/$31.00 ©2014 IEEE

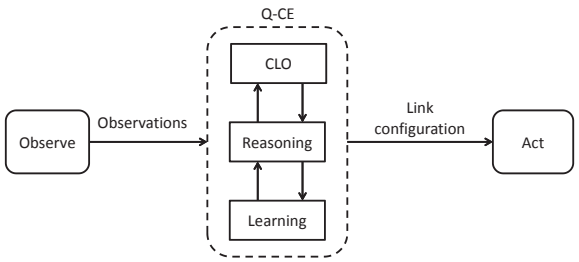

Fig. 1. Q-CE architecture and it's interactions with other processes

autonomously adapts the link during communication process under variable PU activity to achieve the highest throughput and lowest signaling overhead, comparing to use CE based on individual AI-approach. Also, the results show how our proposed CE applies online learning to access the best channel without interfering with PUs.

The structure of this paper is organized as follows: our proposed Q-CE architecture is explained in Section II. The implementation of QL is presented in section III .Our simulation scenario is presented in section IV. Section V discusses the results of our simulated scenario. Finally, concluding remarks and observations are presented in Section VI.

II. Proposed Architecture for Q-CE

Our proposed architecture for Q-CE consists of three entities: 1) CLO, which adapts parameters from different layers to achieve desired link objectives (e.g. high throughput, low power consumption, etc.); 2) Reasoning, which is used to apply past experience in the form of state-action pairs for future actions to speed up the convergence; 3) Learning, which is used to study the behavior of the radio environment over time to find the best action which achieves QoS requirements.

Figure 1 Shows the proposed architecture of Q-CE and its interactions with other processes in the cognition cycle.

A. Cross-Layer Optimization (CLO)

SU transmitter uses ADPSO [9] as a CLO algorithm to derive a link configuration according to the observed link state. Q-CE gets the observations about the radio environment (such as Noise σ_n, Loss ℓ, free channel ch) and information about channel characteristics (i.e. channel availability f_{ChAv} and probability of packet loss $f_{PacLoss}$ of the free channels, then, it runs ADPSO to propose the best possible link configuration across the free channels. ADPSO defines a link configuration which achieves a trade off between maximizing achievable throughput f_{thr} and minimizing achievable transmission delay f_{delay}. In order to achieve the aforementioned objectives, ADPSO optimizes the link configuration in terms of the transmission parameters such as packet length L, transmit power P_t, free channel ch and modulation scheme K. The aggregate objective is defined in terms of the total fitness function f_{total} in Eq. 1:

$$f_{total} = w_1 f_{thr} + w_2 f_{delay} + w_3 f_{ChAv} + w_4 f_{PacLoss} \quad (1)$$

where, f_{total} is the total fitness function at channel ch, $f_{total} \in [0,1]$, w_1,w_2,w_3 and w_4 are the weights of each objective function.

The optimization algorithm evaluates each proposed link configuration as follows [9]:

$$f_{delay} = \frac{\left(\frac{1}{K_{max} R_s}\right) L_{min}}{\left(\frac{L}{R_b}\right)} \quad (2)$$

$$f_{thr} = \frac{\left(\frac{L}{L+Ov}\right) PSR\ R_b}{\left(\frac{L_{max}}{L_{max}+Ov}\right) R_{bmax}} \quad (3)$$

The components of Eq. 2 and 3 are:

$$PSR = (1 - BER)^{L+Ov} \quad (4)$$

$$R_b = R_s K \quad (5)$$

where,
BER is the probability of bit error rate,
R_s is the symbol rate,
R_b is the bit rate,
R_{bmax} is maximum bit rate,
L_{max} is the maximum packet length,
L_{min} is the minimum packet length,
BER_{max} is the maximum acceptable BER,
Ov is the overhead in the data packet.

To achieve QoS requirements, the following has to be achieved: 1)$BER \leq BER_{max}$; 2) $R_b \geq R_{bmin}$. ADPSO proposes a link configuration which has highest f_{total} and achieves the QoS requirements. Then, CE sends the proposed link configuration to the SU receiver for validation in response to its current radio environment.

