Metacognition and the Next Generation of Cognitive Radio Engines

Hamed Asadi, Haris Volos, Michael M. Marefat, and Tamal Bose

Much of the previous research on cognitive radio has focused on developing algorithms based on artificial neural networks, the genetic algorithm, and reinforcement learning, each with its pros and cons. In this research, we present a new approach based on metacognition. We believe that the meta-cognitive framework can be the foundation for the next generation of CRs and further the performance improvements in CR.

Abstract

Much of the previous research on cognitive radio has focused on developing algorithms based on artificial neural networks, the genetic algorithm, and reinforcement learning, each with its pros and cons. In this research, we present a new approach based on metacognition. We believe that the metacognitive framework can be the foundation for the next generation of CRs and further the performance improvements in CR. In this work, we present the elements involved in metacognitive radio, discuss the challenges in their development, present solutions to the challenges along with a possible meta-CR architecture, and show results from our implementation. Each cognitive engine (CE) algorithm has strengths and limitations that make it more suitable for certain operating scenarios (channel conditions, operating objective, available hardware, etc.) than other algorithms. A meta-CE can adapt faster and improve performance by exploiting the characteristics and expected performance of the individual CE algorithms. It understands the operational scenarios and utilizes the most appropriate algorithm for the current operational scenario by switching between the algorithms or adjusting them as necessary.

Introduction

Conventional radios typically use a fixed set of settings (modulation, channel coding, etc.) selected by their operator. A cognitive radio (CR) automatically adapts its configuration to meet its goal under the operating conditions. CR was first described by Mitola; his ideal CR can optimize its own capabilities and self-determine the operation goal by observing its operator and the environment [1].

To make CR possible, we developed an intelligent agent (IA) called the cognitive engine (CE) that enables the radio to have the desired learning and adaptation capabilities [2]. An IA is a system that senses its environment (operating channel scenario), acts by using a communication method based on its experience, and monitors its own performance to learn its capabilities.

Cognitive Engine Challenges

Since Mitola's seminal work [1], CEs have been researched extensively, and several algorithms and approaches were developed. That said, there are remaining challenges that our work seeks to address. First, CEs are typically designed around one or two primary algorithms or frameworks. Examples include the genetic algorithm (GA), the case-based reasoning (CBR) framework, and the epsilon-greedy (ε-greedy) exploration technique. The chosen algorithms and their parameters make them more suitable for certain operating scenarios than others. An operating scenario is a combination of the channel conditions and the operating objective (e.g., maximum throughput, maximum energy efficiency). For different operating scenarios, either the existing algorithms' parameters need to be adjusted or new algorithms need to be used. For example, in a low signal-to-noise ratio (SNR) environment a conservative algorithm may be preferred that focuses on energy-efficient methods with low error rates vs. in a high SNR environment where an aggressive algorithm may be preferred that prioritizes high spectrum-efficiency methods. Consider a complex system, such as massive multiple-input multiple-output (MIMO), with numerous modes; the problem of identifying the best[1] adaptation technique for the channel conditions becomes more challenging. The first contribution of the article is the introduction of a framework that facilitates the selection of the right learning algorithms and their parameters based on the operating conditions.

Moreover, the "predictability" of a CE's performance is of paramount importance to the adoption of a CE radio. A radio that provides estimates of its expected performance is easier to certify than a radio that cannot provide this information. "Predictability" means that given the channel environment, the CE can predict its performance range based on its experience. If the CE has significant experience in the channel environment, the predicted performance range will be narrow; otherwise (limited experience), the predicted performance range will be wide. Performance predictability and the suitability of each CE algorithm given the operating scenario motivates the development of automated methods for evaluating a CE's adaptation capabilities and expected performance, which is the second contribution of this article. A CE that can estimate its own performance is preferred because other parts or operators of the system will know what to expect from the CE and plan accordingly.

[1] Best in terms of the time and total data sent until the optimal or a near-optimal communication method is found.

The authors are with the University of Arizona.

0163-6804/16/$25.00

The Metacognitive Solution

We use a metacognitive engine (meta-CE) framework that inherently addresses the mentioned challenges. From the outside, a meta-CE appears to be a regular CE, that is, a CE that provides expected performance estimates (through confidence intervals). On the inside, a meta-CE is made of one or more CEs. A meta-CE has the following features:

- Knows at which operating conditions each CE is best
- Knows the expected performance of each CE and in turn knows its own performance when it chooses a certain CE
- Automates the process of evaluating and selecting the appropriate CE for the operating conditions

Considering all possible channel conditions, the meta-CE performs better than only using one individual CE. The meta-CE, as discussed in the sequel, can also provide knowledge indicators (KIs), which measure how much a CE has learned and how close it is to reaching the state at which there is nothing more to be learned (for that specific operating scenario).

