Cognitive Radio Formulation and Implementation

Thomas W. Rondeau, Bin Le, David Maldonado, David Scaperoth, Charles W. Bostian
Center for Wireless Telecommunications, Virginia Tech, Mail Code 0111, Blacksburg, VA 24061-0111, USA

Abstract— **This paper approaches cognition on the physical and MAC layers by defining a common language of "knobs" and "meters" to discuss adaptation and learning. Cognitive radio merges artificial intelligence and software defined radios (SDR). It requires a simple language for communicating between these two levels. We define a method for doing this. We also discuss a genetic algorithm approach to perform intelligent radio adaptation, using the GNU Radio platform as an example. We provide both conceptual and practical implementation details of a cognitive radio acting at the physical and MAC layers. Results presented show the promise for the genetic algorithm adaptation within the multi-objective optimization environment of the cognitive radio.**

I. Introduction

Much of the current interest in cognitive radio is based on its potential use for dynamic access to spectrum – radios that find and use "white space" with little or no operator intervention. Other applications include modifying radio and waveform parameters to improve link performance or to meet objectives like minimizing occupied bandwidth and maximizing battery life. Unfortunately, because of the lack of a universal definition, we have seen how an emphasis on particular applications has started to narrow the scope of research associated with the general area of cognitive radio.

For the purposes of this paper, we define a cognitive radio as a transceiver that is aware of (a) the RF environment, (b) its own power, frequency, and waveform capabilities, (c) its user's requirements and operating privileges, and (d) the regulations, etiquette, and protocols that govern its operation. Functionally, we picture it as an intelligent software package called a cognitive engine controlling a software defined radio (SDR). The characteristic that makes a radio cognitive and distinguishes it from an "adaptive" or "adaptive-aware" radio is the ability to deal with unanticipated situations and to modify its behavior based on past experience. For more detailed information about cognitive radios, we refer to the reader to our web site [1] and to [2].

As we move forward in the process of developing cognitive radio, we are trying to see how we can get a given radio to emulate our thinking process to reach a level of functionality like that of an expert human operator. The fundamental issue is to create a real cognitive engine that configures a real transceiver to optimize overall performance as defined by its operator subject to a set of constraints.

The optimization challenge faced by the cognitive radio is to develop a better understanding of the interaction between all the communication aspects of the radio's performance. We explore that issue in this paper and also try to provide a better understanding of the costs and benefits of such a radio optimization process using parametric analysis.

Section 2 of this paper presents the problems of link optimization of the physical (PHY) and medium access control (MAC) layers. To work with the radio platforms and the algorithms that cognitive radios require, we develop in Section 3 a language for cognitive radios, and in Section 4, a method for presenting SDR capabilities to the cognitive engine. Section 5 introduces using genetic algorithms to solve the optimization problems, and Section 6 applies this to a practical example of representing and optimizing a GNU Radio SDR. Section 7 concludes the paper with some thoughts on where the research must go next.

II. Optimization in Cognitive Radios

Cognitive radios are all about optimization to suit the applications' and users' needs. While this may seem like a simple idea, the complexities of cognitive radio emerge from the actual methods necessary to achieve this goal. Optimization must occur in many dimensions and is subject to a variety of constraints. We will first introduce the optimization process and then develop our basic approach to implementing it. Throughout this work, we point out many open research questions that still require a considerable amount of work.

A. *Optimization for QoS*

Optimization must take place in three major areas: the user/application domain, compensating for channel propagation effects, and network interactions.

In Mitola's original vision of cognitive radios [3], he focused on user interaction with the radio. While most current work in cognitive radio emphasizes spectrum management and agility, the user remains the fundamental

1-4244-0381-2/06/$20.00 ©2006 IEEE

component of the cognitive radio. The cognitive radio should adapt as best it is able to reflect the applications' and users' preferences for quality of service (QoS). Understanding this optimization domain and providing an effective solution remains one of the most difficult problems in the practical implementation of cognitive radio.

The cognitive engine must both understand and quantify the user's needs. Furthermore, the engine must translate these needs into specific radio actions. The radio could continuously poll the user for this information, or the user could define a desired level of QoS. Either of these two approaches involves user intervention and assumes that the user is expected to understand what he or she needs from the radio. This task may be a bit unrealistic for a standard user. The challenge is to develop passive methods to monitor the activities of the user and from these intelligently develop and model user behavior and QoS needs.

Another important area of optimization is in the link quality of the point-to-point communications between nodes. The propagation channel has a large impact on the QoS that a link delivers. Channel conditions, such as the fading type, fading level, Doppler spread, and path length will greatly impact bit errors which ultimately translate to QoS. How to choose the proper waveform like modulation, channel coding, interleaving, and spreading is an important issue. Embedded here (and in higher layers) is the challenge of *awareness*. What methods can the radio and cognitive engine use to detect and model channel conditions to make the correct decisions?

When extending the optimization problem to the network level, it is important to both quantify the cognitive radio interactions and measure the impact of each radio on others. When networks are interacting and competing for spectrum resources, decisions on waveform adaptation must properly reflect and respect the needs and operations of other radios and networks in the same RF environment. Frequency agility is one way of working in this optimization field, but many other techniques also exist. Orthogonal spreading and modulations allow spectrum reuse in coding, antenna directivity allows spatial reuse, and perhaps cooperative timing schemes will allow spectrum sharing in the time domain.

