On Quantifying the Experience Level of a Cognitive Engine

Conference Paper · March 2015

CITATIONS
2

READS
69

4 authors, including:

Hamed Asadi
The University of Arizona
13 PUBLICATIONS **38** CITATIONS
SEE PROFILE

Haris Volos
46 PUBLICATIONS **316** CITATIONS
SEE PROFILE

Michael Marefat
The University of Arizona
95 PUBLICATIONS **890** CITATIONS
SEE PROFILE

On Quantifying the Experience Level of a Cognitive Engine

Hamed Asadi, Haris Volos, Michael Marefat, and Tamal Bose
Dept. of Electrical and Computer Engr.
The University of Arizona
Tucson, AZ 85721-0104
{hasadi, hvolos, marefat, tbose}@email.arizona.edu

***Abstract*—A cognitive radio engine (CE) is where the advance adaptation algorithms for a cognitive radio (CR) is implemented. A CE is an intelligent agent which observes the radio environment and chooses the best communication settings that best meet the application's goal. In this process, providing reliable performance is one of the major tasks in designing CEs for wireless communication systems. In our previous work, we have proposed a metacognitive engine (meta-CE) that is able to automatically evaluate various CE algorithms performance. A meta-CE is generally considered to be comprised by a set of CE algorithms and a metacognition module that provides the meta abilities of the CE. We have identified that the most important task in order to enable metacognition is characterizing the performance of each individual CE algorithm on a given operating scenario. An operating scenario is defined as the set of the operating objective, channel availability, and channel quality metrics. In this paper, we develop performance evaluation metrics to quantify the amount of knowledge of different CE algorithms independently of the implementation approach and operating scenarios. By using these new metrics, we will be able to provide a more accurate estimation of the performance of each individual CE algorithm. Also, we can facilitate our CE algorithm by the new metrics due to their flexibility and ultimately predictability of the Cognitive Radio Systems' (CRS) capabilities. Finally, we use the proposed metrics to control the exploration rate of the CE algorithm. Our results show that the proposed contextual CE algorithm is able to improve our objective rewards by 10% vs. the ϵ greedy algorithm.**

I. Introduction

Conventional radios typically use a fixed set of communication methods that were preprogrammed by the radio's manufacturer. Nowadays, work in cognitive radio (CR) aims to change that by designing radios that are able to adapt their communication methods based on the operating conditions. The idea of CR comes from Mitola that described CR as a radio, which is not only able to optimize its own capabilities, but also can self-determine the goal by observing its operator and environment [1]. In our work, we focus on designing radios that optimize their capabilities.

CR is the key to pushing the limits of radio performance and maximizes utilization of the available resources such as power and radio frequency (RF) spectrum. To date, a plethora of research has been done [2]–[4] on the methods needed for developing CRs and understanding of their limits and capabilities. Besides the academia, the CR research has begun to influence the wireless communication standards through the relevant national and standardization bodies in communications.

By the fast-growing use of CR applications and CEs, providing predictable performance at all times is of paramount importance in different CE techniques. In this work, we focus on how to learn the experience level of individual CE algorithms in distinct operating scenarios. The new evaluation metrics presented in this paper aim to enable a CE and a meta-CE to predict the performance level of different CE algorithms. To this end, in the first step, we characterize the performance of different CE algorithms based on the operating channel scenarios and measure the confidence level of each CE algorithm with lower and upper bounds of their expected performance. Then, we predict the training rate and the expected performance of individual CE algorithms for ongoing operating channel scenarios based on the information from the previous step. Finally, we can create a meta-CE [5] [6] that can choose from a set of available algorithms for providing the needed performance level of CR by using the new proposed metrics. This approach allows a powerful estimation of the performance level of CEs and helps a radio to provide a deterministic output.

The first contribution of this paper is proposing three performance evaluation metrics. These metrics' aim is to quantify the amount of knowledge for each CE algorithm. Secondly, we show how a CE or meta-CE algorithm can be facilitated by using the proposed metrics as an index to approximate the expected performance level of each individual CE algorithm. The main purpose of this paper is to provide predictable performance based on the available CE's experience database. And thirdly, a meta-CE predicts the performance level of a CE algorithm for an operating scenario by using of learning curves and knowledge indices.

This paper is organized as follows: Section II, presents an overview of different CE algorithms and problem description. Section III, describes our recent advancement on defining new performance evaluation metrics. Finally, Section IV provides concluding remarks.