This article presents the metacognitive concept in detail with initial innovations and discussion of future directions for maturing the meta-CE framework. First, we present a brief background on CEs and the origin of the metacognition concept. We then present the elements of metacognition along with key results. Finally, we provide a discussion on future work and concluding remarks.

Background

Brief Background on Cognitive Engines

For nearly two decades, CE designers have been continuously working to understand and develop better learning techniques for CR. They have typically borrowed ideas from machine learning and artificial intelligence [3] to design their CEs. Notable examples include artificial neural networks (ANNs), GA, and CBR [3]. In addition, they used reinforcement learning techniques such as the ε-greedy, softmax, and Gittins index [4, 5]. Furthermore, other techniques such as particle swarm [6] and ant colony optimization are also used to create CEs.

Different types of CEs have their own advantages and disadvantages. Some perform really well in high SNR conditions; others are more effective in low SNR conditions. In addition, providing predictable and more confident performance levels is the most important aspect of designing various CE algorithms. Therefore, the meta-CE is being proposed to provide the mentioned level of prediction for various types of CEs. The first effort was Gadhiok *et al.* [7], who proposed a primitive architecture of metacognition. Moreover, we proposed a more advanced and general meta-CE in [8, 9], which can classify various CE algorithms based on the operating conditions (objective, channel condition, radio capabilities, etc.). The proposed meta-CE employs a general performance characterization method to evaluate the performance of individual CE algorithms. Also, the meta-CE can identify distinct operating scenarios based on the performance level of the individual CEs.

More Details on the Metacognition Concept

The motivation for incorporating metacocognition into a CE comes from psychology. Metacognition is defined as thinking about thinking [10]. There are three interrelated components to metacognition: metacognitive knowledge, monitoring, and control [11]. These components are integrated with the primary cognition, which refers to object-level thoughts or an individual CE algorithm's process. *Metacognitive knowledge* derives from the beliefs individual CEs generate about their decisions (e.g., "the CE does not consider all the communication methods when it is facing low power"). *Metacognitive monitoring* is the process by which an agent evaluates its own thoughts for comparison (e.g., "failing to consider all the available communication methods can lead to choosing an inappropriate one for the channel conditions"). *Metacognitive control* refers to the regulation of the agent's thinking (e.g., "if the CE does not have enough power or time to consider all the communication methods, it needs to stay with the most robust communication method it currently knows"). Achtziger and colleagues [12] note that metacognitive monitoring and control processes connect secondary thinking (meta thoughts) with primary thinking about an object; monitoring processes represent information flowing from the object level to the meta level; and control processes represent information flowing from the meta level to the object level. Metacognitive processes serve a self-regulatory function when monitoring permits secondary thinking to inform the state of primary thinking about an object, and the control processing permits primary thinking to be informed by secondary thinking. Research shows [12, 13] that secondary thoughts can play an independent role in judgment and behavior when they modulate (e.g., increase, decrease, or reverse) the impact of primary cognition. An important outcome of metacognition is the confidence that people perceive in their thinking.

The scope of our work is to implement the needed components of metacognition as a meta-CE. Figure 1 shows the connection between the three metacognition components and the operation of the meta-CE: the meta knowledge translates to calculating the KIs that indicate the status of the learning progress, and learning curves and performance characterization that analyze the performance of the individual CEs. Meta monitoring translates to channel characterization, comparing the performance results of the individual CEs, and monitoring the meta-CE's real-time performance. Finally, metacognitive control translates to selecting the most appropriate CE and adjusting its parameters for the current operating scenario. The following sections elaborate on how each concept affects the meta-CE's operation and our current implementation.

Elements of Metacognition

Metacognitive Engine Operations

Figure 2 shows that the operation of the meta-CE can be summarized in six steps. First, any available prior information such as results from previous experience and learning outcomes is used to provide initial estimation of the expect-

The motivation for incorporating metacocognition into a CE comes from psychology. Metacognition is defined as thinking about thinking. There are three interrelated components to metacognition: metacognitive knowledge, monitoring, and control. These components are integrated with the primary cognition, which refers to object-level thoughts or an individual CE algorithm's process.

