

Hybrid Cognitive Engine for Radio Systems Adaptation

Ismail AlQerm and Basem Shihada

CEMSE Division, King Abdullah University of Science and Technology, Saudi Arabia,
{ismail.qerm, basem.shihada}@kaust.edu.sa

Abstract—Network efficiency and proper utilization of its resources are essential requirements to operate wireless networks in optimal fashion. Cognitive radio aims to fulfill these requirements by exploiting artificial intelligence techniques to create an agent called cognitive engine. Cognitive engine exploits awareness about the surrounding radio environment to optimize the use of radio resources by adapting relevant transmission parameters. In this paper, we propose a hybrid cognitive engine that exploits Case Based Reasoning (CBR) and Decision Trees (DTs) to perform radio adaptation in multi-carriers wireless networks. The engine complexity is reduced by employing DTs to improve the indexing methodology used in CBR cases retrieval. The performance of our hybrid engine is validated using software defined radios implementation and simulation in multi-carrier environment. The system throughput, signal to noise and interference ratio, and packet error rate are obtained and compared with other schemes in different scenarios.

Index Terms—Cognitive engine, decision-trees, case-based reasoning software-defined radio (SDR).

I. INTRODUCTION

Future wireless networks will be subjected to performance degradation due to the increasing spectrum demand and absence of sophisticated radio resource management systems. For instance, in IEEE 802.11 network [1], there are several different devices with multiple services operating in limited spectrum. Therefore, the proposed solution for resource management and system adaptation for such network must be aware of the environment conditions and dynamically adapts the radio parameters to match with network conditions variation. Cognitive Radio (CR) came into practice with a vision to improve the network efficiency by re-configuring radio parameters according to the environment obtained conditions using designated cognitive engines. Cognitive engine integrates artificial intelligence algorithms with radio platforms to decide radio parameters configuration based on environment observation, and to learn from experience. However, integrating those algorithms with radio platforms to achieve competent performance with minimum complexity is a difficult task.

There are several attempts in the literature to design cognitive engine with different artificial intelligence techniques. An adaptive multi-objective optimization was proposed in [2] to improve the system efficiency with improved version of Genetic Algorithm (GA). Despite the power of GA in multi-objective optimization, its complexity and the relatively slow convergence limit its performance. Other heuristics techniques such as Particle Swarm Optimization (PSO) and Ant Colony

Optimization (ACO) are also considered for cognitive engine implementations. Simulation results in [3] have shown that the PSO can solve multi-objective optimization problems and dynamically select transmission parameters in CR applications. The work in [4] applied ACO to the cognitive engine design. However, long-term learning ability, which is essential function of CR to accumulate knowledge from its past running experiences is not investigated in these engines. Learning based algorithms such as CBR is exploited to create cognitive engines. The work in [5] exploits CBR in cognitive system adaptation in IEEE802.22 networks. CBR is fast in decision-making with good scalability. However, it is not efficient if the encountered scenario does not match with any of the cases stored in its database. In order to overcome the limitations of each of the above techniques, hybrid decision-making approach was introduced. This approach aims to combine learning and optimization algorithms to perform more efficient system adaptation. Authors in [6] exploits the advantage of quantum GA to design a hybrid engine with CBR. Another hybrid approach was investigated in [7] in which CBR and PSO are used as the core of the cognitive engine. Ashwin et al. [8] proposed a hybrid engine based on CBR and GA that has the capability to still adapt to new environments using GA. However, all the above hybrid engines relies on GA as an optimizer which is slow in convergence and is stuck in local optima. In addition, non of them considered efficiency, complexity and configurability range.

In this paper, we propose a hybrid cognitive engine that is generic and can fit with any cognitive resource management architecture from one side and addresses the trade-offs of different artificial intelligence algorithms from the other side. The engine is capable to overcome the complexity problem and extends the range of configuration to include not only the physical layer. The hybrid engine comprises CBR for learning and Decision-Trees (DTs) for parameters prediction. The design of the hybrid engine benefits from the rich experience of CBR to reduce the radio parameters adaptation time in domains with minimum knowledge [6]. DTs [9] are useful for classifying situations with large amounts of data as they are simple and require minimum processing time. The performance of this hybrid engine is demonstrated through the implementation using Software Defined Radios (SDRs) called USRP-N210 [10] in home network. In addition, simulation was conducted to validate the engine performance in multi-

carrier environment. The home network was selected to be the implementation environment as it is convenient and comprises multiple users with different applications. Therefore, different cases with different traffic can be experienced. Throughput, Signal to Interference and Noise Ratio (SINR), and Packet Error Rate (PER) are the performance metrics used to quantify the network performance of the hybrid engine.