B. Reasoning

One of the main tasks of CE is to decide and instantaneously act so as to achieve fast link adaptation between SUs, in accordance to the observations and application requirements. Meanwhile, owing to the dynamic radio environment, there is a need to propose a set of pre-computed actions that achieve the requirements under the different observations. So, reasoning is one of the methods which can be used to perform fast decision for CR under radio environment with repeated manner. One of the most famous approaches for making fast reasoning is CBR which can be used to save some expected observations and link configurations in form of state-action pairs. In state-action pairs, the state part includes the frequently occurred free channels j in addition to σ_n and ℓ of the free channels. The action part includes best link configuration such as P_t, K, L, and f_{total} of the proposed link configuration.

An example operation of ADPSO and CBR (shown in Fig. 2) can be broken down as follows: 1) it determines the current observations (e.g. σ_n=88 dBm, ℓ=60 dB, Channel ID =1); 2) searches in CBR for state-action pair similar to the current observations; 3) selects the state-action pairs in CBR of the same channel ID and similarity of 90% to the current σ_n and ℓ; 4) uses the action as a proposed link configuration; 5) if there is no similarity, runs ADPSO to determine a new link configuration [10].

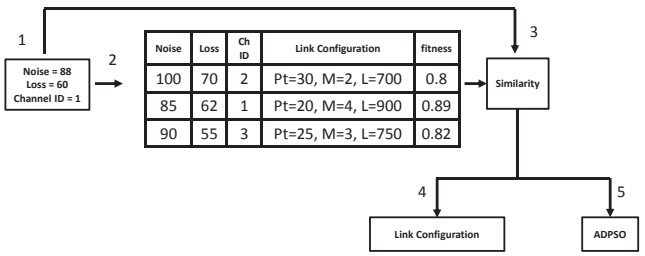

Fig. 2. Operational procedure of ADPSO+CBR [10]

III.Q-LEARNING: LEARNING THE OPTIMAL CONFIGURATION

In this section, we explain briefly the Reinforcement Learning solution. Then, we describe our proposed QL model for applying learning in Q-CE. Later, we explain the procedure of operation Q-CE and the revised negotiation process between CRs at transmitter and receiver.

A. Reinforcement Learning

Reinforcement Learning (RL) is a machine learning solution where, through interaction with its environment, an agent learns optimal behavior for certain states of the environment [11]. The RL problem is usually modeled as a Markov Decision Process (MDP) and solved using Dynamic Programming (DP) techniques. DP techniques however require precise RL problem models, which is seldom possible. Q-Learning (QL) is model free RL which uses the Temporal Difference (TD) method to solve model-less RL problems by estimating a value function, called the Q-Value, for each state-action pair. The Q-Value estimates the expected reward of taking an action $a \epsilon A$ from an action set A when starting from state s_t and following a fixed policy π.

With CBR, we are able to select the best among a set of previous states with similar conditions as the observed state. Each of these would however have been used only once and may not represent the best overall solution in case of significant changes in the radio environment. We therefore require to try the possible solutions a number of times (in order to ensure the absolute accuracy of the solution). The solution is a Q-Learning that learns the best action a at t, that maximizes the instantaneous reward r at time $t+1$.

B. Proposed QL Approach

We modified the QL model given in [7] according to our system. The learned value includes the estimated future value in traditional QL while the estimated future value is obtained through the fixed states. In our system the state is not fixed as states represent ch and their characteristics (i.e. σ_n and ℓ).

The CBR consists of the states and actions. The fitness value of each action performed on states is stored. Here, the fitness value describes the quality of the action performed for the specific problem. The reward (performance history) is also stored for each applied action. By utilizing the information of fitness and performance history the learner can make better decisions about channel selection and transmission parameters.

Our proposed version of QL is to re-compute the current fitness value of the selected action by evaluating it on both transmitter and receiver sides f_{ct} and f_{cr}. Then, update the fitness value according to the following equations:

$$f_{cavg} = \frac{f_{ct} + f_{cr}}{2} \tag{6}$$

$$f_u = f_{cavg} + \alpha \left(f_{cavg} - f_p \right) \tag{7}$$

where,
f_u is the updated fitness value for the proposed action,
f_{cavg} is the average current fitness value,
f_p is the previous fitness value,
α is the learning rate.

For each link configuration, the QL computes a reward, r for the proposed channel according to 1) the number of times, s, that the channel is free at both the transmitter and receiver; and 2) the number of times, acc, that the transmitter evaluates the channels for link configuration:

$$r = \frac{s}{acc} \tag{8}$$

QL then updates the reward r_u, using the calculated reward r and the previous reward r_p

$$r_u = r + \alpha \left(r - r_p \right) \tag{9}$$

where r_p is the previous reward of the channel.