II. Background and Problem Formulation

The main objective of different CE algorithms is improving the radio performance in various operating scenarios. This improvement could be increasing spectral efficiency, throughput,

power consumption or efficiency, etc. Different CE algorithms will go across various paths to meet its objective. Therefore, one of the most critical phases of the CE's operation that makes a considerable difference between CE techniques is the training phase. This is the time that the CE is called to maximize an objective, but it doesn't have enough knowledge of the radio's abilities, and it has to experiment by sending training packets. It was found that near maximal performance can be reached relatively fast if the majority of the operating configuration sets meet minimum performance requirements. On the other hand, a significant amount of trials is required if the number of operating configuration sets that meet minimum performance is extremely small. In this journey, CEs will learn from their mistakes and explorations that how distinct configurations behave, in particular, channel scenarios. Since CE algorithms will pass distinct roads to fulfill their objectives, we need to define some metrics to evaluate their progress during training phase [7].

A. Exploration Vs. Exploitation

A primary function of the CE is to learn the capabilities of the radio. This is generally done by trial and error. When the radio performs this function, it is said to be exploring. On the other hand, when the radio is choosing methods with the best-known performance, it is said to be exploiting. The exploration operation consumes valuable resources such as time and energy and might significantly impact the link performance (i.e., dropped packets). One option to avoid these negative effects during the radio's operation is to put the CE through prolonged training sessions covering most expected operating conditions. However, even if the CE is assumed to go through prolonged learning (exploration) sessions, it is practically impossible to expose it to all possible channel conditions a priori. Consequently, it is reasonable to expect that the CE sooner or later will face unknown conditions. In such a case, if the radio is operating in a critical mission it may not have the luxury of time to learn what is best before operating; it has to establish a connection and learn at the same time. Optimally balancing exploration vs. exploitation ensures that the negative effects of learning will be kept to a minimum. Therefore, we need to evaluate the performance of the different CE algorithms, to determine the optimal CE for various operating scenarios.

The goal of this paper is to investigate and apply some performance evaluation metrics to estimate the amount of knowledge that is obtained by the CE algorithm during an exploration vs. exploitation balancing technique, to utilize these techniques based on their capabilities in appropriate conditions.

B. Training overview

Training is used in many AI based systems. For example, a learning-tree-based classifier is typically trained using a specific training set, with a goal of minimizing the classification error. In a back-propagation ANN, the least mean squares (LMS) algorithm is widely used to minimize the training error. In κ-means, clustering is used to identify κ groups in a set of data that minimize the sum of the squares of the distances of each data point to its assigned group. The examples cited are batch techniques, i.e., the whole dataset is available before training commences. On the other hand, online versions are present that can process data as the data arrive.

Three primary types of learning, reinforcement, supervised, and unsupervised learning, are used in the context of a CE. Because various types of learning can be applied in the context of a CE, we refer to the different learning types more generically as training to keep the discussion general. First, in a CE, the training task is assumed to be online and a joint learning and optimization process takes place. This operation can be cast as a reinforcement learning task, which attempts to learn so that a reward is maximized. In reinforcement learning, the behavior is adjusted as rewards are received. Second, supervised learning is based on examples of the desired behavior or attributes being learned. Supervised learning exists in the context of a CE when the capabilities of the system are learned by observing action-outcome pairs. Both reinforcement and supervised learning exist in the context of a CE: reinforcement learning is used to decide, based on previous experience, upon the next communications method to be used; supervised learning is used when the action-outcome pairs are used to estimate system abilities. One example is the training of a Bayesian classifier, in which "action" is the communication method that was used to establish the communications link and the "result" is the number of successful and failed packets using this method. Finally, in unsupervised learning, no explicit groupings are specified in the collected data. The unsupervised learner extracts features from the data, such as clusters of similar items. Unsupervised learning can be used for data organization and memory compression; this type of learning is omitted from our study.