The rest of the paper is organized as follows, Section II presents the flow of control, the mechanism of the hybrid decision-making including the structure of the used algorithms: CBR case structure and retrieval and DTs for decision-making. In addition, performance metrics and radio parameters definitions are described there. The use of DTs to overcome the CBR related complexity issues is described in Section III. Section IV and V present the evaluation methods for our hybrid engine including testbed in home network and simulation for multi-carrier environment respectively. The paper concludes in Section VI.

II. HYBRID COGNITIVE ENGINE

This section presents the structure of the hybrid cognitive engine, which involves the radio system parameters and the adaption process using CBR and DTs.

A. System Parameters

System parameters are classified into three groups: transmission parameters that are tuned by the cognitive engine to reach the optimal state of operation, environment parameters, which are the indicators that define the communication environment state, and performance measuring parameters. Minimizing PER and maximizing throughput and SINR are the objectives, which the adaptation process aims to fulfill. Therefore, the considered transmission parameters include transmission power (P), modulation index (m), coding rate (R_c), bandwidth (B), Frame size (L), contention window size (CW) and transmission range between each two hops (d). The appropriate choice of these parameters guarantees efficient network operation. Environment parameters are Interference power (I) and noise power (N) which indicates the quality of the channel, path loss (PL), and ratio of successfully received frames to all the transmitted frames (SR) which helps evaluate the current transmission quality.

Performance measuring parameters are the metrics exploited to trigger the hybrid engine and quantify its performance. We selected throughput, PER, and SINR to be the performance metrics as they are vital for any wireless network performance evaluation.

B. Radio Adaptation Process

Radio adaptation is the process in which the cognitive engine decides on the best configuration of radio transmission parameters that suits the current environment status. The engagement of the cognitive engine in parameters configuration is tied to the mentioned performance objectives (i.e. PER, throughput and SINR). The contribution of these objectives in

the encountered scenario is emphasized using weights according to the application requirements. For example, if throughput is the main performance objective with the highest priority in the encountered scenario, then, it is the indicator that triggers the cognitive engine and it is assigned the maximum weight.

The adaptation process consists of three stages as in Fig. 1: observation, decision-making, and learning. Typically, observations include monitoring environment conditions and performance metrics that trigger the need for system adaptation. The control proceeds to the decision-making stage when the performance metrics are not within certain predefined thresholds. The decision-making exploits one of the following techniques: CBR decision-making, which is used in case the current environment conditions match with a decision-making case stored in the database. Otherwise, DTs decision-making is employed if CBR fails to find a case that matches the encountered scenario. The last stage is learning to retain the selected configuration for future purposes. This is accomplished by checking the performance metrics for improvement after applying the configuration. If it achieves better performance, then the configuration is added as a new case to the CBR database. The focus of this paper is only on the decision-making and learning stages. Therefore, spectrum sensing and environment monitoring details are omitted. In the following sub-sections we illustrate the CBR, and DTs contribution in the decision-making process

1) *Case-Based Reasoning (CBR)*: CBR is exploited to obtain the proper radio transmission parameters by matching the encountered scenario with list of cases for previous scenarios stored in the CBR module database. Each case consists of three attributes: the first one identifies the environment which are represented by (PL, SR, I) in our designed engine. The second attribute corresponds to the configuration of radio parameters obtained from DTs that was stored for the corresponding scenario. The last attribute records the quality feedback (F_q) received after the configuration is applied. The quality feedback is the obtained performance metrics after the configuration is applied. This feedback determines whether to store this case in the database instead of other case with less quality or to retain the old one. At the initialization stage, CBR populates the case database to create the history and experience for the cognitive engine. The database is populated according to the implementation environment with different cases represented by different environment conditions.

