This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the ICC 2008 workshop proceedings.

Cognitive Radio Testbed: Exploiting Limited Feedback in Tomorrow's Wireless Communication Networks

Christian Sokolowski, Marina Petrova, Alexandre de Baynast, Petri Mähönen
Department of Wireless Networks, RWTH Aachen University
Kackertstrasse 9, D-52072 Aachen, Germany

***Abstract*— The next generation of wireless communication devices should support advanced features such as high spectral efficiency, broad bandwidth, diverse Quality of Service (QoS) requirements, and adaptivity. The Cognitive Radio (CR) is a new paradigm which has a high potential to become a basis for the future wireless systems. This paper is a first step towards the implementation of such a system. Our CR testbed is based on a GNU Radio platform which enables flexibility and reconfigurability of transmission parameters. As machine learning component, we invoke Genetic Algorithm (GA) to optimize the transmission parameters such as transmission power, modulation order and frequency channel based on the current spectrum conditions. Unlike other CR implementations, our approach requires very limited feedback information at the transmitter ($\approx$ 8 bits/packet duration). No transmission model nor additional network state information (NSI)is needed at the transmitter side. Experimentations show that our CR is capable to find free channels within 4-5 iterations even in a highly occupied spectrum scenario. It also offers the optimal trade-off between throughput, reliability, and power consumption depending on the user's QoS requirements.**

I. Introduction

The Cognitive Radio paradigm introduced by Mitola III and Maguire [1] has created new opportunities for building reconfigurable and optimized wireless systems. So far the main focus of exploiting the potential of the CRs has been in the field of dynamic spectrum access (DSA). However, the CR paradigm gives excellent possibilities for cross-layer optimization and achieving optimal system performance using machine learning and reasoning techniques.

Recently different research groups have proposed concepts and frameworks for CR using machine learning based optimization techniques for the decision making process. An early approach was introduced by Rieser *et al.* [2] at Virginia Tech., known as Cognitive Engine. It is a simple genetic algorithm based engine for cognitive radios which can change the transmission parameters of the radio based on a set of predefined objectives. Petrova *et al.* [3], [4] extended their approach towards a cross-layer paradigm for Cognitive Radios considering all OSI layers, and proposed a framework called Cognitive Resource Manager (CRM). Similar work has also been started by Sutton *et al.* [5] at Trinity College Dublin in Ireland. Recently Newman *et al.* [6] from Kansas University derived both single carrier and multiple carrier fitness functions for a genetic algorithm driven CR engine. All these approaches require perfect knowledge of the transmission model at the transmitter side and therefore induce a large overhead. In a time-varying radio environment, it is not clear if such schemes will increase the overall throughput.

In this paper we propose a new implementation of CR on a GNU Radio hardware platform [7], which requires very little information. We employ a genetic algorithm (GA) to optimize the transmission parameters in a dynamic wireless radio environment. As feedback information we only use the error rate estimate of the transmission, so that no additional network state information (NSI) estimation is needed. For example, NSI may consists of the channel SNR, number of users, interference level, transmission range, channel impulse response, etc. The NSI estimation introduces large overhead especially in time-varying environment, which decreases the overall throughput of the system. Our implementation requires very limited feedback information at the transmitter ($\approx$ 8 bits/packet duration) and is still optimal for slow to moderate fading environment. Throughout the paper, we focus on the optimization of the parameters at the physical layer. However, our approach can be extended to support additional parameters. The CR is able to dynamically adapt the transmission parameters to a changing spectrum environment and meet the users' current needs, e.g., jump to a "free" channel and offer the optimal trade-off between throughput, reliability, and power consumption depending on the user's QoS requirements. Our system operates in the unlicensed 433 MHz ISM band, but can be extended to other bands by switching the front-end board.