C. Problem Description and Formulation

The general problem in a link adaptation CE is that there is a list with a large number of possible communication methods that can be used. Each potential method is a discrete combination of modulation, coding, antenna techniques, and other possible parameters defining the communication method to be used. In this list, some of the methods are eligible and the rests are ineligible. An eligible method is a method that meets minimum performance requirements of the operation objective, and an ineligible method fails to meet those requirements given the current environment. The minimum performance requirements are typically a given PSR or bit rate (bits/s/Hz). Also it can be some minimum objective level such as throughput, power consumption, etc., For example, if a method has a 90% PSR in the current environment but the minimum required PSR is 95%, then this method will be ineligible. The goal in a link adaptation CE is to find the eligible method with the highest performance metric. In most cases, such as when the goal is to maximize bandwidth efficiency, the maximum potential performance of each configuration is already known. Therefore, the list of configurations can be potentially sorted by how well each item serves the current goal as apriori knowledge. In this case, the problem becomes a search through a sorted list.

Rather than finding the perfect technique for minimizing the learning cost (which we have done in previous publications [8], [9], [5] by adopting an optimal exploration vs. exploitation balancing strategy), the objective of this paper is to estimate the knowledge of each CE algorithm during operation. In our derivation, the estimation of the amount of knowledge that is obtained by a learning algorithm in its memory is only indirectly dependent on the performance metric. For this task to be analytically tractable, some assumptions must be made. We assume that our radio has K communication methods. For each method k, we have a potential reward R_k. Each method is assumed to be evaluated until its eligibility or ineligibility is verified. Also, for each method k, we have a belief state $\pi_k(n)$ which represents our knowledge about the underlying reward distribution at a time step n. $\pi(n)$ is a vector of all K belief states at time step n: $\pi(n) = [\pi_1(n), \pi_2(n), ..., \pi_k(n)]^T$. The belief state is $(\overline{\mu}_k(n), \overline{\sigma}_k^2(n), n')$ the estimates of the mean μ_k and the standard deviation σ_k, using n' samples, of the underlying reward process.

The reward in our problem depends on the objective. It can be throughput when our objective is maximizing throughput, or it can be power consumption, or spectral efficiency. If we use method k, at a time step n, we receive a reward $R_k(n)$. Therefore, the belief about the reward distribution changes from $\pi(n)$ to $\pi(n+1)$. The goal of CE is to select a method that will maximize the expected reward.

In the rest of this section, we present three different CE algorithms. The algorithms are ϵ-greedy strategy [10], softmax strategy based on the Boltzmann exploration [10], and the Gittins index strategy [11]. All the techniques have two things in common. First, all of them are based on stochastic principles. Second, they all have a factor that affects the exploration rate.

D. The ϵ-Greedy Strategy

The greedy algorithm [10] is the simplest and famous algorithm in this topic. The idea of greedy is choosing the best method (with highest reward) with the probability of $1-\epsilon$. And choosing a random method with probability of ϵ. The ϵ amount is called exploration rate that means the CE will do exploration with the fixed rate through all the times in this algorithm. There is another version of ϵ-greedy algorithm called annealing ϵ-greedy. In this strategy the amount of ϵ will decrease in each iteration to reduce the probability of randomness as we collect more information. The idea of annealing algorithms comes from this that at the beginning, since we don't have any information about communication methods, we need to do more exploration, however, by collecting more information, we need to decrease this rate. The exploration parameter ϵ is updated at every time step n by:

$$\epsilon = \frac{\epsilon_0}{1 + nd_\epsilon} \tag{1}$$

where ϵ_0 is the initial value of ϵ, and d_ϵ is the decrease rate.

E. Softmax strategy (Boltzmann Exploration)

Softmax algorithm is a kind of probabilistic strategy that works based on the idea of as we do more experiments the probability of choosing each method should be match with their rewards. In the reinforcement learning area, a softmax function can be used to do the conversion of rewards into action probabilities [10]. The function commonly used is:

$$P_k = \frac{e^{\overline{\mu}_k(n)/T}}{\sum_i e^{\overline{\mu}_i(n)/T}} \tag{2}$$

where T is a positive parameter called the temperature. When the value of temperature is high $(T \to \infty)$, the algorithm chooses all possible actions with the almost same probability. However, in the low-temperature value, the algorithm is highly probable to choose the method with the highest expected reward. Therefore, when $(T \to 0)$, the probability of the method with highest rewards to be chosen tends to 1.

The temperature T is updated at each time step n by:

$$T = \frac{T_0}{1 + nd_T} \tag{3}$$

where T_0 is the initial value of temperature, and d_T is the decrease rate.