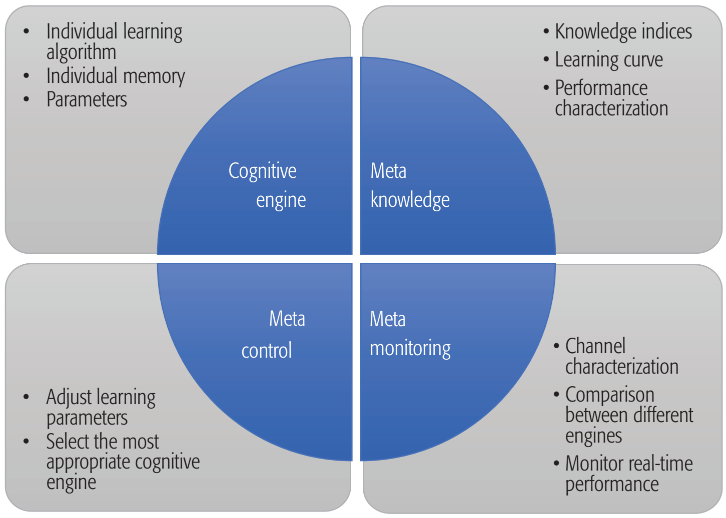

Figure 1. The psychological metacognitive components and their application to a metacognitive radio engine's functions.

ed performance. Second, the performance characterization module uses the prior information and real-time data from the currently selected CE to update the CE's performance and knowledge metrics. Third, the Best Cognitive Algorithm Classification (BCAC) compares the available CEs and identifies the most appropriate CE for each operating region. Fourth, the current operating scenario is updated with the current channel conditions and the radio's operating objective. Fifth, the meta-CE selects the most appropriate CE algorithm for the current operating scenario based on the metacognitive knowledge gathered by the previous steps. Sixth, the selected CE algorithm is used to control the communications link while the operating conditions remain unchanged. Finally, once the CE is done with link adaptation, it feeds its performance data (monitoring) to the CE performance characterization block, and the process repeats.

Following this overview of the meta-CE operations, we now provide more detail on how to realize a meta-CE.

Metacogntive Engine Development

An ideal meta-CE needs to develop the three aforementioned interrelated components that have distinct functionalities (Fig. 1). These components are:
- Meta knowledge
- Meta monitoring
- Meta control

A prerequisite for developing these components is a systematic method for evaluation of performance of CEs [14]. The challenges in this systematic evaluation are due to the following. First, the performance is affected by a multitude of parameters, such as channel parameters, operating objectives, learning techniques, and radio capabilities. Second, the evaluation method must be able to compare performance in different scenarios (channel conditions, objectives, etc.) against each other meaningfully. We have addressed these challenges through utilizing the concept of a "learning curve." A learning curve is a graphic representation of a CE's learning rate.

Third, it is necessary to be able to identify the operating environments (channel) and match them to the performance of the CEs. In theory, there are infinitely many operating conditions (e.g., there are infinite SNR values); however, classification into only a finite number of regions matter (e.g., low SNR, medium SNR, and high SNR). Once the meta-CE can characterize and identify these regions, it can match each region to the CE that performs best in a particular region. We treat this "region to CE matching" process as a classic pattern classification problem, which is BCAC.

A different challenge in the development of meta-CE relates to the metacognitive control component. The role of this component is to make a decision based on the information provided by the other components. An ideal meta-CE should be able to adapt the learning parameters or switch among possible algorithms in order to provide the most effective and efficient way of learning.

Metacognitive Knowledge

Metacognitive knowledge is related to knowledge or belief about an individual CE's decisions and utility. Metacognitive knowledge allows a meta-CE to make sense of performance and understand the state of individual CEs. In order to generate/develop metacognitive knowledge, we need to define metrics and methods that serve this purpose. One of the most utilized indicators is the learning curve. The meta-CE will create each CE's learning curve(s) by observing its operation after numerous operating sessions. The learning curve is characterized by using statistical inference based on the achieved performance by a CE's decisions. The performance is derived from multiple repetitions of the CE algorithm in the same or similar channel conditions.

Figure 3 presents a plot of the learning curves of three CE algorithms. The plot is generated by observing the mean and variance of the achieved rewards of each CE. The reward quantifies how well the operating objective of the CE is met; in this case, the reward is the achieved throughput. Figure 3 shows examples of different CE learning behaviors. However, before explaining these three CEs' algorithms, let us see what kind of information this plot provides for us. For example, CE Algorithm 2 has zero average performance between time steps 0 and 50. Moreover, after time step 180, it quickly reaches 38 Mb/s, which is the maximum possible in this example. On the other hand, CE Algorithms 1 and 3 provide a slower but steadier performance increase. This is because CE Algorithm 2 is more aggressive with risk taking in applying the communication method options. The standard deviation shown by the error bars indicate that CE Algorithm 2 not only has a zero average performance for the first 40 time steps, but also is consistently zero since its standard deviation is also very low. Between time steps 50 and 200 (approximately), its standard deviation increases and peaks, which means that there is a great variability in performance between those steps. This is an intuitive result which captures the fact that the perfor-