Each of the optimization domains interact strongly with each other; changing to a more robust channel coding method may improve robustness in a bad channel, but the cost in latency may negatively affect the user's QoS. Later sections of this paper will show how these interactions occur and present methods for understanding, representing, and optimizing overall performance with respect to them.

B. *Limitations*

While the optimization on the PHY and MAC layers tries to build a waveform for the point-to-point link that maximizes the QoS, a cognitive radio must always respect any local regulatory limitations. Spectrum and power are two areas of major concern here. Devices operating in different frequency bands will obviously be subject to different restrictions, for example, those of Part 15 of the FCC specifications [4] or those governing operation in a satellite band. Certain bands would have strong restrictions against certain power level transmission (GPS), locations (TV broadcast), or time (public safety). There is a lot of discussion about opening many of these bands for communications, such as the IEEE 802.22 [5], but there are also legitimate concerns. Any operation in these bands must ensure non-interference with the licensed operators either through sensing techniques [6], a policy/regulatory database and language [7], or a combination of the two.

Of course, optimization for QoS in the different domains means little if the hardware is incapable of supporting the resulting waveform. The hardware architecture will always provide limitations for every decision that the cognitive engine can make. Not only this, but the cognitive radio should realize the constraints imposed by computational power and battery (or system) power available for a given waveform. While OFDM might provide a high data rate and flexible spectrum occupancy, the complexity involved to transmit and receive such a waveform may not make it a practical consideration. A system with limited battery life might find a narrowband modulation technique more suitable for the power limitations while still satisfying the other QoS requirements.

III. COGNITIVE RADIO LANGUAGE

Cognitive radios are realized through the combination of artificial intelligence and flexible (probably software-defined) radio architectures. Looking to the AI literature, the intelligent radio must have *sensors* to read in the external information, *actuators* to effect changes (in the waveform), and an intelligent core to tie the two together [8]. Sensors take in information about user, propagation, and network QoS requirements, and actuators implement the waveform to affect the required QoS. The intelligent core is an intelligent learning machine that develops the relationships between the environmental information and how to develop the waveform. In radio terminology, we refer to the actuators as *knobs* (turn the knob of a radio to adjust the carrier frequency), and sensors as *meters* (read the signal power).

A. *Defining the Parameters (knobs)*

Knobs are those parameters which determine the output waveform of the radio. Table 1 lists the PHY and, to a more limited extent, the MAC layer knobs. The properties listed here are only those that are relevant to the work in this paper. For a more complete list, see [9].

TABLE I. PHY AND MAC KNOBS

Symbol	Meaning
S	Signal Power
B	Bandwidth
R_s	Symbol Rate
Mod	Modulation type
M	Modulation order
PSF	Pulse shape filter type
α, β	Roll-off factor for root-raised cosine or Gaussian filters

B. *Defining the Objective Functions (meters)*

Meters are parameters that allow the radio to recognize its operational environment. From this awareness, it can extrapolate or extract information that relates to any problems with the link quality and calculate how far from the desired QoS the radio is. Some meters can be inherently present within the radio while others can be created by manipulating the information or creating the objective functions. The cognitive radio should be striving to meet the QoS requirements exactly, not going higher or lower. Lower QoS is obviously bad. Higher QoS is also bad because it wastes resources that could drain battery power or reduces the available resources for other users.

The meters of interest in the work of this paper on the PHY and MAC layers include: bit error rate (BER), signal to noise plus interference ratio (SNIR), data rate, occupied bandwidth, spectral efficiency, latency, computational complexity, and power consumption.

C. *Developing a Language of Meters and Knobs*

The meters listed above play a critical role as part of the optimization process of the cognitive radio. As discussed in Section 2B, these objective functions require multi-dimensional analysis as they affect QoS on different levels. These objective functions are also not mutually exclusive; changing a single parameter affects all others in some way, to a greater or lesser extent. Figure 1 shows a single-layer graph of these interactions and how one objective affects others, possibly through an indirect path.

Each meter reflects a dependency on the knobs of Table 1, either directly or through another meter. Table 2 develops these meters as generic objective functions showing the dependency between knobs and meters. These objective functions then become the basis for evaluating the effect of the set of knobs in the optimization algorithm. The mathematical equations for these objective functions can be found in standard texts [10] [11].

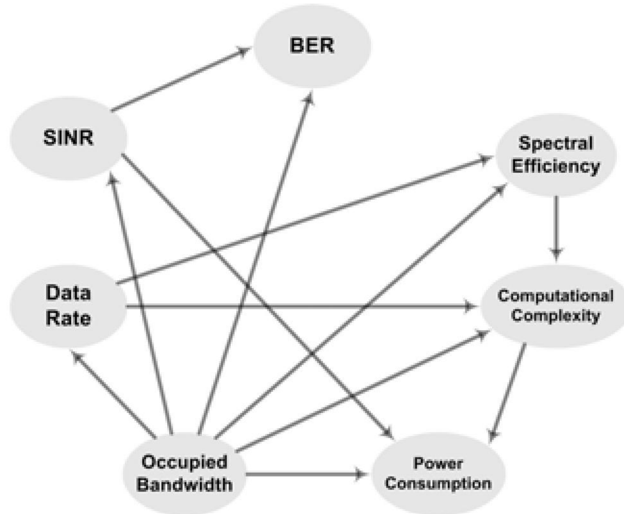

Figure 1. Graph of objective function interactions and dependencies.