Case retrieval is the process that engages to fetch the cases that can be used to configure radio parameters for the current scenario. The similarity value is the factor that manage the retrieval process. The similarity value is a measure of how close the encountered scenario to the cases stored in the system database. This measurement relies on the environment parameters comparison. Let us assume that the case is stored as a vector of number where E elements are used to describe the environment parameters and C_i denotes the i_{th} case. The similarity measurement between the two cases C_i and C_j can

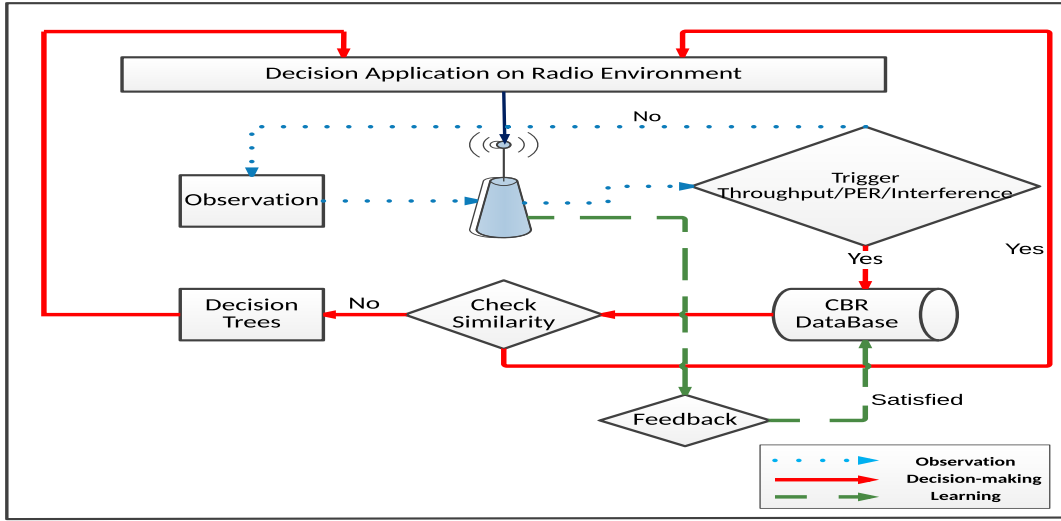


Fig. 1. Hybrid cognitive engine adaptation stages

be found as follows,

$$Sim_{ij} = \sum_{z=1}^E \frac{TS_z}{|C_i(z) - C_j(z)| + 1} \quad (1)$$

where TS_z is the similarity threshold for the z element and $C_i(z)$ is the z_{th} element of i_{th} case. As the considered elements for environment parameters are (PL, SR, I), if the absolute difference values of these elements between C_i and C_j are zeros, then the similarity measurement of the two cases is 4 which is the maximum achievable value. We define T_z as the threshold that determine if two cases are similar. If $Sim_{ij} > T_z$, then the two cases are similar enough to exploit the configuration that corresponds to that case. However, there can be more than one case similar enough to be exploited. Therefore, only the case with maximum (F_q) is selected. The CBR database is maintained by adding the configurations that have satisfactory performance and removing the ones that are not used very often. The process of maintaining the database is executed every time satisfactory configuration is applied to the system. .

2) *Decision-Trees (DTs)*: DTs are exploited as a machine learning technique [11] for classifications and predictions of transmission parameters. DTs consist of a set of hierarchical rules that divide the data into groups, where a decision is made for each group. We follow the Quinlan style of decision-tree algorithms, [12] which is based on utilizing observed data to create the decision-tree. The machine automated method for putting together a decision-tree based on observed data and the performance metrics indicators. The configuration of transmission parameters is linked to the estimation of performance metrics. PER estimation and SR are the indicators used for frame size adaptation. SINR measurement and SR are exploited for bandwidth adjustment. For power and modulation adaptation, if the target PER is not achieved, then the modulation order is progressively reduced. If the minimum modulation order still does not lead the PER to reach its desired target, then power is adjusted. The engine decides

how much to increase the power by estimating the interference power and SINR, and then it increases the power to reach a target SINR.

The construction of the tree starts by checking which one of these metrics has the highest gain i.e. the one with the highest priority. This depends on the performance and the radio environment. For example, if the objective is to maximize throughput, then, throughput is the metric that has the highest priority and it will be assigned as the root of the tree. After the root of the tree is specified, the branches are determined according to certain threshold and ranges. In each range, the action is taken to tune the transmission parameters or to consider another performance objective. The sample decision-tree for the multimedia scenario is shown in Fig. 2 provided that PL, I, and Noise are 10, -70, and -116 dBm respectively. We notice that the root of the tree is throughput per user with three different ranges: less than 300 kbps, between 300 and 600 kbps and above 600 kbps. Therefore, the typical range is between 300 and 600 kbps which extends the tree to consider the second priority metric. If the throughput is above 600 kbps, the decision to be made is to decrease parameters including (m_i, R_c, d, CW, L). If the throughput is less than 300 kbps, the action will be to increase these parameters. The second priority metric is PER, which involves three ranges including, above 10^{-2} , between 10^{-2} and 10^{-4} , and below 10^{-4} . If PER is less than 10^{-4} , the engine increases parameters including (m_i, R_c, d, CW, L) and reduces the bandwidth. On the other hand, when the PER is above 10^{-2} , all the parameters are decreased except the bandwidth. The last objective is SINR which leads to the last three decisions shown in Fig. 2. The decisions made are to increase, decrease or keeping the current configuration of bandwidth and power if the SINR typical range is not violated. The length of the tree is an essential parameters to consider. Thus, in our implementation, the length of the tree is determined according to number of the objectives involved in the decision-making process.