F. The Gittins Index Strategy

Gittins and Jones [12] proved that our K-dimensional problem can be reduced to K one-dimensional problems by using a dynamic allocation index based strategy. Specifically, the optimal strategy that maximizes our rewards through time is simply selecting the method k that has the highest index ν_k at each time [11]:

$$\nu_k(\pi_0) = \sup_{N>0} \frac{E\{\sum_{n=0}^{N-1} \gamma^n R_k(n) | \pi_k(1) = \pi_0\}}{\sum_{n=0}^{N-1} \gamma^n | \pi_k(1) = \pi_0\}} \tag{4}$$

which is the expected total discounted reward normalized by the expected total discount time up to the stopping time N. The stopping time N may be different for each method k and depends on the initial belief state π_0. For example, after a few trials using a promising method k, the newly acquired knowledge may suggest that it is not worthwhile to continue using method k and the process needs to stop. This ratio is taken over all possible stopping times N and $\nu_k(\pi_0)$ is the maximum value. The optimal strategy is simply to use the option k with the highest ν_k. This method has been proven to be optimal by Gittins [12] [11] and others [13]. We discuss the use of gittins indices in more detail in two of our prior publications [8], [9].

The Gittins index is dependent upon the underlying distribution of R_k. In this work, we consider the Gittins index for the Normal Reward Process (NRP). In our application, if a transmitted packet is successfully received, then we assume a return equal to the rate of the communication option used, otherwise the return is zero. For an NRP, the Gittins index is equal to:

$$\nu(\overline{\mu}, \overline{\sigma}^2, n', \gamma) \equiv \overline{\mu} + \overline{\sigma}\nu(0, 1, n', \gamma) \tag{5}$$

where $\nu(0, 1, n', \gamma)$ is the Gittins index for a zero mean, unit variance distributed process and $\gamma \in (0, 1)$ is a discount

factor. Although easier than solving the original problem, calculating the Gittins indices is still not a trivial task. For the interested reader [14] provides a concise description of the method for calculating these indices. However, for most practical purposes, the indices tabulated in [11] are sufficient, and thus we use a table-lookup approach.

III. Cognitive Radio Engine Evaluation

One of the most crucial tasks in creating new CE algorithm is evaluating the CE performance in different operating scenarios. The effect of changing environment, objectives, learning techniques, and radio capabilities implies that making an evaluation method for CE algorithms is nontrivial. The first challenge is to find a way for monitoring the performance of individual CE techniques. In fact, every CR device is designed with various functionalities and expectations. Based on the definition of a CE, different CEs should be able to sense their own environment and autonomously adapt to the changing operational conditions. However, various CE designs provide distinct levels of situation awareness and cognitive functionality. Such diverse expectations make performance evaluation a great challenge [15]. To this end, drawing learning curves of CE techniques could be helpful. A learning curve is a graphic representation of the relationship between learning and outcomes. A learning curve can be characterized by the mean, variance and other statistical metrics of the CE performance. The learning curves can be very useful for comparing the behavior and expected performance of the various CE algorithms. The main idea of a learning curve is that it shows "The more you experiment, the better your outcomes will be, through learning" [16].

All of the proposed learning curves in our previous work [5], which are used for the evaluation of individual CE performances are based on the objective function of CE. For instance, if CE's objective is maximizing throughput, the learning curve will be created based on the estimated mean and variance of received throughput for each decision. The problem is that if we have another CE algorithm which operates with another objective such as power efficiency, we cannot compare the created learning curve for power efficiency with the learning curve of maximizing throughput. Hence, we need to define some metrics at a level of abstraction that is not depend on objective function and implementation details of CE algorithms. If we can have such unitless metrics, we will be able to compare the learning capabilities of two different CE algorithms with two distinct objective functions and implementation details.

A. Quantifying the Amount of Knowledge

In this paper, we present three different metrics to estimate the experience level of CEs. First, as we defined in previous section, we classify all available communication methods into two distinct classes, eligible and ineligible. The goal of a link adaptation CE is to find the eligible method with the highest performance metric. We use the belief state of each method k to estimate the upper bound and lower bound of the obtained rewards. To calculate the confidence intervals of a random process for unknown mean and standard deviation, we use t distribution as follows [17]:

$$R_{lk}(n) = \overline{\mu}_k(n) - (t(\frac{1-C}{2}, n'-1) * \frac{\overline{\sigma}}{\sqrt{n'}}) \quad (6)$$

$$R_{uk}(n) = \overline{\mu}_k(n) + (t(\frac{1-C}{2}, n'-1) * \frac{\overline{\sigma}}{\sqrt{n'}}) \quad (7)$$

where $\overline{\mu}_k(n)$ is the sample mean of obtained reward at time step n, $\overline{\sigma}$ is the sample standard deviation of the rewards, C is the confidence, n' is the number of trials that we tried method k, t is the upper $(1-C)/2$ critical value for the t distribution with $n'-1$ degrees of freedom, and R_{lk} and R_{uk} are the lower and upper bounds for the underlying reward distribution's mean of method k at time step n respectively.