TABLE II. OBJECTIVE FUNCTIONS

Objective	Affecting Knobs
BER	$f_{BER}(S, N, I, B, R_s)$
SNIR	$f_{SINR}(S, N, I)$
Data Rate	$f_{DR}(R_s, M)$
Occupied Bandwidth	$f_{OB}(R_s, M, \alpha)$
Spectral Efficiency	$f_{SE}(R_s, M, \alpha)$
Computational Complexity	$f_{CC}(f_{DR}, f_{SE}, f_{OB})$
Power	$f_P(f_{CC}, f_{OB}, f_{SINR})$

IV. DEVELOPING AN INTERFACE

A. *XML – Defining Radio Hardware*

The process of passing information to and from the cognitive radio is a critical design task. The cognitive engine should be designed to be a radio platform and hardware independent. The software framework for the cognitive engine must reflect this independence in its design. This is not a unique problem in modern software engineering.

An inherent difficulty with software implementation is interfacing incompatible applications with each other. One solution, developed in 1996, was the eXtensible Mark-up Language (XML), simple in structure (formal and concise), supported by a wide variety of applications, and human legible [12]. XML provides a generic software solution to give connectivity to programs that were previously difficult to interface. One such application is software defined radio (SDR) and cognitive radio.

One of the most powerful features of an SDR is its ability to be reconfigured in a large number of ways in real time. An XML file can act as an efficient solution to bridge the gap between the cognitive engine and the SDR. In our approach, XML files are generated to act as a middle-ware in order to exchange information between the cognitive engine and the SDR in a way that is independent of the SDR platform used

B. Limitations of the XML in Cognitive Radio

The limitation of using XML as a middle-ware for the cognitive radio is the speed with which the software can complete the task of exchanging information between the cognitive engine and the radio. The ability to parse the XML file and implement the commands may be a bottleneck as systems become more complex. This limitation may become critical for applications that depend on real-time adjustments (i.e., police communication in disaster / emergency situations). Another challenge presented when using XML documents as the middleware for an SDR is the ability to comprehensively define the abilities of the SDR when certain parameters are mutually exclusive, like a particular channel coding and modulation technique.

C. Representing the Radio

We use two XML documents: a SDR definition file and an implementation file. The SDR definition XML file describes the comprehensive capabilities of the radio, including any reconfigurable parameters of the cognitive engine from the PHY layer parameters up to any portion of the protocol stack controlled on the SDR. The SDR definition file contains both discrete and continuous parameters. The discrete parameters (different FEC, modulation, etc.) are distinct sets of information that change the configuration of the radio. Continuous parameters (power level, center frequency, etc.) contain ranges of values that the SDR is capable of implementing. The SDR definition file is created by the designers of the radio. This document needs to be created once the radio development (hardware and software) is complete.

The implementation XML file is a specific realization of the SDR definition XML file. In the implementation XML file, the values are particular instantiations of the parameters defined in the hardware document (e.g., instead of a range of power levels from -50 dBW to 2 dBW, a value of 0 dBW is chosen).

With these two XML documents, the cognitive engine and SDR are able to interface with each other to make a cognitive radio. The use of XML files enables the cognitive engine to be used on potentially any radio that is capable of software reconfiguration. Figure 2 shows the approach for creating the cognitive radio using the XML files and the cognitive engine. The complete set of information is passed from the SDR definition file to a cognitive engine. The cognitive engine will use that search space defined by the SDR definition file to decide which implementation of the SDR is the most effective at that moment. This information will then be combined with any information obtained from sensing the environment. The solution from the cognitive engine is passed to the implementation file that will reflect the final settings of the SDR. For a radio platform that cannot be directly configurable by XML, a hardware-specific application programmable interface (API) can translate the XML information into commands required to reconfigure the radio.

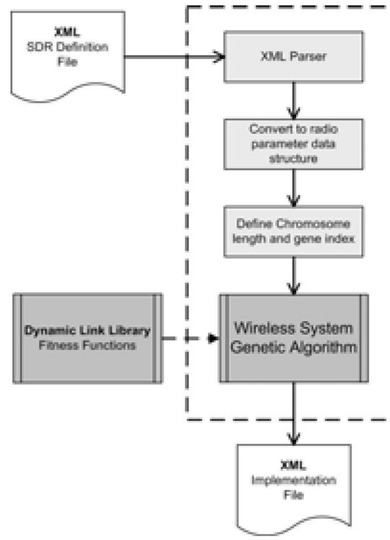

Figure 2. General processing flow of adaptation portion of cognitive engine.

With these two XML documents, the cognitive engine and SDR are able to interface with each other to make a cognitive radio. The use of XML documents enables the cognitive engine to be used on potentially any radio that is capable of software reconfiguration.