Algorithm 1 illustrates the adaptation process of the hybrid engine. The process engages only when it notices that the

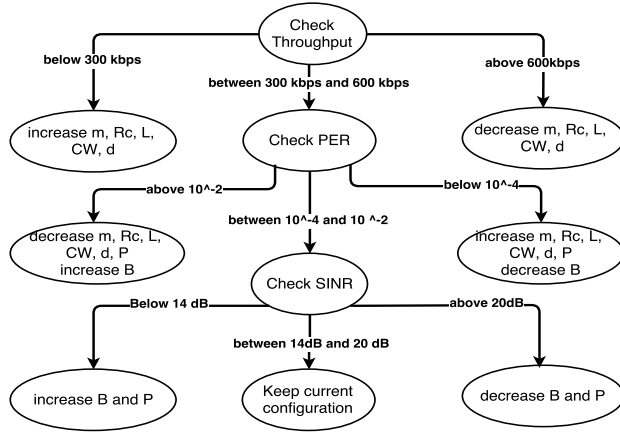


Fig. 2. Three objectives decision-tree

performance metrics are not in the typical ranges. If so the decision-making process advances to CBR to find a similar case with maximum quality feedback. If CBR cannot find a similar case, the DTs module engages for decision-making. The achieved performance is monitored to update the CBR with new cases that record adequate results. The $Case_{size}$ in

Algorithm 1 Hybrid Cognitive Engine Decision-making

Require: Environment Parameters (PL, SR, I), QoS metrics (Throughput, PER, SINR), Threshold (th) for QoS metrics

Ensure: Transmission Parameters (P, m, B, R_c, CW, d)

BEGIN

Case formulation: (Check environment parameters and monitor QoS metrics)

if ($PER \geq PER_{th}$ and $Throughput \leq Throughput_{th}$ and $SINR \leq SINR_{th}$) **then**

while ($Case_{size} \neq max$ and $out = 0$) **do**

Case Matching: check similarity value (Sim)

if ($Sim \geq Sim_{th}$) and (F_q) is the maximum **then**

 Extract the corresponding case solution

 Output (P, m, B, R_c, CW, d), $out = 1$

else

$Case_{size} \leftarrow Case_{size} + 1$

end if

end while

if ($out = 0$) **then**

 Execute DTs decision-making

 Output (P, m, B, R_c, CW, d)* $out = 1$

end if

end if

Monitor performance metrics (Throughput, PER, SINR)

if ($PER \leq PER_{th}$ and $Throughput \geq Throughput_{th}$ and $SINR \geq SINR_{th}$) **then**

 Update the CBR database,

end if

END

Algorithm 1 refers to the size of the tree branch that contains the cases that fall within the similarity predefined threshold.

III. HYBRID ENGINE COMPLEXITY CONSIDERATION

Studies conducted in [13] revealed that there are timing issues when CBR is implemented using hardware testbed. A non-negligible latency is introduced in case searching, and certain timing deadlines may not be met, thus causing problems such as frames collisions. Therefore, latency and complexity in case searching and retrieval is an important issue to consider. In the efforts to solve these issues, we can improve the performance of the searching algorithm used to find similar cases. In the regular searching process, the case similarity is checked for all the cases stored in the database which requires large computations. Our approach looks for unique aspects of each case to reduce the overall searching time and this allows the CBR database to grow large for more complex networks, and maintains fast access time. Predefined thresholds for similarity relative to each radio parameter are used to index different cases within the database. The appropriate selection of the threshold for each radio parameter simplifies the determination of similarity. We take advantage of DTs module to split the search space into parts (nodes) which contain a number of similar cases according to the similarity calculated in (1). Thus, every node in the constructed tree represents a subset of the cases of the CBR database and the root leads to the whole cases. The tree design does not require any computation on deciding where to split the search space, as it is predefined before any case enters the database. The average achieved complexity for this tree-based indexing is $O(\log(n))$ where n is the number of cases in the CBR database. As our cognitive engine is mainly dependent on decision-trees, the average complexity of our DTs algorithm is $O(\log(n))$. The convergence speed investigated in the evaluation section shows how fast our hybrid engine in decision-making.