The remainder of the paper is structured as follows: in Section II we propose a new transmission scheme with limited feedback in which the resources are optimized according to the user's requirements. Section III gives an overview of our implementation based on GNU Radio. In Section IV we present the experimental results of two test cases: 1) rate adaptation and power control and 2) dynamic spectrum access in time-varying spectrum scenario. Finally we conclude the paper in Section V.

II. Closed-loop GA-based CR Transceiver

Let us assume a closed-loop system as in Figure 1. For the transmission of the first packet, a random set of transmission parameters is used. The radio parameters that we consider in this paper are the transmit power (16 different power levels), the order of Phase Shift Keying (PSK) modulation (1 to 4) and the carrier frequency within 428 MHz-459 MHz range (32 channels). The receiver attempts to detect the packet, estimates the number of errors within the packet and sends this value back to the transmitter via the control channel. Assuming that the transmission channel is constant over several packet durations, the number of errors is a measure of the reliability of the transmission for the current set of transmission parameters. Based on this number, the transmitter estimates the probability of transmission errors P_e for each packet sent. These probabilities will be used by the GA to optimize and time-tune the transmission parameters.

In general, there are several objectives that radio systems may want to achieve such as minimizing the bit error rate (BER), maximizing spectral efficiency, minimizing power consumption, providing interference avoidance, maximizing data rate, minimizing network latency, etc. In this work we consider three objectives that represent common wireless communication goals:

A. *Minimizing the bit/packet error rate* increases the reliability of the radio transmission . This objective can formulated as

$$f_{min\_BER} := 1 - (1 - P_e)^N, \tag{1}$$

where N is the packet length and P_e is probability of transmission errors.

B. *Maximizing the throughput* is also an important objective and it is given by

$$f_{max\_tp} := \frac{k}{k_{max}}, \tag{2}$$

where k is the number of bits per symbol and k_{max} the maximum order. In our implementation, we choose $k_{max} = 4$.

C. *Minimizing the transmission power* aims to minimize power consumption and interference with other transmissions. In the case of portable devices with limited battery power this issue is very important. Additionally, transmission power has to be also set low in order to limit the interference with the other transmissions. The objective function can be stated as

$$f_{min\_power} := 1 - \frac{P_{Tx}}{P_{Tx\_max}}, \tag{3}$$

where P_{Tx} is the transmission power and P_{Tx_max} is the maximum available transmission power.

Determining the optimal set of decision variables for a single objective, e.g., minimizing power, often results in a non-optimal set with respect to the other objectives. Consequently, if we take all of the aforementioned objectives as inputs to a multiobjective optimization, they need to be ranked with regard to the transmission requirements. In our work we use a weighted, aggregate sum approach where each objective is weighted according to its importance. This approach has been presented in [8] as an attempt to maximize the sum of the positively normalized, weighted single objective fitness scores for each parameter set. In [9] Rondeau et al. use also this approach to autonomously adapt a cognitive radio. Aggregating our three fitness objectives, the scalar fitness function holds:

$$f = w_1\, f_{min\_BER} \,+\, w_2\, f_{max\_tp} \,+\, w_3\, f_{min\_power}. \tag{4}$$

This method suits the Cognitive Radio scenario well [6] since any solution with respect to the user's QoS requirements can be reached simply by changing the weights in (4).

After L consecutive packets have been transmitted, the objectives in (1), (2) and (3) are evaluated for the L corresponding sets of transmission parameters. GA generates L new sets of transmission parameters with better average scores with respect to the overall fitness function in (4). Throughout the paper, L will denote the population size of GA. For more details on the basic operations of GA (selection, crossover, mutation), the reader is refereed to [10]. The transmission of the next L packets proceeds with these new sets of parameters, which will be evaluated in turn at the transmitter. This iterative optimization process repeats for several iterations until (4) is maximized.