To classify the communication methods we use following statement:

$$R_{uk}(n) < \underset{j\in[1,K]}{\operatorname{argmax}} \overline{\mu}_j(n) \quad (8)$$

means each communication method that has an upper bound smaller than the best obtained rewards mean is not eligible for next decision step. CE considers the statement 8 for all communication methods at each step to determine their eligibility. After labeling all methods, the CE will just consider the eligible methods as its option for next decision. By the definition of the classes, we can say at each time step the number of eligible and ineligible methods are:

$$N_K = N_E + N_I \quad (9)$$

where N_K is the number of all methods and N_E, N_I are the number of eligible and ineligible methods respectively.

The statement 9 is correct in all time steps. Therefore, one of the potential metrics to evaluate the learning algorithm could be the number of eligible methods. Generally, the fraction of eligible methods to the number of all available methods could tell us that how much the CE is close to find the best method. This metric is completely independent of the objective function and implementation details of CE algorithm, and we can represent it as a first index:

$$I_1(n) = \frac{N_K - N_E(n)}{N_K - 1} \quad (10)$$

where n represents the time step. If the $(I_1 \rightarrow 0)$ means all the methods are eligible, therefore, the CE does not have any confidence for choosing the best method, however, if the $(I_1 \rightarrow 1)$ means we have just one eligible method, so the CE is highly confident about the best method. Moreover, by evaluating the amount of this index, we can say how much of the journey to find the best method has already paced.

Since I_1 shows the real picture from the learning journey, we call it left-brain index (LBI). We chose this name because in humans, the left brain considers the analytically and logical side that provides a more realistic image of a phenomenon.

Although this index can give us some useful information about the progress of CE algorithm, it doesn't have any knowledge about the received rewards from methods that CE has tested until now. For instance, we cannot distinguish between a CE which considers the low potential communication methods with the CE which is working with high-potential

ones. Furthermore, if a CE has many methods with almost same rewards, the LBI index shows less progress to be close to the goal. Hence, we need another index, which will be able to give us information about the received rewards. To end, we define:

$$I_2(n) = \frac{\text{argmax}_{j\in[1,K]}\,\overline{\mu}_j(n)}{\text{argmax}_{j\in[1,K]}\,R_{uj}(n)} \tag{11}$$

The I_2 reflects the amount of CE's progress to find the highest reward. When $(I_2 \to 0)$ means CE does not have much information, and it is far from the maximum potential rewards. Conversely, when $(I_2 \to 1)$ means the CE is completely sure about the best method, and its obtained reward is close to its imagination about the maximum potential reward. In contrast with I_1, the I_2 is highly related to the reward's value, not the number of processed methods. For instance, if all the available methods are eligible, and they have very close rewards' value, this index will show us a high value close to 1, however, the first index will show a low value that reflects less knowledge. Moreover, The I_2 gives an information about how far CE is from its imaginary possibilities by maximum potential rewards. Since I_2 considers CE imagination and the human right brain also operates our imagination, daydreaming, and emotions, we call this index the right-brain index (RBI).

Finally, we present another index, which is the combination of two proposed indices. The new index is:

$$I_3(n) = 1 - \frac{\sum_{i=1}^{N_E}(R_{ui}(n) - \text{argmax}_{j\in[1,K]}\,\overline{\mu}_j(n))}{\sum_{j=1}^{N_K}(R_{ui}(0) - R_{li}(0))} \tag{12}$$

where the nominator is the summation of distance between upper bounds of eligible methods with the mean of best-known method at time step n. The denominator is the summation of upper bounds to lower bounds at the start point for all methods. The I_3 illustrates the amount of knowledge by considering the eligible methods and their obtained rewards. Since this index can reflect both aspects of training phase and makes a connection between the aforementioned indices, we call it the corpus callosum index (CCI). The corpus callosum is the brain structure links the two sides of the human brain together.