We measure the access time achieved for different CBR base size and compare it to the traditional CBR access time. The measured access time presented in Fig. 3 shows that our cases retrieval approach has minimum access time regardless of the CBR database size. This makes our engine capable to handle large scale networks with larger database.

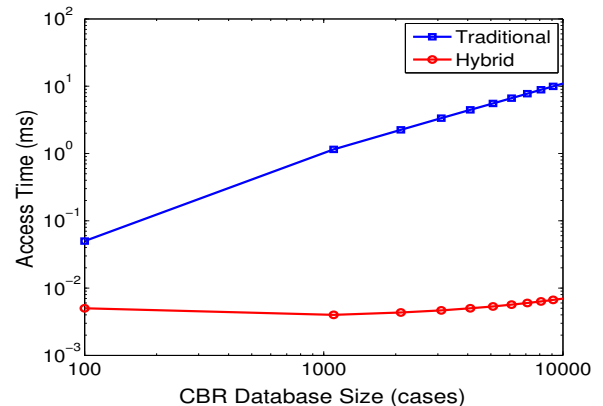


Fig. 3. Average CBR access time with different database size

IV. HYBRID COGNITIVE ENGINE TESTBED IMPLEMENTATION

In this section, we demonstrate the capability of our hybrid cognitive engine through real network testbed. The demonstration involves hardware implementation using USRP-N210 as SDR front-end in typical home environment. The USRP-N210 [10] is used in combination with the liquid-DSP software suite for software based signal processing [14].

A. Hardware and SDR Interfacing

USRP-N210 are flexible RF hardware front end for sending and receiving data as they have high-dynamic range processing capability. The control messages exchange between the USRP-N210 hardware and the host computer that includes the hybrid engine implementation and data processing for performance metrics collection, decision processing and configuration feedback collection from the network are handled using separate threads. This provides a reliable base for data collection, both online and offline, in a multi-node ad hoc network. For radio parameters configuration, daughter board is installed on top of the USRP-N210 to determine the possible configuration ranges for the physical layer parameters. We use RFX2400 as a daughter board in our implementation, which can support typical frequency ranges for home network. For Upper layers radio parameters, windows NDIS 6.0 is exploited as an interface to facilitate the communication between the hybrid engine and the Network Interface Controller (NIC) to access the upper layers, process data, and apply the configuration decided by the hybrid engine.

The choice of the ranges of transmission parameters is determined in a way that eliminates unnecessary data that complicates the searching process conducted by the decision-making algorithms. For example, the USRP-N210 allows one to set transmit power between -80 dB and 50 dB. However, the results of the experiments we conducted to set the power gain and measures SINR showed that the usable range of transmission power is between -50 and -10 dBm. The rest of the physical parameters were selected to match with the hardware capability. For performance metrics collection, we measure the data sent over a specified number of frames, which can accurately provide results to the profiling mechanism. For example, PER is presented on a 20 frame window. That is, for the last 20 received frames, the engine records the number of frames that failed in the Cyclic Redundancy Check (CRC). For throughput and SINR, the measured data is averaged at the receiver as a measure of the transmitter efficiency and it also reflects the status of the network in terms of interference.

B. Hardware Implementation Scenarios

We implemented two scenarios to demonstrate our hybrid engine radio adaptation capabilities. The first scenario focuses on the reliability while the second scenario is multimedia where throughput is the main objective. Both scenarios are detailed in the next sub-sections.

1) *Reliability Scenario*: This scenario incorporates a TCP file transfer between two nodes. Each node is connected to one USRP-N210 hardware for energy detection spectrum sensing and configuration. We evaluate the reliability of our hybrid engine under interference effect. The interference is formulated by using a third USRP-N210 that injects interference in a form of jamming signal (narrow-band interference) The interfering signals are generated by varying the bandwidth, the center frequency and the transmission power. In this scenario, we compare our hybrid engine performance with non-cognitive (non-CE) radio that is incapable of changing its initial configuration parameters. In addition, we compare it with the cognitive system (CRM) implemented using USRP2 hardware in [15] and the hybrid engine (CBR+GA) proposed in [8]. At the initial stage of testing, the radio parameters were configured to have ideal values which are the minimum value of transmission power, and the maximum value of contention window, transmission range, symbol rate, modulation, coding rate, bandwidth, and packet size. Then, the adaptation of radio parameters starts and follows the same process explained in Section II.B. Table I presents the parameters changes in reaction to the variation of PER and SINR to their threshold values. The measured values as function of time for PER and SINR are presented in Fig. 4 (a) and (b). As it can