III. CR IMPLEMENTATION ON GNU RADIO PLATFORM

The Cognitive Radio setup used in our experiments is shown in Fig. 1. The setup comprises two USRP motherboards from Ettus Research that are equipped with RFX 400 daughter boards (RF front-end). One daughterboard is used for transmitting the data and the other one is used as a receiver. All components required for the RF/baseband conversion are implemented on the FPGA of the daughterboard. Baseband operations are executed in the GNU Radio software environment which is running on the host computer. The USRP hardware is connected to the host computer via a USB connection. This interface is utilized for programming the hardware and for carrying the data stream between the GNU Radio flow graphs and the USRP boards. Throughout the paper we use the same

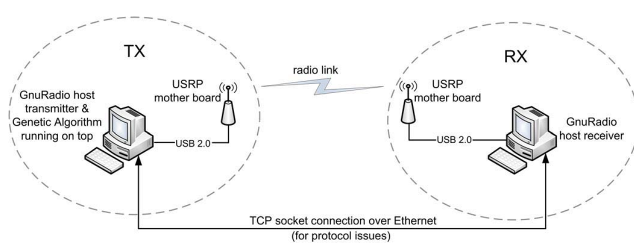

Fig. 1. Testing setup of a Cognitive Radio.

structure for all packets, which consists in the concatenation of three parts: 1) a preamble for synchronization, 2) the payload length, 3) and the payload itself. The preamble is used by the receiver to detect the beginning of a packet. It consists of a 64-bits pseudo-noise sequence [11] which was proven to be efficient even in the presence of frequency selective channels. Our implementation supports variable length packets ranging from 1 byte to 4096 bytes. The packet length is duplicated in order to mitigate the channel impairments that may occur during the transmission. The payload constitutes the third part of the packet. The modulation order k, used for the current packet is transmitted to the receiver before the packet transmission through a wired TCP control channel. The receiver sends the feedback information to the transmitter through the control channel. Our feedback information is relayed through error-free perfect channel. This is done without losing generality, and the purpose of this is to find out the maximum gain that can be achieved for the data communications.

A. *Transmitter flow graph*

Assume that the current solution of the genetic algorithm consists of the following set of parameters: k (modulation order), P_{Tx} (transmission power), and i (carrier frequency). The packet to be transmitted is modulated with k-PSK first. The differential PSK (DPSK) modulated signal is obtained by encoding only the phase changes of two consecutive symbols. Since a DPSK modulated signal considers the phase changes only, the receiver is insensitive to a phase offset. This significantly reduces its implementation complexity [12]. Additionally, the differential schemes might yield a better bit-error rate over wireless transmission than the non differential schemes in the presence of residual phase offset. The digital signal is then interpolated and filtered using a root-raised cosine finite impulse response (FIR) filter. Before the signal is fed to the USRP board, the digital gain of the signal is set to P_{Tx}.

B. *Receiver flow graph*

The receiver structure for the GNU Radio is shown in Fig. 2, where all the elementary processing blocks of the implemented receiver are depicted.

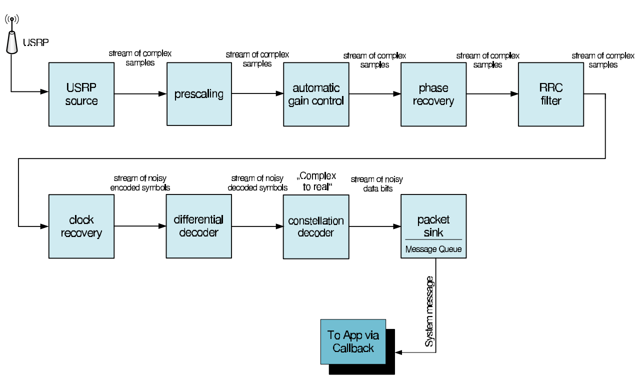

Fig. 2. Flow graph of the receiver.