C. Revised Negotiation Procedure

In order to validate the proposed link configuration at the receiver side, the transmitter sends the proposed configuration to the receiver in form of a control packet to validate it. The transmitter and the receiver use the two way handshaking mode like in IEEE 802.11 MAC over Common Control Channel (CCC) to exchange the proposed link configuration [12].

We modified the negotiation procedure in [10] to apply QL approach. By applying QL, Q-CE updates the fitness and reward values for each proposed action according to the history of the radio environment. Thus, CE can select the best decision according to the history of the channel and the radio environment.

Figure 3 shows a flow chart which describes the operation of decision making in QCE and the negotiation between CRs at transmitter and receiver.

The procedure of the revised negotiation is as follows:

1) After sensing frame, Q-CE gets the observations (ch, ℓ, σ_n) at CR transmitter side.
2) Q-CE triggers CBR to search for state-action pairs of same channel ID and selects those whose states are similar to the current observations.
3) If no similarity has been found in CBR or the similarity between the current observations and the observations in CBR is less than 90%, Q-CE runs ADPSO algorithm to determine a new link configuration. Otherwise, Q-CE selects the actions of the similar observations.
4) The Q-CE at transmitter side re-computes the fitness value of each proposed action f_{ct} (from CBR) in response to its current observations (σ_n and ℓ) as in Eq. 1.
5) The transmitter sends selected actions to the receiver

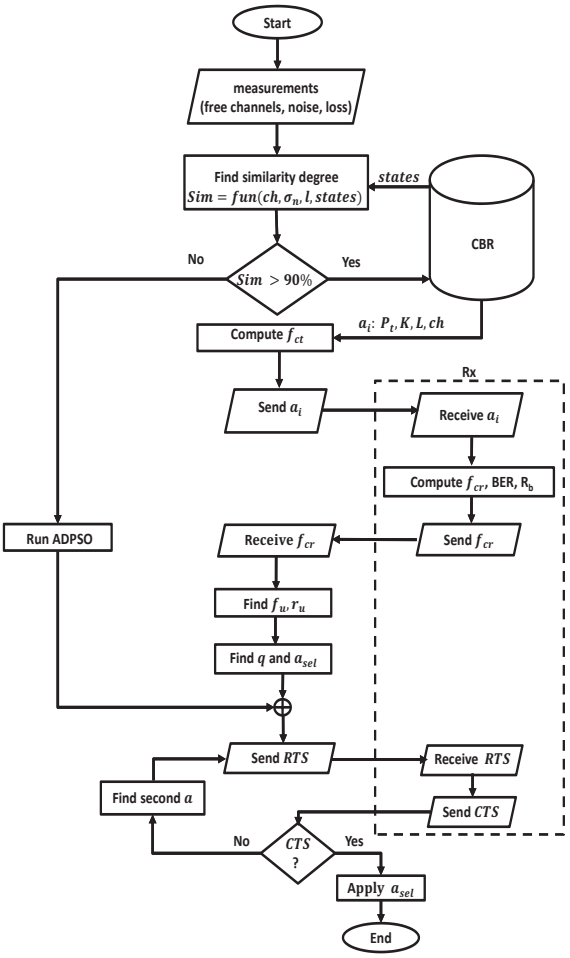

Fig. 3. Decision making and negotiation procedure in Q-CE

side within Ready To Send (RTS) packet. The receiver validates the proposed actions according to its radio environment. If the proposed actions achieve QoS requirements, the receiver calculates the fitness value f_{cr} using the same fitness function in Eq. 1.

6) The receiver sends back f_{cr} to the transmitter with in Clear To Send (CTS) packet.
7) Q-CE at transmitter side calculates f_u of each proposed action as in Eq. 6 and 7.
8) Q-CE updates the reward r_u of each proposed action as in Eq. 8 and 9.
9) For each proposed action, Q-CE computes its quality q_i as follows:

$$q_i = w_1 \ f_u + w_2 \ r_u \tag{10}$$

where, w_1 and w_2 are the weights of fitness and reward.

Then, Q-CE selects the action a_{sel} with highest q value:

$$a_{sel} = max(q_1, q_2, ...) \tag{11}$$

10) In order to ensure that a_{sel} is still valid at receiver side, the transmitter sends a_{sel} in RTS packet to the receiver. The receiver validates a_{sel} according to QoS requirements (BER and R_b) in addition to the availability of the proposed channel. If a_{sel} is validated, receiver sends CTS.
11) If the Q-CE at the transmitter receives CTS, CBR updates the fitness and reward of a_{sel} to f_{up} and rwd_{up}, and applies a_{sel} to CR. Otherwise, Q-CE selects the action with second highest Q, and repeats the validation process with the receiver side.