Parameter	Initial Value	Optimized Value
Transmission power (P)	-22 dBm	- 15 dBm
Frame size (L)	1024 Bytes	750 Bytes
Modulation type (m)	64-QAM	8-PSK
Coding Rate (R_c)	3/4	2/3
Bandwidth per sub-carrier	812.5 kHz	625 KHz
Contention Window	64 for min and 4096 for max	32 for min and 2048 for max
Transmission Range (d)		d/4

TABLE I
CONFIGURATION PARAMETERS ADAPTATION

be seen that our hybrid engine outperforms the CRM and CBR+GA engines in terms of PER and SINR achieved. In addition, our engine is faster in making decisions as it relies on CBR learning with DTs decision-making capability to avoid complicated optimization unlike the CRM engine which has no learning capability and CBR+GA, which only relies on complicated GA optimization. The average PER achieved by our engine is 0.006, which is less than the threshold 0.01, less than 0.025 achieved by CRM, and less than 0.011 recorded by CBR+GA. Moreover, Fig. 4 (b) shows that our hybrid engine is the only one that managed to keep SINR above threshold, which is 14 dB.

2) *Multimedia Scenario*: In this scenario, we deploy two pair of nodes that have different types of traffic to send. The first one sends real time video traffic while the second one has best effort file sharing (FTP) traffic. The nodes were stationed in different rooms to avoid the high power leakage of WiFi cards that severely affects the expected performance. The traffic of the second pair of nodes was fixed to be on channel 11 while the first pair of nodes can select the best channel among the rest. Throughput measurement is the trigger

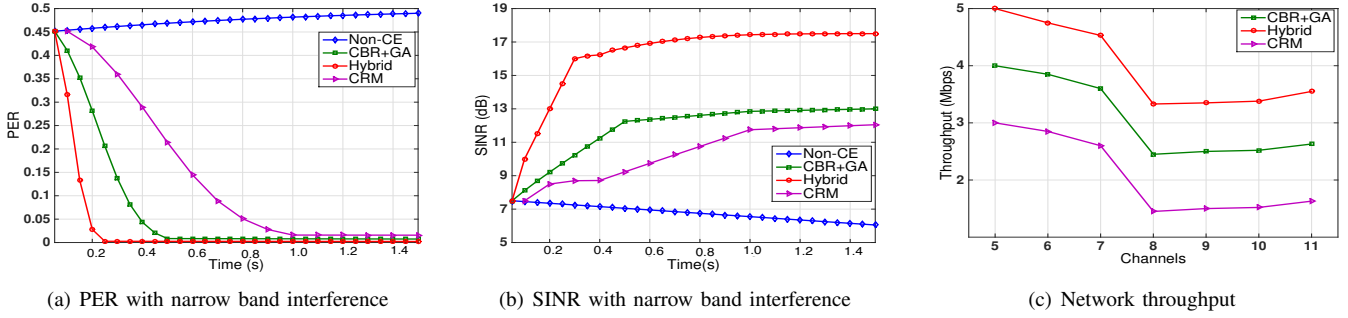


Fig. 4. Implementation results in reliability and multimedia scenarios

of the hybrid engine in this scenario. Fig. 4 (c) shows the cumulative throughput of both pair of nodes achieved by our hybrid engine compared to the CRM and CBR+GA engines. The measurements are plotted against the center frequency (channel number). The results in Fig. 4 (c) show that our hybrid engine achieved better throughput than the CRM and CBR+GA systems. One thing to note is that the throughput decreases as the transmission channel is closer to channel 11. Therefore, the best practice is to transmit on channel that is far from channel 11.

V. HYBRID COGNITIVE ENGINE EVALUATION

We simulated a multi-carrier system with 64 sub-carriers. Each sub-carrier was assigned a random attenuation value to simulate a dynamic channel. Hence, the SNR varied for each channel, inducing a need for the adaptation for each individual channel. The performance of the hybrid decision-making engine is compared with a GA based cognitive engine proposed in [16], neural networks based engine (ANN) proposed in [17], the hybrid engine (CBR+PSO) in [7], and with GA optimizer (CBR+GA) in [8].