V. Genetic Optimization

Radios have traits (knobs) and goals (meters), and we have already seen how they function in terms of artificial intelligence. These properties also allow us to think about the radio in simple biological terms where the sets of knobs determine the particular radio species. Taking this down to the genetic layer, each trait is represented by a *gene* in a *chromosome*. Representing a radio in these terms then allows us to adapt and evolve the radios using genetic algorithms [13]. The radios then can evolve in real-time to adapt to changing environmental conditions.

The genetic formulation is not only convenient for talking about radio adaptation and behavior, but genetic algorithms offer many advantages to the optimization problem. Genetic algorithms (GAs) are powerful and flexible optimization algorithms. They are flexible in their analysis of problems as long as the chromosome and the objective functions are defined properly for a particular domain. The internal behavior of the algorithm is largely domain independent as crossover and mutation operations have little to do with the specifics of the optimization problem. They are also powerful, especially in this application, because of their convergence behavior.

Holland's original work in genetic algorithms proved that GAs will always converge on the optimal solution [14]; however, no one can prove when they will converge. It is widely accepted that GAs take a long time to find the optimal solutions, but, conversely, they take a short time to find very good solutions. Goldberg [15] discusses convergence behavior and shows fast tracking by a GA as the optimization problem changes. Similarly, the radio domain does not require the absolute best solution at any given time, just a good enough solution to maintain a communication link. As radio environments change, the cognitive radio must track the changes, and GAs have been shown to excel at this task.

Finally, the multi-objective optimization problem we describe here defines one of the greatest challenges to cognitive radio realization. Multi-objective optimization problems pose problems that standard optimization techniques cannot often handle, and so they have become their own field of study [16]. GAs have proven themselves well-suited to multi-objective optimization [17] [18]. Below, we explain in greater detail how to apply GAs to the cognitive radio multi-objective optimization problem.

A. *Defining the Radio as a Chromosome*

The first two tasks in creating a genetic algorithm are to define the chromosome structure and develop the fitness, or objective, functions to measure the fitness of the chromosomes. The objective functions have been discussed already and summarized in Table 2. The chromosome must then represent the radio in such a way that it fully defines the radio's behavioral traits and their interdependencies and is useful in the optimization process.

Because the chromosomes of a GA are simply vectors of data structures, the genes are represented by data types like bits, characters, integers, or floating point values. We define genes in a slightly different way, where the basic data structure of the chromosome is a bit, and arbitrary collections of bits are combined to define a gene, more consistent with the biological concepts discussed by Dawkins [19]. A chromosome defined in this way provides an extra level of flexibility to better represent a given radio platform. The idea is to use the minimum number of bits to represent all possible values that a radio knob setting may take without losing the level of precision necessary. A gene for frequency in a spectrally agile radio covering a few GHz may therefore take twenty or more bits to represent all possible frequency values, but the same radio may only have a half a dozen modulation values to choose from, which will require three or four bits.

The chromosome is defined by the radio platform via the XML SDR definitions file to ensure that the chromosome represents all possible values of each knob. Furthermore, the minimum number of genes protects against over representing genes: a gene that only needs four bits but is represented by a thirty two-bit word has two problems. First, it wastes memory. Second, if not all *n*-bit values are defined for the radio, those extra values represent illegal hardware values and therefore create illegal chromosomes, or solutions. Scaling could be performed here, but with the risk of biasing the results from uniformly distributed random variables.

The portion of the chromosome in Figure 2 shows the bit-wise orientation of the chromosome with certain blocks of bits representing the gene. In this radio, there are just over one million possible values to represent the frequency, which represents all possible values for a software defined radio (SDR) that can set its frequency from DC to 6 GHz with 10 kHz steps. The modulation of this radio is limited to sixteen different values, perhaps a few PSK, QAM, and analog modulations. The radio also has a range of 64 values for the transmit power, between, for example, -30 and 30 dBm in steps of 1 dBm.

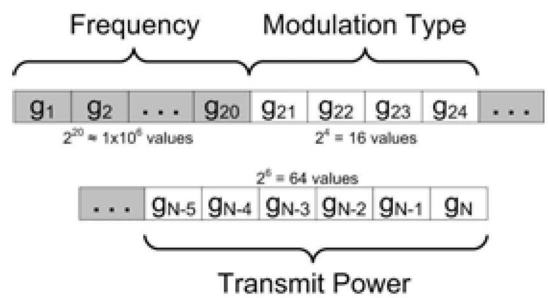

Figure 3. Portion of a sample chromosome.

B. *Genetic Operators: Crossover and Mutation*

The genetic algorithm then performs crossover and mutation in standard ways. There is both a crossover and mutation rate, and the algorithm has a flexible number of crossover points to use. If crossover occurs on two parents to create an offspring, a set of random crossover points are selected in the chromosome. These crossover points can actually split parameter values, possibly cutting the frequency gene in half. However, this cut would preserve the higher frequency value in one offspring and the lower frequency value in the other offspring. These are *schemata* as described in Holland's convergence work [14], and the optimal solution for this radio might exist in the higher frequency bands, and so the first offspring would benefit from this cut. Typical mutation rates should be high, around 90% or more.