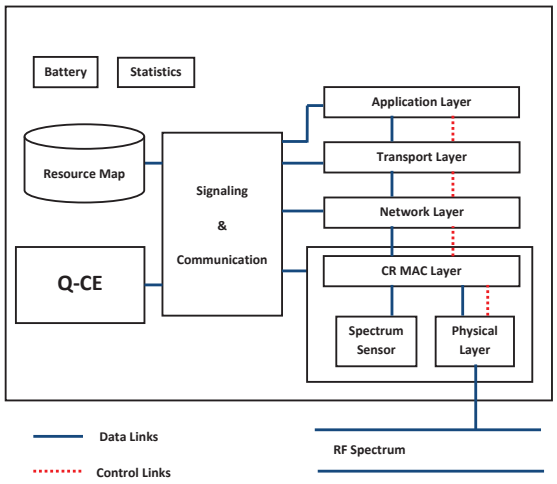

Fig. 4. Architecture of CR node

IV. Simulation Environment and Scenario

A. Simulator

To implement the proposed CE design we used the crSimulator, a discrete ad-hoc networks simulation model for CR in OMNeT++ developed by the ICS Group at the Technische Universität Ilmenau [13]. The crSimulator is the most suitable simulator to carry out this work as it provides a component based node architecture; where every component in the node can be implemented and modified according to its functionality.

The crSimulator provides layered architecture for implementing communication protocols at different layers. The basic architecture of the CR node, as shown in Fig. 4, provides the components required for CR operations including sensing module and Q-CE [14], as well as conventional communication layers including the network, transport and application layers and combined with CR specific/modified MAC and physical layers. Owing to the need for dynamic channel availability, the MAC layer is reinforced with support for channel handovers that are not required in conventional wireless networks. To perform channel handovers MAC layer has to be tightly interfaced with the sensing module that can inform the MAC layer about the current spectrum knowledge [13]. The spectrum sensing module continuously senses all the available channels and informs the CRMAC layer, the resource map and the Q-CE module about the PU activity. The resource map gets the sensing information and makes statistics about the radio environment, such as channel availability $ChAv$ and probability of packet loss $PacLoss$, in order to use them in the Q-CE's decision making process.

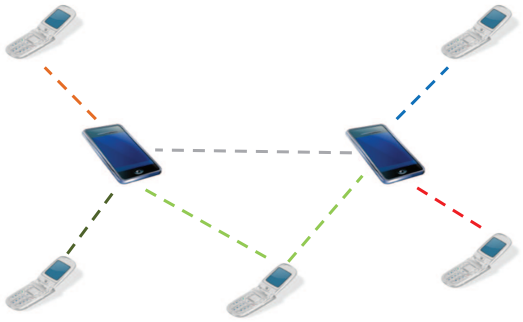

Fig. 5. System Scenario

TABLE I. NETWORK ASSUMPTIONS

Parameters	Value
Channel type	AWGN
R_s (ksps)	50
α	0.5
R_{PU} (sec)	[0.1, 1.0]
PU_t (sec)	[0.2, 1.0]
ch	[1, 5]
σ_n (dBm)	[-85, -100]
ℓ	[60, 90]
R_b	100 kbps
BER	10^{-4}
P_t (dBm)	[0, 25]
K	BPSK,QPSK,8PSK, 16QAM,64QAM
L	[100, 1000]

B. Simulation Scenario

The CR Ad-Hoc Network (CRAHN) is generated using the OMNeT++ topology generator. Our simulated network consists of five PUs and two SUs as shown in Fig. 5. Each SU is equipped with two transceivers: the first transceiver is operating over the CCC to exchange control signals. The second one is using a Software Defined Radio (SDR) module that can tune to any of the data channels to sense for the unused spectrum and moreover receive/transmit the SUs' date packets [12].

In Table I, we define the ranges of our network assumptions. The inputs to Q-CE are: free channels index ch, σ_n and ℓ. QoS parameters are: BER and R_b. The output parameters (link configuration parameters) are: P_t, K,L and ch.