The hybrid engine performance is demonstrated using two evaluation metrics which are PER and throughput. Table II presents the ranges of the transmission parameters, CBR related parameters, and performance metrics.

Parameter	Value
Transmission range(d)	$d, d/2, d/4$ and $d/8$
Contention window size (CW)	CWmin 8 values between 4 and 64. CWmax 8 values between 32 and 4096
Frame size (L)	24 bytes to 1024 bytes
Bandwidth (B)	312.5 to 812.5 KHz
Channel coding rate (R_c)	1/2, 1/3, 2/3, 3/4
Modulation type and order	M-PSK and M-QAM
Transmission power (P)	-50 dBm to -10 dBm
Symbol Rate	1 to 5000 Ksps
Case Base size	400
PER	5×10^{-3} (threshold) to 10^{-2} (tolerable)

TABLE II
SYSTEM PARAMETERS AND PERFORMANCE METRICS

The average PER is measured and plotted in Fig. 5 for all the engines. As anticipated, our hybrid engine achieved the minimum PER with the highest convergence speed. However, the CBR+PSO achieved less PER than the regular GA and ANN. CBR+GA converges to low PER, but the iteration consumes more time than CBR+PSO.

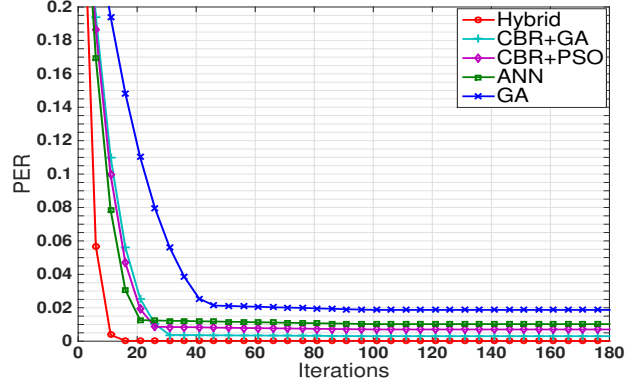


Fig. 5. Average PER for different cognitive engines

Fig. 6 presents the average user throughput achieved by the cognitive engines with variation of the number of nodes in the network in comparison with other cognitive engines. We observe that all the algorithms manage to achieve good

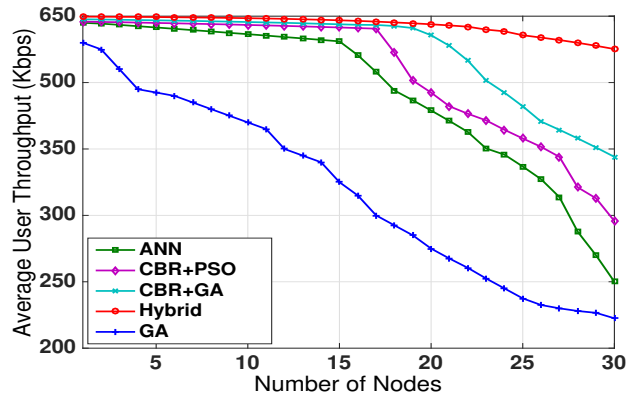


Fig. 6. Average user throughput achieved by different cognitive engines

performance with 15 or less nodes connected to the base station. However, for larger number of nodes, we can observe that GA has the worst performance as it is a stochastic search algorithm that require large computation. The ANN and CBR+PSO engines achieve good throughput within short time when the number of nodes is low. However, they may simply not be able to find a viable solution under some extreme radio scenarios (e.g., when the required resources approach the capacity limit of the system). CBR+GA achieves

comparable performance to our hybrid engine. However, the hybrid decision-making engine managed to achieve the highest throughput within the shortest time as it employs an improved case-based learning mechanism and powerful learning DTs.

Table III presents the results of average time consumed for each decision-making for all the engines. The minimum time recorded by our hybrid engine is (0.052 sec) for 400 case CBR reflects the complexity consideration of the CBR case retrieval. Our hybrid engine managed to handle the trade-off between quality and speed of decision-making.