Mutation may then occur on any or all bits in the chromosome. For each gene, a random value is generated; if this random value is less than the mutation probability, the gene is mutated by inverting its bit value. Mutation can have a large effect on the structure and fitness of an offspring, so mutation should be kept small, 10% or less. Mutation can also be scaled throughout the generations of the GA, using higher mutation rates when the overall fitness is not improving to expand the search capabilities of the algorithm, or using lower mutation rates when exploitation of fit parents will work better [20].

C. *Multi-objective Optimization Over Single-objective Optimization*

The choice of the radio parameters at all layers affects the radio's behavior in many dimensions. Bit error rate (BER), bandwidth, power consumption, and network latency are just a few examples. Each of these dimensions has some relationship to the QoS, and these relationships change in their relative importance, depending on the application being used as shown in Figure 1. To define this multi-objective optimization problem properly, we use a multi-objective genetic algorithm (MOGA) as a powerful algorithmic approach to adapting a radio autonomously.

The radio user has some desirable operation in mind that values certain goals more than others, such as the minimum latency requirement of a video conference. We associate each optimization dimension in the radio with a weight to delineate the relative importance of the goals in the decision-making process. As the MOGA analyzes each dimension, optimization in the higher-weighted dimensions leads to a solution tailored to the user's preferences.

In wireless communications, like many real-world problems, the interdependence of the operation aspects to each other and to various performance requirements makes it difficult to analyze the system in terms of any one single objective, creating a highly complex search space. Furthermore, the needs of the user and of the network cannot all be met simultaneously, and these needs can change dramatically over time or between applications [17].

Because it could potentially depend on the user and the application, the search space is more complex. For certain users or applications, different objectives will mean different levels of quality. Given that the overall optimization goal is to provide the best quality of service to the user, there is no single search space that can account for all the variations in needs and wants from a given radio-user relationship.

From this analysis, we have developed a few important points about ways to analyze the multiple objectives used in optimizing a radio:

- There are many objectives, making a large N-dimensional search space;
- Different objectives may only be relevant to certain applications/needs;
- The needs and subjective performances of users and applications vary;
- The external environmental conditions determine what objectives are valid and how they are analyzed.

D. *Multi-objective Optimization Methods*

A multi-objective optimization algorithm is a mathematical method for choosing the set of parameters that best optimizes over the set of objective functions.

A basic formula for defining multi-objective optimization is shown below:

$$\begin{aligned} \min/\max\{\bar{y}\} &= f(\bar{x}) = [f_1(\bar{x}), f_2(\bar{x}), \ldots, f_n(\bar{x})] \\ \textit{subject to}: \ \bar{x} &= (x_1, x_2, \ldots, x_m) \in X \\ \bar{y} &= (y_1, y_2, \ldots, y_n) \in Y \end{aligned}$$

Where there are n dimensions in the search space and $f_n(\bar{x})$ defines the mathematical function to evaluate dimension n. Both x, the set of input parameters, and y, the set of dimensions, may be constrained to some space, X and Y. The optimal solutions lie on the *Pareto front*, which is the set of input parameters, x, that is non-dominated in any dimension, which is often a trade-off of goals.

Multi-objective genetic algorithms define multiple fitness functions according to specific objectives and calculate their values on the population of competing chromosomes. The algorithm evolves in order to pick the one (or more) chromosome with the best performance to balance the combined fitness. Through the evolving process, which consists both of genetic operations and selection, the Pareto front moves so that the optimal solution provides the most efficient performance for the user's QoS requirements under radio-domain and regulatory constraints. Here, efficiency and optimization mean providing a QoS without over-maximizing, which may waste radio resources such as spectrum and power. For example, a user sending email does not need a 100 Mbps link with a 30 dB carrier to noise ratio.

E. *Genetic Selection Method*

Multi-objective genetic algorithms have been around for decades, but many methods exist to realize the process of selection and fitness evaluation. The most promising techniques involve an analysis of the solution space with respect to the Pareto front, the set of non-dominated solutions in the population. Both Zitzler [17] and Fonseca [18] provide excellent studies of some of the most popular approaches.

In the genetic algorithm for wireless communications, we have chosen a Pareto ranking method of selection and evaluation. Each member of the population is compared to all other members in all objective domains. Any member that is not dominated in all objectives by another member is a non-dominated solution and given a rank of 1. Other members are ranked by the number of solutions that dominate them. Fitness is therefore a discrete value between 1 and the total population size (where a solution might be dominated by all other solutions). The population can then be sorted in terms of this fitness.

In each generation, parents are selected by a tournament or roulette wheel selection [13]. For the examples presented in this paper, a tournament selection method was used. Offspring are created through the crossover and mutation routines, and the worst parents are replaced by the best offspring. This method preserves parents on the Pareto front and only replaces them if offspring during a generation push the Pareto front closer to the optimum values.

The biggest problem in multi-objective optimization is making the final decision. Evolving the Pareto front to the set of optimum solutions must then be translated in the final generation to a single decision that is a trade-off between the objectives. The chosen solution must therefore represent the user's and application's preferences. This is where the weights enter the algorithm. By weighting, the importance of each objective can help pinpoint a solution on the Pareto front that properly represents the objectives. In a situation where data rate is more critical to the user/application than minimizing occupied bandwidth, choosing a solution that has a higher symbol rate at the expense of a larger bandwidth may be the better one.