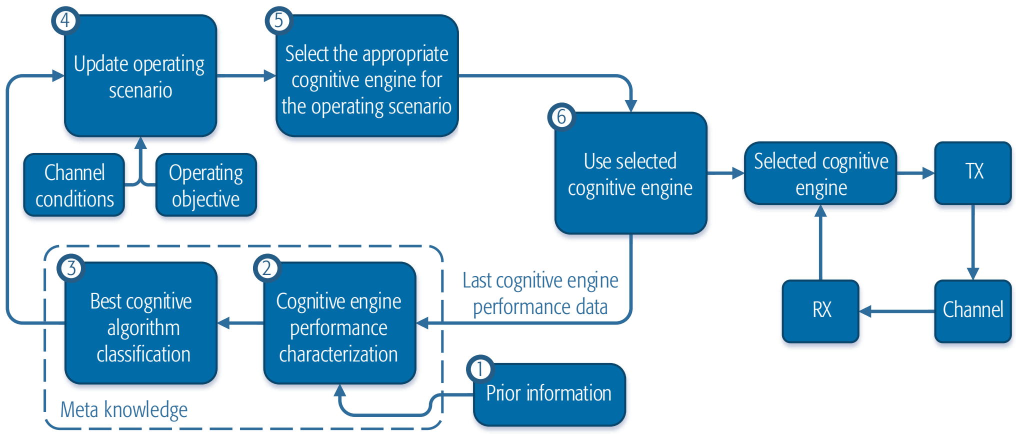

Figure 2. The metacognitive engine operation graph.

In order to generate/develop metacognitive knowledge, we need to define metrics and methods that serve this purpose. One of the most utilized indicators is the learning curve. The meta-CE will create each CE's learning curve(s) by observing its operation after numerous operating sessions.

mance becomes non-zero and reaches near maximum somewhere between the 50th and 200th time step.

Two of the three CE algorithms shown in Fig. 3 are based on the ε-greedy strategy, and the third one is based on the Gittins index. The ε-greedy strategy randomly explores the different communications methods with probability ε and uses the communication method with the highest average throughput with probability 1-ε. The ε parameter controls how aggressively exploration is performed. A higher ε will cause preference for exploration over using existing data. It may result in arriving at an optimal or near-optimal option faster. However, the high exploration rate may also cause reduced overall returns because of the higher exploration cost.

CE Algorithm 1 uses an ε = 0.05 and does not utilize a priori estimates of the maximum possible throughput. CE Algorithm 2 is based on the Gittins index strategy [15] with a normal reward process and a discount factor equal to 0.9. The strategy is simply to use the method with the highest Gittins index, which is based on the reward statistics of each method and must be estimated only when those statistics change (i.e., only when each method is used). Gittins proved that exploration vs. exploitation can be optimally balanced using a dynamic allocation index-based strategy. This strategy maximizes the total sum of rewards collected over a long-term horizon. CE Algorithm 3 uses a more aggressive ε = 0.5; however, ε decreases by 0.001 at each time step. This allows CE Algorithm 3 to decrease the exploration rate as more information is collected about each communication method.

Although observing outcomes of a CE algorithm provides much information about how it learns, the actual amount of knowledge that is obtained by a CE algorithm can vary. For instance, if we achieve a wireless link of 10 Mb/s, we may assume that we have an excellent outcome; however, if we know that the actual potential of this link is more than 100 Mb/s, we know that there is much work to do; we will have to either achieve 100 Mb/s or determine that it is not practically possible. Hence, we need some metrics to show us the true amount of knowledge for each CE algorithm.

In a previous publication [9], we proposed several KIs to estimate the amount of knowledge obtained by different CE algorithms. Different types of KIs focus on various aspects of the learning process. Some KIs reflect the amount of trials attempted by the algorithm, and some KIs depict the level of achieved performance in respect to the maximum achievable performance. To be able to have a general estimation of the amount of obtained knowledge, we utilize the concept of information entropy. This new KI behaves similarly to the CCI indicator presented in [9]. The meta-CE observes the information entropy of the expected performance for each communication method. The variation of the entropy represents the changes in the learned information by a CE. The lower the entropy, the more knowledge is already obtained by a CE algorithm.

Figure 4 illustrates the amount of knowledge obtained by our CE algorithms. It is shown that the level of knowledge CE Algorithm 2 obtained until time step 50 (approximately) matches CE Algorithm 3; however, they have drastically different performance (as shown in Fig. 3). Although CE Algorithm 3 appears to have a performance edge, the obtained knowledge suggests that both CE algorithms significantly improved. This was evident at the 150th time step where CE Algorithm 2 outperformed CE Algorithm 3.