B. Experimental Results

To demonstrate the benefits of new proposed indices in evaluation of CE performances, we present a simple example. In this instance, we assume a 4 by 4 MIMO system with QPSK, 8PSK, 16, 32, 64, 128 and 256 QAM as a modulation type with eight error correction rates: 1, $\frac{7}{8}$, $\frac{3}{4}$, $\frac{2}{3}$, $\frac{1}{2}$, $\frac{1}{4}$, $\frac{1}{6}$ and $\frac{1}{8}$ and antenna techniques: VBLAST, STBC and MRC. For our channel scenarios, we consider an SNR in the range of 0-50 dB and the log_{10} of the eigen spread (κ) of the channel matrix in the range of 0-12. The CR also has 12 channels available with different SNR and bandwidth (either 1.25 or 2.5 MHz).

Figure 1 illustrates the results of four different CE algorithms. The plot is created by the averaging of 1000 distinct trials over 1000 time steps on the same channel scenario for all CEs. The first CE is the ϵ-greedy strategy with the constant

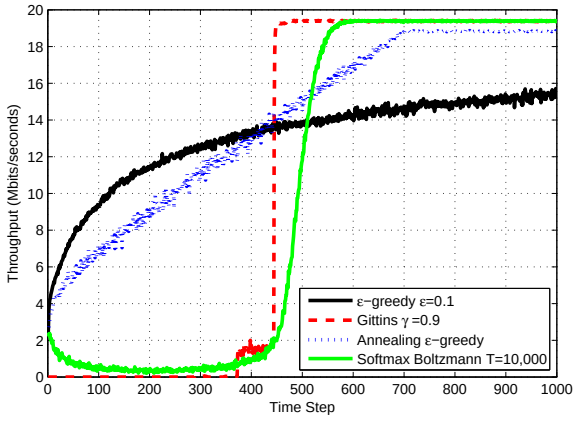

Fig. 1. Four different Cognitive Engine algorithm's performance

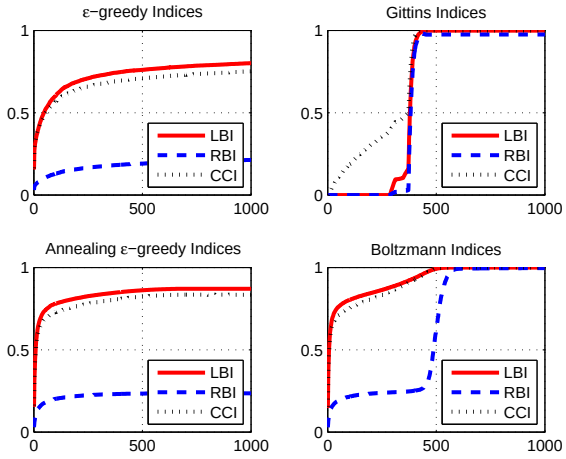

Fig. 2. Cognitive Engine algorithm's learning Indices

ϵ equals to 0.1. Since this strategy will choose the best-known method by the probability of 0.9, it has a conservative approach. Therefore, The CE stays with a method as soon as receiving a minimum acceptable reward. The second CE is based on the Gittins index strategy with the discount factor of 0.9. In contrast with the previous CE, this CE follows its own indices and in almost 370 time steps does not have any rewards. Nevertheless, after 450 time steps, it converges to the highest potential rewards very fast. The behavior of Gittins CE is much more aggressive than the greedy CE. Third CE is annealing ϵ-greedy with the initial ϵ equals to 0.7, and at each time step decreases by 0.001. Since this CE has the higher exploration rate at the start, in comparison with the first CE with constant ϵ, it will find the better communication method after a while. Therefore, the annealing ϵ-greedy strategy is more aggressive at the start; however, as it goes, it becomes more conservative. Fourth CE is based on the Boltzmann exploration strategy which is a probabilistic technique. At the start, CE has been decreasing rewards, since the CE initialized by the maximum potential rewards. Hence, the probability of