VI. Cognitive Engine Examples

A. *GNU Radio as a Cognitive Platform*

The genetic algorithm with the XML interface to the radio platform is our basis for a hardware-independent cognitive engine. To test the adaptation process, we are building a cognitive radio platform using the affordable and available GNU Radio software package and Universal Software Radio Peripheral (USRP) [21] [22]. As an open source project, the GNU Radio allows great flexibility and control over the SDR for realizing different knobs. Much more work needs to be done in the GNU Radio to fully realize the SDR platform, but at this point we are able to simulate basic functions. The following example of the genetic algorithm for cognitive radios is based on what we have done and what we know we can do with the GNU Radio platform. Many more knobs and meters will be developed soon to extend this concept.

GNU Radio Knobs and Meters

Table 3 provides a listing of knobs we can turn on the GNU Radio using the BasicTX and BasicRX daughterboards. These values do not reflect the absolute capabilities of the GNU Radio but are based on current capabilities we have developed and processing limitations of our systems. The frequency has been split into two segments as an exercise instead of a hardware restriction. Table 4 presents the objective functions (meters) currently used by the genetic algorithm.

In Table 4, K is the signal space dimensions (e.g., 1 for PSK and QAM, 2 for BFSK), N is the AWGN noise power and I is the interference power from signal i. In this simulation, no interference is used.

TABLE III. GNU Radio Knobs

Knob	Range
Carrier Frequency (f_c)	DC – 10 MHz (1 kHz steps) 20 MHz – 44 MHz (1 kHz steps)
Transmit Power (P)	-100 – -10 dBm (0.1 dBm steps)
Modulations	PSK (M) QAM (M={8, 16, 32, 64, 128, 256})
Symbol Rate (R_s)	5 kbps – 100 kbps
Pulse Shaping	Root Raised Cosine (α = [0,1]) Gaussian (β = [0,1]) Low pass filter

TABLE IV. GNU Radio Meters

Meter	Function
BER	$BER \propto erfc(E_d)$
Spectral Efficiency	$\eta_B = \frac{R_s \log_2(M)}{BW}$
Occupied Bandwidth	$BW = \frac{R_s}{2} K(1+\alpha)$, for RRC
SNIR	$SNIR = \frac{P}{N + \sum_i I_i}$
Data Rate	$R_b = R_s \log_2(M)$
Power Consumption	$P_c = P R_s K$

Since there is no closed-form solution to the bandwidth for a Gaussian filter, empirical data are used as described in Rappaport [23].

The power consumption is difficult to measure and highly platform-specific. This objective captures both the amount of power used in the analog circuitry to transmit the waveform (such as the power amplifier) as well as the digital power required to process the waveform. In the GNU Radio, analog power depends only on the transmit power and digital power depends on the modulation and symbol rate as all other processing components stay constant. Based on simple experiments with our GNU Radio platform, we approximate the effects on power consumption of the symbol rate to have a linear relationship, and the modulation's effects by the signal space dimensions, K.

The BER calculations are based on standard BER formulas for PSK and QAM modulations, and hence are all proportional to the complimentary error function [10]. Because of this relationship, we can directly compare the BER values by only looking at the energy distance between symbols in a constellation, E_d. The larger E_d, the lower the BER.

For QAM constellations, the energy distance is approximated, in terms of the average symbol power P, as:

$$E_d = \left[\frac{\sqrt{P/2}}{\sum_{i=0}^{K-1} 2^i}\right], \qquad where\, K = \log_4(M)$$

For PSK constellations, we analyzed the energy distance for more generic PAM-PSK, or star, constellations. For these modulations, there is an order, M, as in all constellations, and a number of amplitude levels, C. While this formula reflects the amplitude levels, only single level, standard PSK was used in the simulations.

$$E_d = \frac{\sqrt{P}}{\frac{1}{2\sin(\Delta\varphi)} + (C-1)}, \qquad where\, \Delta\varphi = \frac{2\pi}{M \cdot C}$$

These calculations are much more efficiently computed than the integral in the error function, although more efficient calculations for the error function can be found in [24] and [25] if the BER must be calculated, chapter 7 in [2].

B. *User and Application Profiles*

The method for evaluating the multi-objective decision making problem, as discussed above, involves finding the Pareto-optimal front and then making a decision depending on the importance of each objective, as described by a weight value. To assess the effectiveness of the genetic algorithm optimization process for cognitive radios, we test it against a set of user and application profiles to represent possible uses of the cognitive engine.

Our algorithm used a crossover rate of 95% with four crossover points, a mutation rate of 5%, a population size of ten with 5 members replaced each generation, and a generation limit of two hundred. The example solutions are listed below by the objective functions and weights associated with the profile, starting with the simplest and going to the most complicated multi-objective scenario. The results show typical outputs of the genetic algorithm for the problems. For the more complicated problems, multiple results are shown to understand the complexity of the choice on the Pareto front.