Metacognitive Monitoring

Metamonitoring is the ability to distinguish among different communication scenarios, and use the metacognitive knowledge gained to dynamically determine the best CE algorithm and parameters for communicating in the current scenario. A meta-CE needs to distinguish between the operating scenarios in order to match the obtained metacognitive knowledge of the CE algorithms to the operating scenarios. The goal is to find the best cognitive algorithm for each operating scenario. This is achieved by BCAC in our implementation. Our approach is to facilitate BCAC by using two different classification techniques: offline classification using a support vector machine (SVM) and online classification using *k*-means clustering. Both algorithms rely on a distance function between the feature points. For classification, we use selected features from the channel scenarios.

To classify operating scenarios based on the performance of CE algorithms, we need the ability to distinguish and identify various operating scenarios. In order to achieve this, the meta-CE should extract distinctive features of operating scenarios. Generally, an operating scenario is characterized by the channel conditions, operation objective, and the number of possible communication methods the radio's hardware supports.

In machine learning and statistics, classification is used to group data into sub-populations according to their similarities. This is usually done using a training data set that contains examples belonging to known categories. In our case, a training set is a set of data used in various areas of channel scenarios to discover potentially predictive relationships.

The offline and online methods have their own advantages and disadvantages; their main difference is that the offline method waits until all training data are collected in order to process them, and the online method processes the data as they arrive. Furthermore, by preprocessing

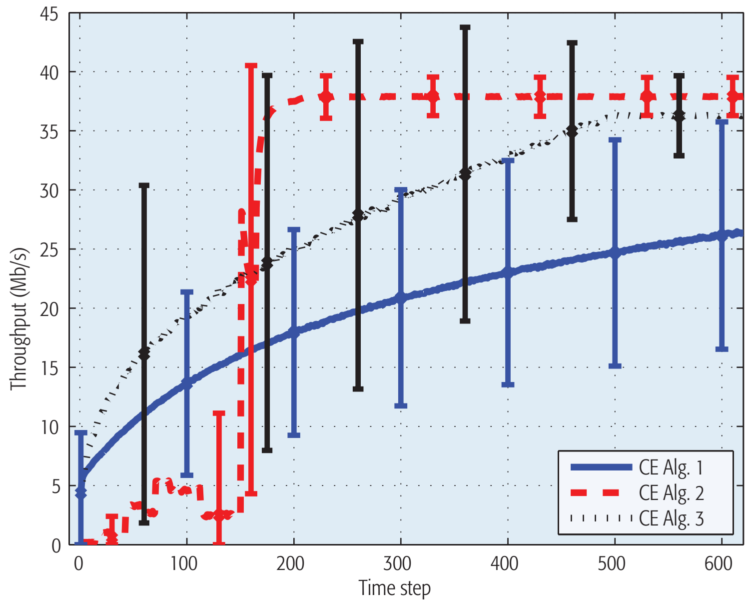

Figure 3. Learning curves based on mean and standard deviation.

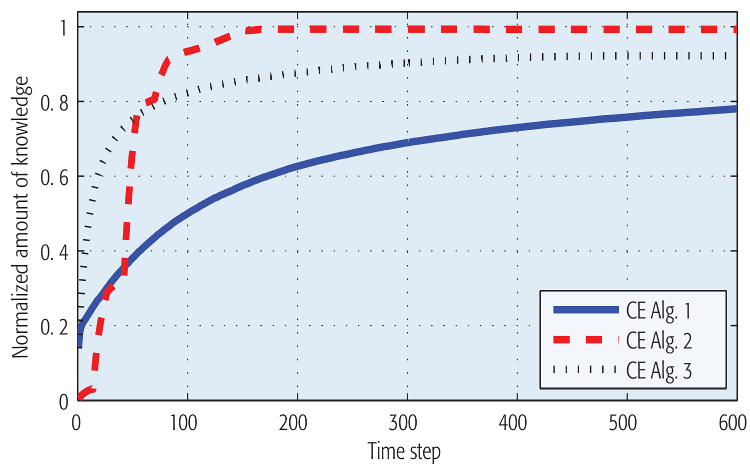

Figure 4. Learning curve based on the estimated amount of knowledge.

the offline training data we are able to create examples of channel conditions and the appropriate CE to be used in those channel conditions. This step makes the offline method a supervised method since it is trained using examples of the desired relationships. On the other hand, the online method is unsupervised since it is not possible to identify the desired relationships until after the data are processed. The offline method is more accurate if the training data cover all the likely scenarios. However, if new scenarios arise, the offline method has to wait until a new batch of data is collected and the classification results are updated. The online method starts adapting as new data become available and gets more accurate over time. Finally, the offline method shifts the computation cost to times when the meta-CE may not be needed and perhaps when the radio is connected to a power source or can offload that processing to the network or cloud. On the other hand, the online method consumes computing and power resources on an ongoing basis.