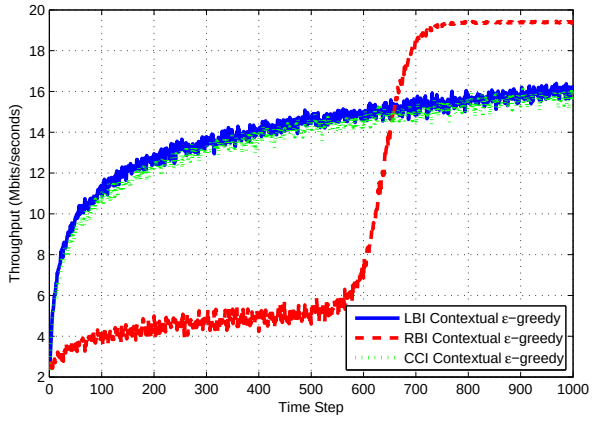

Fig. 3. Three Contextual exploration Cognitive Engine algorithms

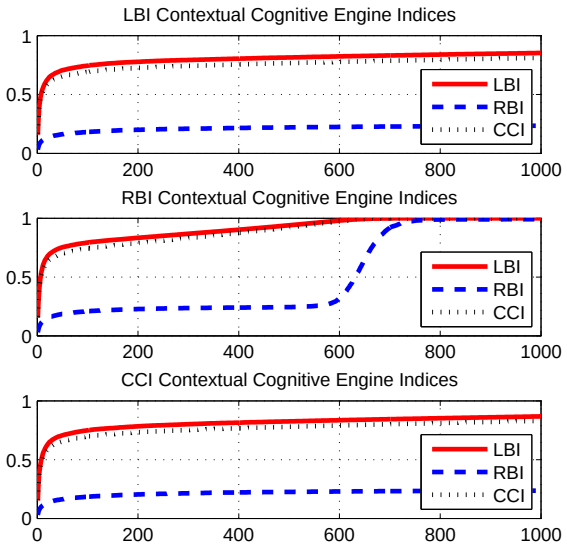

Fig. 4. The Indices of Contextual Cognitive Engine algorithms

finding the best method decrease until translating the most of apriori knowledge to real obtained reward in the current channel scenario.

In Figure 2, we can see the corresponded learning indices to each of the pointed out CE algorithms. All of the indices are unitless and completely independent of the implementation details and operating scenarios. The plots depict that the RBI behaves more aggressively for Gittins and Boltzmann strategies because they have an aggressive approach. This happens because they will be close to the imagination of maximum potential rewards after doing more exploration. Conversely, since more conservative algorithms such as ϵ-greedy with constant ϵ do not explore many times the RBI line is monotone, however, the LBI line shows lots of progress at the start point. The interesting point of CCI is the ability of prediction which it shows for various CEs. For instance, although the Gittins and Boltzmann do not have a good output at the first 400 steps, the CCI shows a very good learning progress. This predicts a jump after a while as we can see that happens for both cases.

There are several applications to utilize the new proposed indices. For instance, we can use them for prediction of the CE's performance. Furthermore, they can be used for comparison and analysis the performance of CE algorithms. Moreover, we can control the exploration rate of the CE algorithms based on the obtained information from the indices. To end, Figure 3 depicts the learning curves of three different contextual CE algorithms based on the ϵ-greedy strategy. Each CE uses one of the proposed indices to control the amount of ϵ as an exploration rate. Since the RBI insists on the maximum potential output based on the maximum upper bound, RBI-based CE explores more to reach its goal. However, the LBI-based CE keeps its obtained reward by doing some exploration based on the number of eligible methods that need to be investigated.

Figure 4 shows the indices of three new contextual CE algorithms. For each aforementioned contextual CEs, we plot the same indices. For instance, the LBI index of LBI-based CE, demonstrates more value than the same index for ϵ-greedy algorithms. Therefore, the LBI-based CE provides better outcome during same time steps. The other parts of Figure 4 illustrate the tree defined indices for the new contextual CEs that are created.

C. Future Work

Quantifying the knowledge/experience level of a CE is an important task because knowing the experience level will make a CE more dependable and predictable. These qualities will make a CE more desirable since system designers will be more comfortable to include them as a part of their system since they will know what to expect at a given time. Furthermore, it will enable the development of meta-CEs that combine the strengths of multiple CEs. That said, we plan to continue refining the metrics we presented in this paper and developing new metrics that potentially better quantify the experience level of the CE and/or are more useful in a practical application. Finally, we plan on further investigating the development of exploration methods that use the experience metrics of the CE.