Testing BER minimization (1)

With this profile, the goal is to reduce BER. Regardless of any other radio functions, the BER should be minimized. The genetic algorithm found a solution waveform using QAM64 modulation, a symbol rate of 13 ksps, and a transmit power of -38.6 dBm, for an SNIR of about 32 dB. This waveform produces a BER of $2.7\text{x}10^{-88}$.

Recall that BER values of less than 10^{-6} are all considered equivalent solutions, so the algorithm does not care that it could have minimized the BER even further by, say, increasing the power.

This result also shows that the algorithm has no concern with power output or consumption since it chose a high power with a high-order modulation; this is not the type of modulation a radio would typically use if it was concerned about BER and power consumption.

BER (1) and Power (1)

By adding power as an optimization objective of equal importance, the algorithm behaves accordingly. The first result chose a BPSK waveform with 7 ksps, a transmit power of -76.8 dBm for a BER of 0.113. This waveform selection was a single point on the Pareto front. With nothing else to guide it, the algorithm found the Pareto front and chose a solution with respect to two equal and competing objectives, which in this case drove the power down and the BER up.

Another run of this scenario found a more suitable trade-off under these conditions. A QPSK waveform with 10 ksps, a transmit power of -71 dBm, and a resulting BER of $1\text{x}10^{-3}$.

BER (1) and power (0.5)

A better representation for many user needs is to keep the BER low and push down the power consumption. Under these conditions, the algorithm found a waveform using BPSK for its modulation, a symbol rate of 5 ksps, and a transmit power of -60.6 dBm. The resulting BER for this waveform is $1.5\text{x}10^{-24}$.

In this solution, the power was conserved by keeping a fairly low transmit power (under 10 dB SNIR) with BPSK, the lowest power-consuming modulation, along with a small symbol rate to help minimize the BER. This waveform produced an excellent trade-off in values, but produces the lowest data rate in the system.

Another problem in this scenario is that often the power minimization still takes over in the Pareto front since many of the solutions in the rather small solutions space have better power performance than BER performance. The next experiment looks at using the SNIR value to correct for this problem.

BER (1) power (0.5) SNIR (0.5)

Another approach to optimizing the BER and power trade-offs is to use the SNIR as a way to help keep the balance. By trying to maximize the SNIR, the algorithm keeps a check on power minimization, so it does not go too low, and helps the BER minimization. Under these conditions, the waveform produced used QPSK for its modulation, 5 ksps, and -63 dBm for a BER of $3.2\text{x}10^{-6}$. These values are much more consistently produced under the three objectives conditions and also better represent a useful waveform.

BER (1) power (0.5) SNIR (0.5) Data Rate (0.5)

Again, though, we have a small data rate. If the application in use requires higher data rates, these, too, can be reflected in the algorithm's objective functions. Adding the data rate as an objective produces waveforms that increase the data rates without sacrificing the other objectives too much. One such waveform produced used a modulation of QAM32, a symbol rate of 38 ksps, and a transmit power of -56.1 dBm. With a data rate of 190 kbps, the BER is 8.63x10-3, which is not a great BER for data channels.

Another run under these same conditions produces a waveform with a modulation of QAM64, 22 ksps, and -34.6 dBm transmit power. Here, the Pareto front analysis undercut the power minimization to produce a data rate of 132 kbps and a BER of 5.6×10^{-218}.

BER (1) power (0.5) SNIR (0.5) Data Rate (0.5) Bandwidth (0.5)

A final objective is introduced in this study to look at the effects of minimizing the occupied bandwidth on the system. The first waveform produced found a trade-off in most objectives with QAM8, 41 ksps, -35.5 dBm transmit power, a RRC pulse-shape filter with a roll-off factor of 0.3, and a BER of 0.

The second waveform uses QAM32, 62 ksps, -40.9 dBm transmit power, a pulse-shape filter with a roll-off factor of RRC 0.04, and a BER of 4.94x10-20.

Again, the biggest trade-off is between power and BER. The data rates and bandwidth find a good trade-off, where the first waveform had 123 kbps with an occupied bandwidth of roughly 106.6 kHz. The second waveform had a data rate of 310 kbps and a bandwidth of about 129 kHz. The maximum bandwidth this SDR can produce is using the full 100 ksps with a pulse-shape filter roll-off factor of 1, which would be a 400 kHz bandwidth.

C. Discussion of Results

These results show promise for the genetic algorithm approach to the multi-objective optimization problem in wireless communications optimization. The biggest problem faced in the theory of this approach is to handle the Pareto front as effectively as possible. The algorithm constantly finds the Pareto front, but choosing the point from those non-dominated solutions can be problematic as the trade-offs are often very exclusive. The most immediate example of this problem is the power-BER trade-off that we showed was problematic in most of the scenarios we presented. Although we are trying to avoid unnecessary information, perhaps some domain knowledge is required to protect against obviously bad solutions. Even though a solution with a BER of 0.1 might lie on the Pareto front because of its minimum power consumption, it is not a viable solution in a real system.

VII. Conclusions

In this paper, we have developed an operating definition of a cognitive radio and defined a language for the cognitive engine and the radio platform required to realize the full cognitive radio vision. Here, we include the important concepts of the user- and application-based QoS requirements to define the goals of the cognitive radio. These goals must be met within the boundaries of physical radio challenges like difficult propagation channels and the networking issues required for proper resource sharing. In addition, this needs to be accomplished while under the consideration of all possible legal, etiquette, and regulatory constraints.