R_{PU} represents the time between the release and the access of PU to the channel ch. PU_t represents the time between the access and the release of PU to the channel ch [13]. We implement these parameters as inputs to the network, in order to study the influence of unstable channel on the decision making and learning process.

V. RESULTS AND DISCUSSION

In this section, we evaluate the results using 3 metrics: 1) the achieved throughput, 2) the signaling overhead and 3) the channel utilization which represents the percentage of time during which a channel is utilized.

Figure 6 shows the achievable throughput by the proposed

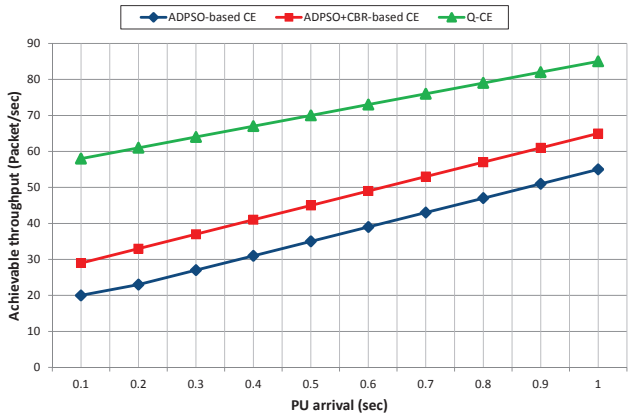

Fig. 6. Achieved throughput against PU arrival rate

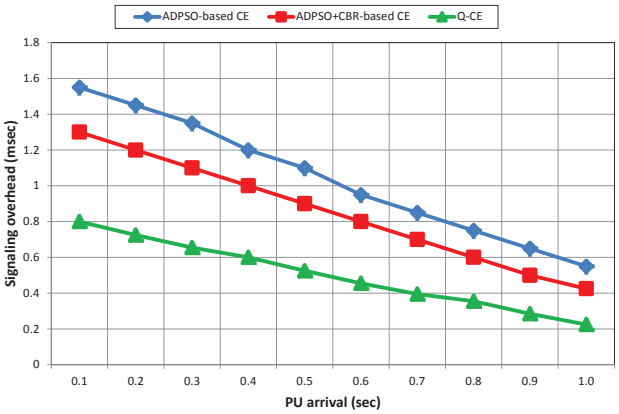

Fig. 7. Signaling overhead against PU arrival rate

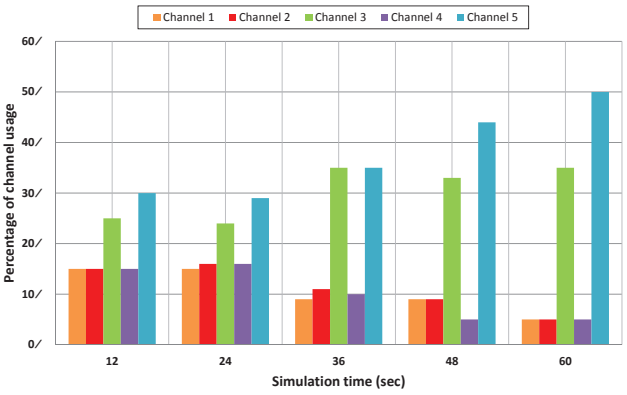

Fig. 8. Channel usage against simulation time

Q-CE, ADPSO-based CE [9], and ADPSO+CBR-based CE [10] against R_{PU}. Compared to the ADPSO-based CE & the ADPSO+CBR-based CE, Q-CE achieves improvements in throughput of respectively 67% and 50%. The reason of the achieved improvement is the usage of QL which, using historical performance, selects transmission parameters that offer the best performance under changing environmental conditions.

Figure 7 shows the signaling overhead by the proposed Q-CE, ADPSO-based CE, and ADPSO+CBR-based CE as a function of R_{PU}. Compared to the ADPSO-based CE & the ADPSO+CBR-based CE, Q-CE achieves improvements in

throughput of respectively 50% and 38%. As noted above, the use of history data by Q-CE ensures that the transmitter proposes transmission parameters with the best probability of acceptance at the receiver.

In Figure 8, we show the channel utilization during the entire simulation. We observe that the channel utilization increases by up to 20% on the channels with least PU activity while decreasing on channels with high PU activity. This happens effectively due to the ability of the QL to learn the amount of activity over time and consistently propose these low PU activity channels. So, by implementing QL in Q-CE the channel with less PU arrival rate gets a higher reward as in Eq. 9, while the channel with higher PU activity gets low reward every time Q-CE attempts to access one of the channels.