IV. Conclusions

In this work, we proposed three evaluation metrics that quantify the experience level of cognitive engine (CE) algorithms. A metric that is based on the number of unexplored configuration options (LBI), a metric that is based on the possible rewards of the remaining unexplored configuration options (RBI), and a metric, which combines elements of the last two (CCI).

It was demonstrated by applying the metrics on four CE algorithms that most of the time the LBI and RBI metrics present a different perspective from each other and that the CCI metric provides a more balanced perspective compared to the other two. Especially the CCI metric can capture the progress made

when using the Gittins Index CE algorithm during the initial exploration phase which appears as if no progress was made if one was looking at the achieved performance and the LBI and RBI metrics.

Finally, we found that the metrics can be used as the basis of contextual exploration algorithms with promising results. We plan to continue development of both evaluation metrics and exploration methods that are based on the evaluation metrics.

REFERENCES

[1] J. Mitola, III, "Cognitive Radio: An Integrated Agent Architecture for Software Defined Radio," Ph.D. dissertation, The Royal Institute of Technology (KTH), Stockholm , Sweden, May 2000.

[2] S. Haykin, "Cognitive Radio: Brain-Empowered Wireless Communications," *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 2, pp. 201–220, 2005.

[3] A. MacKenzie, J. Reed, P. Athanas, C. Bostian, R. Buehrer, L. DaSilva, S. Ellingson, Y. Hou, M. Hsiao, J.-M. Park, C. Patterson, S. Raman, and C. da Silva, "Cognitive Radio and Networking Research at Virginia Tech," *Proceedings of the IEEE*, vol. 97, no. 4, pp. 660–688, 2009.

[4] M. Bkassiny, Y. Li, and S. K. Jayaweera, "A Survey on Machine-Learning Techniques in Cognitive Radios," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 3, pp. 1136–1159, Jan. 2013.

[5] H. Asadi, H. I. Volos, M. Marefat, and T. Bose, "Learning Characterization Framework and Analysis for a Meta-Cognitive Radio Engine," in *Proceedings of SDRWInnComm 2014 Wireless Innovation Conference on Wireless Communications Technologies and Software Defined Radio*, Mar. 2014, pp. 132–139.

[6] H. Asadi, H. Volos, M. Marefat, and T. Bose, "Metacognitive radio engine design and standardization," *Selected Areas in Communications, IEEE Journal on*, 2015.

[7] H. I. Volos and R. M. Buehrer, "Cognitive Radio Engine Training," *Wireless Communications, IEEE Transactions on*, vol. 11, no. 11, pp. 3878–3889, 2012.

[8] ——, "On Balancing Exploration vs. Exploitation in a Cognitive Engine for Multi-Antenna Systems," in *Proceedings of the IEEE Global Telecommunications Conference*, Nov. 2009, pp. 1–6.

[9] ——, "Cognitive Engine Design for Link Adaptation: An Application to Multi-Antenna Systems," *IEEE Transactions on Wireless Communications*, vol. 9, no. 9, pp. 2902–2913, Sept. 2010.

[10] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. The MIT Press, March 1998.

[11] J. C. Gittins, *Multi-Armed Bandit Allocation Indices*. Wiley, Chichester, NY, 1989.

[12] J. Gittins and D. Jones, "A Dynamic Allocation Index for the Sequential Design of Experiments," *Progress in Statistics*, pp. 241–266, 1974.

[13] J. N. Tsitsiklis, "A short proof of the gittins index theorem," *The Annals of Applied Probability*, vol. 4, no. 1, pp. 194–199, 1994.

[14] D. Acuña and P. Schrater, "Bayesian Modeling of Human Sequential Decision-Making on the Multi-Armed Bandit Problem," in *Proceedings of the 30th Annual Conference of the Cognitive Science Society*, 2008.

[15] Y. Zhao, S. Mao, J. O. Neel, and J. Reed, "Performance evaluation of cognitive radios: Metrics, utility functions, and methodology," *Proceedings of the IEEE*, vol. 97, no. 4, pp. 642–659, April 2009.

[16] J. Deane and S. A. Yourstone, "Learning and Thinking Systems," in *Learning Curves*, M. Y. Jaber, Ed. Boca Raton, FL: Taylor and Francis Group, LLC, 2011.

[17] H. Pishro-Nik, *Introduction to Probability, Statistics, and Random Processes*. USA: Kappa Research, LLC, 2014.