The following provides more detail on our BCAC implementation. For the offline supervised learning algorithm, the training process works by first mapping data to a feature space so that these data points can be categorized. Then the algorithm finds a function that maps the input data points to distinct classes. In our simulations, we use a predefined training dataset of 200 different operating scenarios. Then the meta-CE uses the classifier's mapping and the features of the ongoing channel conditions to assign the most appropriate CE algorithm for operation. For online classification, generally there is no data available at the start unless prior information and results are available that can be used to initialize the classifier. When no prior information or results are available, at the beginning the meta-CE chooses a CE algorithm randomly (uniformly distributed) among the available algorithms. As the CEs start to operate, the meta-CE collects information about their behavior. The meta-CE compares each operating scenario with similar previously experienced scenarios. The scenarios' similarity is determined by the mahalanobis distance of the channel scenario feature vectors.

The metacognitive monitoring component operates by regularly monitoring the real-time performance of the selected CE algorithm. Therefore, the classification algorithms used to categorize the operating scenarios are continuously updated. The metacognitive control component uses the current monitoring information to make its critical decision for switching among the available CE algorithms or adjusting the learning parameters of a particular CE algorithm.

Metacognitive Control

The metacognitive control component refers to the regulation of the learning process. This regulation can be done by changing the exploration parameters of the learning algorithms, switching between the different algorithms, or by combining two or more CE algorithms to make a new one. In our implementation, the metacognitive control switches among the available CE algorithms by using the provided information from the other components (metacognitive monitoring and metacognitive knowledge), which is based on current operating scenarios.

For instance, at the very first starting point, the metacognitive control chooses among the CE algorithms randomly since it does not have any information about their performance. However, by collecting more data through the monitoring and meta knowledge components, the metacognitive control identifies the CE algorithm that can provide the best outcome (reward) for the ongoing operating scenario.

Implementation Results

To demonstrate the benefits of metacognition in CR, we present a simple example that assumes a 4 × 4 MIMO system with the following modulation options: quadrature phase shift keying (QPSK); 8PSK; 16, 32, 64, 128, and 256-quadrature amplitude modulation (QAM); with error correction rates 1, 7/8, 3/4, 2/3, 1/2, 1/4, 1/6, and 1/8; and multi-antenna techniques: VBLAST, space-time block code (STBC), and maximum ratio combining (MRC). Furthermore, we consider an SNR 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 varying SNR and bandwidth (either 1.25 or 2.5 MHz). At each time step, the CR can transmit over only one of the available channels.

In Fig. 5, we show the results from each CE algorithm and the meta-CE algorithm using offline classification for BCAC. The meta-CE algorithm simply selects the CE algorithm that was found to have the best adaptation performance for the current operating scenario. Best performance is defined as the total throughput achieved during the adaptation session. We used a support vector machine (SVM) as an offline classifier to be trained by 200 operating channel scenarios. We assume that the channel conditions remain static for 100 time steps; therefore, each adaptation session's duration is also 100 time steps. As a result, at time step 0, 100, …, 500 the algorithms have to readjust to the new channel conditions. It was found, that the meta-CE selects with a probability of 92 percent the CE algorithm that is better suited to the given channel conditions. Figure 5 depicts the total data transferred for all 500 time steps (we assume that each time step takes 0.1 ms), which clearly shows the benefit achieved by the meta-CE. The meta-CE transfers a total of 900 kbits vs. 700 kbits transferred by the best individual CE algorithm.

In Fig. 6, we compare the performance of meta-CE, when the online BCAC method is used, with the performance of the individual CEs. The operating objective of the CEs in this example is maximizing throughput. We use the regret concept to compare the CEs. Regret is defined as the difference between the expected reward sum (throughput) using optimal decisions and the actual reward sum that each CE algorithm obtains through its decisions.

Here we compare the results from the offline and online BCAC methods (Figs. 5 and 6). With the offline method (Fig. 5), the meta-CE selects the best CE algorithm 92 percent of the time. When the meta-CE uses the online method (Fig. 6), since it begins with no information, at the start its accuracy is 33 percent with a normalized regret equal to 0.39. This means that the performance of the system with the meta-CE at time step 0 is 61 percent (1 – *regret*) of the ideal throughput. As the online meta-CE learns at each time step, its decision accuracy improves and its regret decreases. Therefore, the meta-CE reaches 85 and 95 percent of the ideal performance after 50 and 120 trials, respectively.

Our results demonstrate that selecting a CE algorithm based on the operating conditions using our meta-CE framework has significant performance advantages. This results in more efficient channel utilization, allowing more users to share the same bandwidth.