First, the received data are scaled with an automatic gain control unit in order to mitigate the channel fading that may occur during the wireless propagation. For the phase recovery, we use a Costas Loop-based phase detector [13] with parameters $\alpha = 0.001$ and $\beta = \alpha^2/4$ as suggested by [14]. The signal is then convolved with the matched root-raised cosine FIR filter to remove the high frequency transitions of the signal. In the next block, a Mueller & Müller discrete-time error-tracking synchronizer [15] performs the clock recovery. The signal then is demodulated.

In our packet synchronization implementation, all demodulated and detected symbols are continuously stored in a First-In-First-Out (FIFO) buffer of the length of the preamble (64 bits). At the end of each symbol period, the current buffer sequence is shifted to the left by one bit; we then calculate the bit-wise correlation between this sequence and the preamble, which is known by the receiver. If this coefficient is larger than a threshold τ, we declare the synchronization successful. In our implementation, we found that $\tau = 2k$ offers the best performance. Once the beginning of a packet is detected, the payload length (N) is detected. If the payload length does not match its counterpart or if the length is out of range, the packet is declared lost, the system switches back to its initial state and searches for the next preamble. Otherwise, the receiver start to detect the payload symbols until the packet length is reached. The transmission errors for this packet are estimated and this number is fed back to the transmitter. This number is the sole environmental input used in our GA implementation for iteratively optimizing the transmission parameters through GA.

IV. Cognitive radio testbed results

The performance of GA is experimentally analyzed in two cases that are essential towards practical implementation of a cognitive radio:

A. Optimal power control and transmission rate adaptation in time-varying environment for fixed channel assignment: In this study, we use 16 transmission power levels P_{Tx} ranging from 100 to 15000 and four modulation orders $k = 1, 2, 3, 4$ but our approach can readily be extended to arbitrary modulations. This gives a total search space of 64. Although it might not be necessary to use GA for solving such a problem, our goal is to demonstrate that GA is a generic algorithm for optimizing the wireless transmission parameters and therefore is a strong candidate for managing the resources in CR network. For example, we are currently extending this approach to more sophisticated transmission systems like OFDM and CSMA/CA. In that case, since GA is a generic optimization tool, our implementation needs only minor modifications.

B. Dynamic spectrum access in a time-varying jammed spectrum environment: In this case, 32 channels of 40 kHz bandwidth are possible candidates for the optimal choice of the GA. The channels are between 428 MHz and 459 MHz, the center frequencies separated by 1 MHz guard intervals.

The goal of CR is to quickly adapt the transmission parameters in (4) for a given set of weights according to the current propagation environment. The main criterion for a fast but still reliable optimization process is the convergence speed of GA. The convergence speed factor can be expressed as the product of the population size and the number of iterations. We

TABLE I
INTERNAL PARAMETERS USED IN OUR GENETIC ALGORITHM IMPLEMENTATION.

population size	18 (Section IV-A) or 16 (Section IV-B)
number of iterations	15 (Section IV-A) or 11 (Section IV-B)
selection method	tournament
elitism rate	10%
mutation rate μ	$1 - 1.8^{-1/L}$ [16]
crossover rate	90%

choose the population size and the number of iterations that offer the best tradeoff between the convergence speed and the performance. Table I summarizes the set of internal parameters that we use for GA in all our tests.

A. GA-based rate adaptation and power control

A ideal system should be able to use the minimum transmission power for a target throughput and a target BER. In order to adapt the transmission parameters in time-varying environment, this strategy requires two features: power control and rate adaptation. For converging to a specific set of transmission parameters that satisfy the QoS requirements of the application, the weights of GA in (4) have to be modified accordingly. If the goal is to maximize the throughput for any target BER ranging from 10^{-3} to 0.3, we found that the weight for (2) should be increased approximately linearly with respect to the signal-to-noise ratio (SNR). If the weight for (3) remains constant, the weight for (1) should decrease accordingly. Specifically, the optimal weight for the throughput is equal to 0.01 for SNR $= 8$ decibels and rises up to 0.4 for SNR $= 18$ decibels.