Engine	Average Time
GA	1.162 sec
ANN	0.125 sec
CBR+PSO	0.22 sec
CBR+GA	0.361 sec
Hybrid	0.052 sec

TABLE III
TIME CONSUMPTION OF DIFFERENT DECISION-MAKING ENGINES

VI. CONCLUSION

This paper presents a hybrid cognitive engine for radio parameters adaptation using CBR for reasoning and DTs for learning and decision-making. We have described the engine structure, process flow, and radio adaptation process. The engine was implemented using USRP-N210 hardware and tested in two different scenarios: reliability and multimedia. Performance results show that our engine is able to deliver services with higher throughput, SINR and minimum PER with different interference effects compared to other proposed engines. In addition, simulation results demonstrate the high speed of convergence and the best throughput compared to other engines with different searching and optimization techniques including hybrid engines.

REFERENCES

- [1] IEEE Std. 802.11-2007, "Wireless lan medium access control (mac) and physical layer (phy) specifications," 2007.
- [2] I. AlQerm and B. Shihada, "Adaptive multi-objective optimization scheme for cognitive radio resource management", in *IEEE Global Communications Conference (GLOBECOM)*, Dec 2014, pp. 857–863.
- [3] Zhijin Zhao, Zhen Peng, Shilian Zheng, and Junna Shang, "Cognitive radio spectrum allocation using evolutionary algorithms", *Wireless Communications, IEEE Transactions on*, vol. 8, no. 9, pp. 4421–4425, September 2009.
- [4] Nan Zhao, Shuying Li, and Zhilu Wu, "Cognitive radio engine design based on ant colony optimization", *Wireless Personal Communications*, vol. 65, no. 1, pp. 15–24, 2012.
- [5] An He, Joseph Gaeddert, Kyung Kyoon Bae, Timothy R. Newman, Jeffrey H. Reed, Lizdabel Morales, and Chang-Hyun Park, "Development of a case-based reasoning cognitive engine for ieee 802.22 wran applications", *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 13, no. 2, pp. 37–48, Sept. 2009.
- [6] Zhi jin Zhao and Hai chao Lai, "A cognitive engine based on case-based reasoning quantum genetic algorithm", in *IEEE 14th International Conference on Communication Technology (ICCT)*, Nov 2012, pp. 224–228.
- [7] Xiaobo Tan, Hang Zhang, and Jian Hu, "A hybrid architecture of cognitive decision engine based on particle swarm optimization algorithms and case database", *annals of telecommunications - annales des telecommunications*, vol. 69, no. 11-12, pp. 593–605, 2014.
- [8] Ashwin E. Amanna, Daniel Ali, David Gonzalez Fitch, and Jeffrey H. Reed, "Hybrid experiential-heuristic cognitive radio engine architecture and implementation", *Journal Comp. Netw. and Commun.*, vol. 2012, pp. 549106:1–549106:15, 2012.

- [9] I. AlQerm, B. Shihada, and K.G. Shin, "Cogwnet: A resource management architecture for cognitive wireless networks", in *22nd International Conference on Computer Communications and Networks (ICCCN)*, July 2013, pp. 1–7.
- [10] "Usrpn210: Software defined radio", <https://www.ettus.com/product/details/UN210-KIT/>.
- [11] A. Navada, A.N. Ansari, S. Patil, and B.A. Sonkamble, "Overview of use of decision tree algorithms in machine learning", in *IEEE Control and System Graduate Research Colloquium (ICSGRC)*, June 2011, pp. 37–42.
- [12] R. R. Quinlan, "Programs for machine learning.", 1993.
- [13] An He, Joseph Gaeddert, Kyung Kyoon Bae, Timothy R. Newman, Jeffrey H. Reed, Lizdabel Morales, and Chang-Hyun Park, "Development of a case-based reasoning cognitive engine for ieee 802.22 wran applications", *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 13, no. 2, pp. 37–48, Sept. 2009.
- [14] J. D. Gaeddert, "Facilitating wireless communications through intelligent resource management on software-defined radios in dynamic spectrum environments," 2011.
- [15] E. Meshkova, Z. Wangt, J. Nasreddine, D. Denkovski, C. Zhao, K. Rerkrai, T. Farnham, A. Ahmad, A. Gefflaut, L. Gavrilovska, and P. Mahonen, "Using cognitive radio principles for wireless resource management in home networking", in *IEEE Consumer Communications and Networking Conference (CCNC)*, Jan 2011, pp. 669–673.
- [16] Thomas W. Rondeau (virginia Tech and Usa Trondeau@vt. Edu, "Cognitive radios with genetic algorithms: Intelligent control of software defined radios".
- [17] B. Bojovic, N. Baldo, and P. Dini, "A neural network based cognitive engine for ieee 802.11 wlan access point selection", in *IEEE Consumer Communications and Networking Conference (CCNC)*, Jan 2012, pp. 864–868.