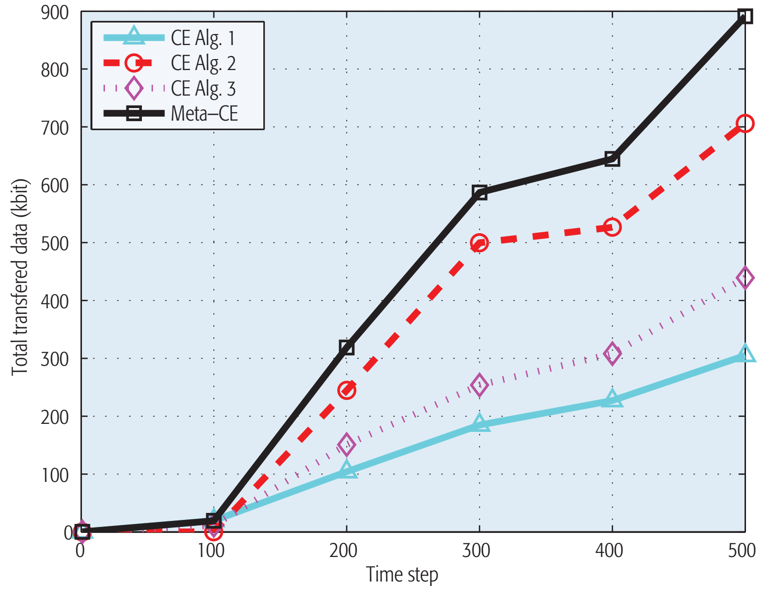

Figure 5. Average metacognitive engine performance (offline classification).

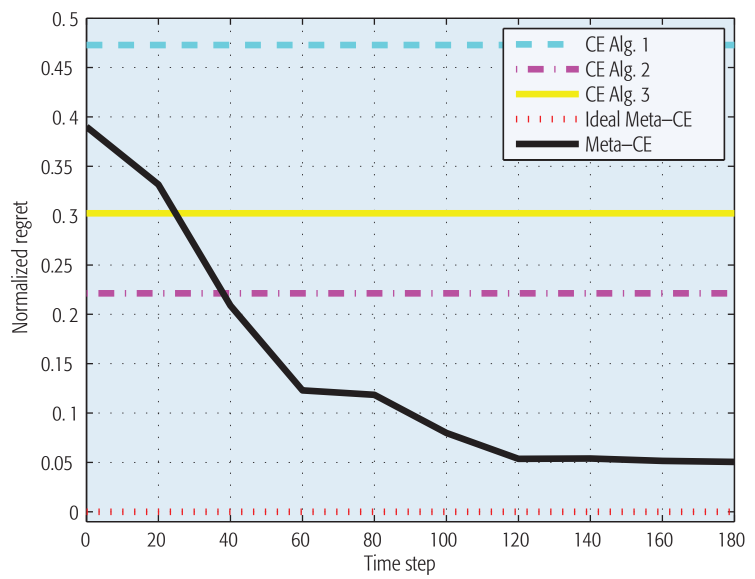

Figure 6. Average metacognitive engine regret (online classification).

A meta-CE inherently solves the problem of characterizing and evaluating the performance of a CE since this process is built-in to the meta-CE. Using this information, a meta-CE can provide estimates of its expected performance. In this work, this information is derived from the performance of individual CEs.

Future Work

Our future plans include the following: first, develop more efficient and mature methods for operating scenario identification and CE selection. The current algorithm classification methods are resource intensive, and their accuracy can be improved. We plan to investigate more appropriate channel metrics to be used for channel scenario identification. Second, assessing the knowledge acquired by a CE and the rate at which that knowledge is acquired is also important. We will therefore work on quantifying the experience level of a CE. Third, we plan to develop a meta-CE that can construct its own CE algorithm by using a set of primitive CE operational elements.

Conclusions

The work presented here shows the evolution from cognitive engines to metacognitive engines for cognitive radio. This evolution can pave the way for another generation of CEs with better performance. A meta-CE inherently solves the problem of characterizing and evaluating the performance of a CE since this process is built into the meta-CE. Using this information, a meta-CE can provide estimates of its expected performance. In this work, this information is derived from the performance of individual CEs. The methods presented in this article are the building blocks for a meta-CE framework, and only the beginning of the development of more advanced and sophisticated techniques and processes for metacognition in radios.

Acknowledgments

This project was partially supported by the Broadband Wireless Access and Applications Center (BWAC); NSF Award No. 1265960.