The approach required to optimize the cognitive radio is multi-objective in nature; many different and competing objectives must be concurrently evaluated to determine a solution. In light of all of the issues in creating a cognitive radio, we have developed a useful and robust method of cognitive radio adaptation utilizing the evolutionary nature of genetic algorithms. We have both described how this formulation works and presented examples of its implementation using a GNU Radio system platform.

We still face many challenges in cognitive radio developments. Foremost among these issues are developing environmental awareness techniques, more specifically in understanding the user's and the applications' QoS needs. Much work has been done in spectrum monitoring and modeling, including our own work on the subject [26]. As a continuation of our work, we must make use of these techniques and the PHY and MAC layer adaptation to move from the point-to-point optimization to the networking concepts of end-to-end optimization. In this area, we must develop the knobs and meters appropriately to reflect the needs of the system.

One complication is the latency involved in receiving information about the network in order to properly use it in the optimization process. Often propagation through a network hides many of the underlying problems involved, and these problems may take a long time to see, model, and react to. In addition, the effect at the network layer on any action taken by a node might not be reflected immediately, adding complexity to the optimization feedback mechanism.

Another issue that must be addressed in our cognitive engine is that our approach generates cognitive radio species, which cannot necessarily talk to other species of radios. For the adaptation to be meaningful, we must create a method by which radio species can communicate/share their chromosomes with other radios.

Acknowledgment

This work was supported by the National Science Foundation under grants 9983463, DGE-9987586, and CNS-0519959 and by the National Institute of Justice, Office of Justice Programs, US Department of Justice under Award No. 2005-IJ-CX-K017. The opinions, findings, and conclusions or recommendations expressed are those of the author(s) and do not necessarily reflect the views of the National Science Foundation or the Department of Justice.

References

[1] Wireless @ Virginia Tech, http://www.cognitiveradio.wireless.vt.edu/, 2006.

[2] B. Fette, *Cognitive Radio Technology*, New York: Elsevier, 2006.

[3] J. Mitola, "Cognitive Radio for Flexible Mobile Multimedia Communications," presented at IEEE International Workshop on Mobile Multimedia Communications, 1999.

[4] FCC, "Part 15 - Radio Frequency Devices," 2006, pp. 1-155.

[5] IEEE 802.22, http://www.ieee802.org/22/, 2006.

[6] S. Haykin, "Cognitive Radio: Brain-Empowered Wireless Communications," *IEEE Trans. Selected Areas in Communications*, vol. 23, pp. 201 - 220, 2005.

[7] XG, http://www.darpa.mil/ato/programs/xg, 2005.

[8] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 2 ed, Upper Saddle River, New Jersey: Pearson Education, Inc., 2003.

[9] T. W. Rondeau, C. W. Bostian, David Maldonado, A. Ferguson, S. Ball, B. Le, and S. Midkiff, "Cognitive Radios in Public Safety and Spectrum Management," presented at Proc. TPRC, Arlington, VA, 2005.

[10] J. G. Proakis, *Digital Communications*, 4 ed, New York: McGraw Hill, 2000.

[11] L. W. Couch, *Digital and Analog Communication Systems*, 6 ed: Prentice Hall, 2001.

[12] W3C XML, http://www.w3.org/XML/, 2003.

[13] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Reading, MA: Addison-Wesley, 1989.

[14] J. Holland, *Adaptation in Natural and Artificial Systems*, Ann Arbor: University of Michingan Press, 1975.

[15] D. E. Goldberg and R. E. Smith, "Nonstationary function optimization using genetic algorithms with dominance and diploidy," in *Proc. of the 2nd International Conference on Genetic Algorithms*, 1987, pp. 59-68.

[16] C. Hwang and A. Syeed, *Multiple Objective Decision Making - Methods and Applications*, New York: Springer-Verlag, 1979.

[17] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms - a comparative case study and the strength Pareto approach," *IEEE Trans. Evolutionary Computation*, vol. 3, pp. 257 - 271, 1999.

[18] C. M. Fonseca and P. J. Fleming, "An overview of evolutionary algorithms in multiobjective optimisation," *Evolutionary Computation*, vol. 3, pp. 1-16, 1995.

[19] R. Dawkins, *The Selfish Gene*, New York: Oxford University Press, 1976.

[20] D. Thierens, "Adaptive mutation rate control schemes in genetic algorithms," presented at IEEE Proc. CEC, 2002.

[21] GNU Radio, http://www.gnu.org/software/gnuradio/, 2006.

[22] Ettus Research, LLC, http://www.ettus.com/, 2006.

[23] T. Rapparport, *Wireless Communications: Principles and Practices*, 2 ed, Upper Saddle River, NJ: Prentice Hall, 2002.

[24] D. L. Decker, "Computer Evaluation of the Complimentary Error Function," *American Journal of Physics*, 1975.

[25] M. Abramowitz and I. A. Stegun, *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*, New York: Dover, 1972.

[26] B. Le, "Modulation Identification Using Neural Network for Cognitive Radios," presented at Software Defined Radio Forum Technical Conference, Anaheim, CA, 2005.