A system supporting rate adaptation maximizes the transmission rate with respect to the SNR of the channel. For example, in IEEE 802.11n standard, modulation schemes ranging from binary PSK (BPSK) to 64-QAM are supported [17]. From this point of view, the system will be optimal. However, it requires the estimates of the SNR of all subcarriers and induces large overhead, which significantly reduces the overall throughput. In order to overcome this shortcoming, our GA-based implementation iteratively optimizes the rate and the transmission power based on the estimation of the probability of transmission errors only. The throughput achieved by GA after several iterations is depicted in Fig. 3. Each solution is averaged over 100 experiments. As aforementioned, we modify the weights in (4) with respect to the transmission SNR such that the throughput is maximized while satisfying a target BER≈ 0.1 (before forward-error coding). We compare our GA-based results with the theoretical performance of PSK modulation. Below a certain SNR threshold depending on the modulation order, it is not possible to transmit data such that the number of transmission errors is equal to or less than target BER. For example, transmission SNR for quaternary DPSK (DQPSK) signaling has to be equal to or greater than 6 decibels in order to meet a target BER of 0.1 [12]. Although GA may not find the optimal solution, the results in Fig. 3 show that GA performs very well and the solution after convergence is near-optimal for a large SNR range.

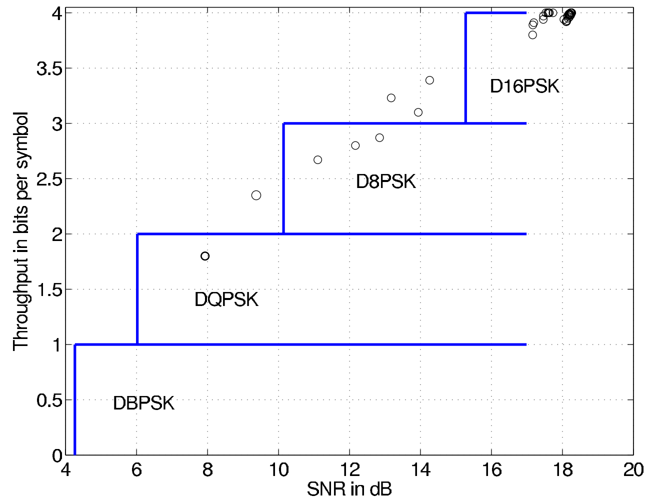

Fig. 3. Weight determination as a function of the transmission SNR in order to achieve optimal rate.

B. Dynamical spectrum access in time-varying jammed spectrum scenario

In a CR setup which supports dynamic spectrum access, some of the available channels might spontaneously be jammed during the transmission. In this section, we investigate the capability of GA to select transmission channels with lower interference level.

In our experiments, no channel is initially jammed; all channels have approximately the same background noise and GA iteratively optimizes the transmission parameters. After 48 packet transmissions (which exactly corresponds to 3 full GA iterations), a subset of the available channels is suddenly jammed. For simulating jamming, we set a high resolution signal generator to transmit random signal with very high power (8 dBm) in the band of interest such that no transmission is possible. Indeed, the power spectral density in the jammed spectrum is approximately 40 decibels higher than the noise power in the "free" channels. In our experiments, we vary the ratio between the number of jammed channels and the number of "free" channels. This ratio is referred as the occupancy level of the system. Specifically, we consider 32 channels of bandwidth 40 kHz as possible candidates for GA. Our analysis includes occupancy levels of 10%, 30%, 50%, 70%, and 90%. The last case, in which 90% of the channels are jammed is the most challenging optimization problem because only 3 out of 32 channels ensure a reliable transmission. In order to steer the GA population to the solutions that correspond to free channels, we stress the transmission reliability at the expense of the transmit power and the throughput by setting the weights to $w_{min\_BER} = 0.8$ and $w_{min\_power} = w_{max\_tp} = 0.1$. The single fitness f_{min_BER} is the only objective which is in function of the channel conditions. We thus assure with high probability that the fitness score (4) of the chromosomes that