VI. Conclusion

In this paper, we proposed a self-organized Q-CE architecture for dynamic spectrum access. The proposed Q-CE is the result of merging ADPSO, CBR and QL in order to achieve fast reasoning with best link configuration and learning from the channels behavior to select the most reliable action a_{sel}. a_{sel} achieves QoS requirements and avoids interfering with PUs. The proposed Q-CE improves the achieved throughput and signaling overhead, in addition it does the analysis of study the channel behavior and learns (on-line) which channel is best to access without interfering with PUs.

Acknowledgment

This work has been carried out within the International Graduate School on Mobile Communications (Mobicom) at Technische Universität Ilmenau, supported by German Academic Exchange Service (DAAD), Carl Zeiss Foundation and University of Baghdad.

References

[1] A. He, K. K. Bae, T. Newman, J. Gaeddert, K. Kim, R. Menon, L. Morales-Tirado, J. Neel, Y. Zhao, J. Reed, and W. Tranter, "A survey of artificial intelligence for cognitive radios," *Vehicular Technology, IEEE Transactions on*, vol. 59, no. 4, pp. 1578–1592, 2010.

[2] T. W. Rondeau, "Application of artificial intelligence to wireless communications," Ph.D. dissertation, Virginia Polytechnic Institute and State University, 2007.

[3] T. R. Newman, "Multiple objective fitness functions for cognitive radio adaptation," Ph.D. dissertation, Department of Electrical Engineering & Computer Science and the Faculty of the Graduate School of the University of Kansas, 2008.

[4] T.Jiang, "Reinforcement learning-based spectrum sharing for cognitive radio," Ph.D. dissertation, University of York, September 2011.

[5] K.-S. Huang, C.-H. Lin, and P.-A. Hsiung, "A space-efficient and multi-objective case-based reasoning in cognitive radio," in *IET International Conference on Frontier Computing Theory, Technologies and Applications*, August 2010, pp. 25–30.

[6] J. Leite, P. H. P. De Carvalho, and R. Vieira, "A flexible framework based on reinforcement learning for adaptive modulation and coding in ofdm wireless systems," in *Wireless Communications and Networking Conference (WCNC), 2012 IEEE*, 2012, pp. 809–814.

[7] A. Galindo-Serrano and L. Giupponi, "Distributed q-learning for aggregated interference control in cognitive radio networks," *Vehicular Technology, IEEE Transactions on*, vol. 59, no. 4, pp. 1823–1834, 2010.

[8] A. Mahdi, J. Mohanan, M. Kalil, and A. Mitschele-Thiel, "Adaptive discrete particle swarm optimization for cognitive radios," in *IEEE International Conference on Communications (ICC)*, June 2012, pp. 6550 –6554.

[9] A. Mahdi, M. Kalil, and A. Mitschele-Thiel, "Cross layer optimization for efficient spectrum utilization in cognitive radios," in *International Conference on Computing, Networking and Communications (ICNC)*, January 2013, pp. 305–309.

[10] A. H. Mahdi, M. A. Kalil, and A. Mitschele-Thiel, "Dynamic packet length control for cognitive radio networks," in *Vehicular Technology Conference (VTC Fall), 2013 IEEE 78th*, 2013, pp. 1–5.

[11] A. Amanna, M. J. Price, S. Bera, M. Gadhiok, and J. H. Reed, "Cognitive engine architecture for railway communications," *LEARNING*, vol. 1000, p. 1, 2010.

[12] M. Kalil, A. Puschmann, and A. Mitschele-Thiel, "Switch: A multi-channel mac protocol for cognitive radio ad hoc networks," in *IEEE Vehicular Technology Conference (VTC Fall)*, 2012, pp. 1 – 5.

[13] S. Khan, M. Kalil, and A. Mitschele-Thiel, "crsimulator: A discrete simulation model for cognitive radio ad hoc networks in omnet," in *Wireless and Mobile Networking Conference (WMNC), 2013 6th Joint IFIP*, 2013, pp. 1–7.

[14] A. Puschmann, S. Khan, A. Mahdi, M. Kalil, and A. Mitschele-Thiel, "An architecture for cognitive radio ad-hoc network nodes," in *International Symposium on Communications and Information Technologies (ISCIT), 2012*, oct. 2012, pp. 394 –398.