References

[1] J. Mitola, III, *Cognitive Radio: An Integrated Agent Architecture for Software Defined Radio*, Ph.D. dissertation, KTH, Stockholm, Sweden, May 2000.
[2] S. Haykin, "Cognitive Radio: Brain-Empowered Wireless Communications," *IEEE JSAC*, vol. 23, no. 2, 2005, pp. 201–20.
[3] A. He *et al.*, "A Survey of Artificial Intelligence for Cognitive Radios," *IEEE Trans. Vehic. Tech.*, vol. 59, no. 4, May 2010, pp. 1578–92.
[4] H. I. Volos and R. M. Buehrer, "Cognitive Engine Design for Link Adaptation: An Application to Multi-Antenna Systems," *IEEE Trans. Wireless Commun.*, vol. 9, no. 9, Sept. 2010, pp. 2902–13.
[5] H. I. Volos and R. M. Buehrer, "Cognitive Radio Engine Training," *IEEE Trans. Wireless Commun.*, vol. 11, no. 11, 2012, pp. 3878–89.
[6] Z. Zhao *et al.*, "Cognitive Radio Adaptation Using Particle Swarm Optimization," *Wireless Commun. and Mobile Computing*, vol. 9, no. 7, 2009, pp. 875–81.
[7] M. Gadhiok *et al.*, "Metacognition: Enhancing the Performance of a Cognitive Radio," *2011 IEEE Int'l. Multi-Disciplinary Conf. Cognitive Methods in Situation Awareness and Decision Support*, Feb. 2011, pp. 198–203.
[8] H. Asadi *et al.*, "Metacognitive Radio Engine Design and Standardization," *IEEE JSAC*, vol. 33, no. 4, Apr. 2015, pp. 711–24.
[9] H. Asadi *et al.*, "On Quantifying the Experience Level of a Cognitive Engine," *Proc. SDRWInnComm 2015 Wireless Innovation Conf. Wireless Commun. Technologies and Software Defined Radio*, Mar. 2015, pp. 9–15.
[10] J. H. Flavell, "Metacognition and Cognitive Monitoring: A New Area of Cognitive Developmental Inquiry," *American Psychologist*, vol. 34, no. 10, 1979, pp. 906–11.
[11] J. Dunlosky and J. Metcalfe, *Metacognition*, 1st ed., SAGE Publications, Sept. 2008.
[12] A. Achtziger *et al.*, "Metacognitive Processes in the Self-Regulation of Goal Pursuit," P. Brinol & K. G. DeMarree, Eds., *Social Metacognition*, 2012, pp. 121–39,.
[13] R. E. Petty and P. Briol, "Persuasion: From Single to Multiple to Metacognitive Processes," *Perspectives on Psychological Science*, vol. 3, no. 2, 2008, pp. 137–47.
[14] Y. Zhao *et al.*, "Performance Evaluation of Cognitive Radios: Metrics, Utility Functions, and Methodology," *Proc. IEEE*, vol. 97, no. 4, Apr. 2009, pp. 642–59.
[15] J. C. Gittins, *Multi-Armed Bandit Allocation Indices*, Wiley, 1989.

Biographies

Hamed Asadi (hasadi@arizona.edu) received his B.S. degree in computer engineering from Isfahan University of Technology, Iran, in 2006. He is currently a Ph.D. candidate at the Department of Electrical and Computer Engineering, University of Arizona. His research interests include wireless communication, cognitive-radio engine design, computational intelligence, machine learning, intelligent systems, and dynamic radio frequency mapping. He has also previously done research in image processing and computer vision. He also regularly serves as a reviewer for wireless communication conferences and journals. He received a Best Paper Award from the Wireless Innovation Forum in 2015.

Haris Volos [S'02, M'10, SM'15] (volos@ieee.org) is currently a research assistant professor with the Department of Electrical and Computer Engineering at the University of Arizona. He is an expert in cognitive radio engines with several publications on the topic. His other research interests include channel modeling, software-defined radios, and dynamic radio frequency mapping. He has received several awards such as the Fred W. Ellersick MILCOM Award for the Best Paper in the Unclassified Technical Program in 2010, the Paul E. Torgersen Research Excellence Award in 2011, and a Best Paper Award from the Wireless Innovation Forum in 2015.

Michael M. Marefat [SM] (marefat@arizona.edu) is an associate professor at the Department of Electrical and Computer Engineering, University of Arizona. He is a member of ACM and AAAI. He received his Ph.D. and MS degrees in electrical engineering from Purdue University in1991 and 1988, respectively, and his B.S. in electrical and computer engineering and B.A. in mathematical sciences from Rice University. He is the author/co-author of about 100 refereed research articles. He has directed about 20 M.S. theses and Ph.D. dissertations.

Tamal Bose (tbose@arizona.edu) is the department head and a professor of the Department of Electrical and Computer Engineering at the University of Arizona. He is also the national director of the NSF sponsored multi-university multi-industry wireless research center BWAC. He is the author of *Digital Signal and Image Processing* (Wiley, 2004,) co-author of *Basic Simulation Models of Phase Tracking Devices Using MATLAB* (Morgan & Claypool, 2010), and co-author of *Partial Update Least-Square Adaptive Filtering* (Morgan & Claypool, 2014). His research interests lie in the domain of signal processing for wireless communications.