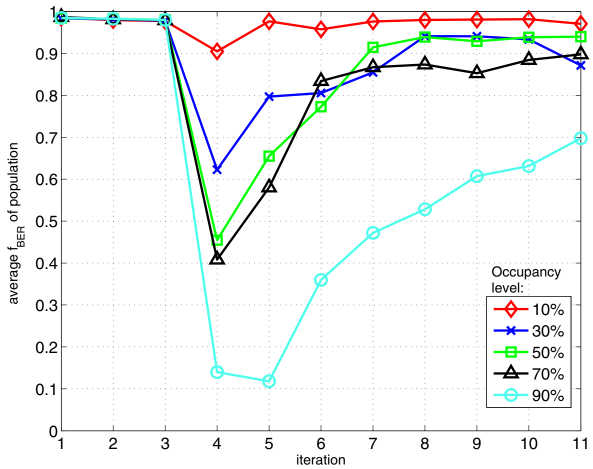

Fig. 4. Channel adaptation for several spectrum occupancy levels. Convergence behavior of f_{min_BER} (1) as a function of the number of iterations.

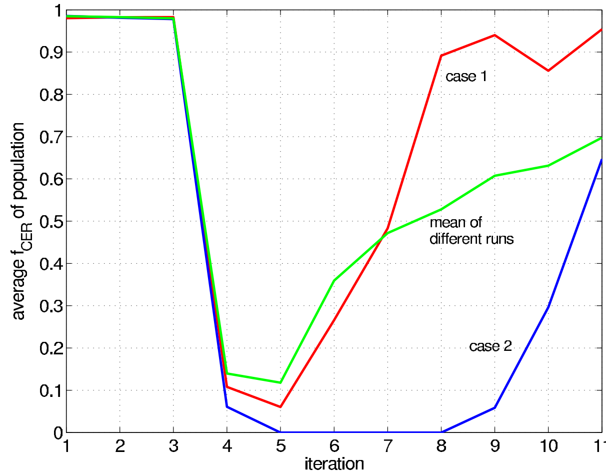

Fig. 5. Convergence analysis for the scenario where 90% of the channels are jammed.

correspond to a high throughput or to a low transmit power level but to a jammed channel is lower than the score of the chromosomes that correspond to free channels. For sake of simplicity, we only use DBPSK and DQPSK modulation for transmission. Additionally, we consider 32 channels between 428.0 MHz and 431.1 MHz with carrier frequencies separated by 100 kHz. The transmission power can take 16 levels ranging from 100 to 15000.

In order to analyze the convergence behavior of GA when the spectrum is partially jammed, the average fitness value f_{min_BER} in (1) is evaluated over all chromosomes. The goal of our GA-based implementation is that all chromosomes in the GA population correspond to some free channels for two reasons. First, all packets that are transmitted over jammed channels are lost and require retransmission which might significantly reduce the throughput. Second, transmitting over jammed channels interferes with signals of the other users. It is essential that all elements in the GA population that correspond to jammed channels are removed in very few iterations. Note that this strategy might not be sufficient if transmission within licensed bands is considered but is acceptable in an unlicensed band scenario.

The convergence behavior of the GA algorithm is illustrated in Fig. 4. Since f_{min_BER} is the main indicator of the transmission reliability, the average fitness score of f_{min_BER} is plotted as a function of the number of iterations. We average the scores over 6 runs for each occupancy level. A fitness score of 0.98 is observed before jamming part of the spectrum. During this period, transmission is reliable over all 32 channels because all of them are free. Since the ambient noise is similar for the 32 channels in our experiments, GA optimizes the rate and the transmission power and transmits over an arbitrary channel which changes from one packet to another.

The optimization difficulty rises after iteration 3 when some of the 32 channels remain "free". In order to limit the interference with potential primary users and to guarantee reliable transmission, population of GA has to quickly evolve to free channels. In Fig. 4, f_{min_BER} drops approximatively to one minus the occupancy level at iteration four. Since the transmission SNRs are similar over all channels, GA population is spread approximatively uniformly over them at iteration 3. Right after jamming a certain ratio of the 32 channels, all packets in the jammed channels are lost and most of the packets in the free channels are successfully transmitted. For example, in the scenario in which 50% of the channels are jammed, the average fitness score drops from 0.9803 down to 0.4546. A similar behavior is observed for the other occupancy cases. In most cases, the average fitness score of f_{min_BER} reaches its initial value (before the jammer) within next few iterations, 3 or 4 in our cases. When a very large number of channels are jammed (90% occupancy level) GA converges very slowly to the free channels. Before jamming, the channel indices of the whole GA population are concentrated onto a few frequencies instead of being randomly distributed for some runs. Once a good channel is generated, this solution will spread out to the next population with high probability. However, if all channels used at iteration 3 are jammed by our signal generator, chromosomes that correspond to free channels can only be introduced through crossover and mutation operations. The probability for this occurrence is rather low with the settings in Table I. Instead of considering constant GA internal parameters in all scenarios, a more sophisticated solution would consist in adapting the internal parameters based on the current occupancy level. However, we used constant rates in this study in order to keep our GA implementation as generic as we could. In Fig. 5, we illustrate this behavior for two extreme runs as well as the average behavior for several runs for the case when 90% of the channels are jammed. In the first case, the population at iteration 4 contains some chromosomes corresponding to free channels.

Most of the solutions correspond to jammed channels, which caused sharp drop from 0.98 to 0.1 as the fitness score at iteration 4. However, the chromosomes corresponding to free channels will quickly propagate during the next iterations due to their high scores.

In the second case, the population at iteration 3 does not contain any chromosome including free channels. Therefore, the average fitness f_{min_BER} of the population drops to nearly zero at iteration 4. There is no significant improvement until iteration 9. Performance even worsen from iteration 4 to iteration 5. The possibility to generate a chromosome corresponding to a free channel can only occur throughout mutation or crossover operation (and not throughout selection). Up to this occurrence, all packets are lost because their transmission occurs in jammed channels for any chromosome of the GA population. Either mutation or crossover introduced a chromosome corresponding to a free channel at iteration 9. The fitness score improves from 0 in iteration eight to 0.058 after iteration 9. This good solution quickly propagates through the new populations within the next iterations. Ultimately, all solutions that correspond to jammed channels will be removed. Finally, the third curve in Fig. 5 represents the average fitness score over six runs and includes both aforementioned cases.

Globally, GA is suitable for efficient dynamic spectrum access. For some extreme cases such as for 90% of the channels jammed, GA may converge slowly to free channels. In that case, one solution consists in increasing the mutation rate of GA. Introducing an adaptive mutation rate may overcome this problem. For implementation complexity reasons, we only considered a constant mutation rate in this paper. It is clear that the channel allocation against jamming based on GA approach may not be always optimal. Energy detectors or feature-detectors [18] combined with simple avoidance algorithms may perform better. The real benefit in CR context is that the jamming avoidance is combined with GA which is performing also other optimizations at the same time.

V. Conclusions

In this paper we have reported on our CR implementation on GNU radio platform that optimizes the transmission parameters including the transmission power, the modulation scheme and the transmission frequency. Since the optimization problem is nonlinear, we have proposed a GA as general optimization tool for CRs. Unlike the other CR implementations that have been proposed so far, our approach requires only a limited feedback information at the transmitter. No additional NSI nor transmission/propagation model is needed at the transmitter. The experimental results show that our CR is able to find very quickly free channels using GA, even in highly occupied channel scenarios. Based on our experiments and the implementation we believe that GA based optimization for CR, especially when combined with the limited feedback, is a promising approach which warrants further research and development work.

Acknowledgment

The authors would like to thank Matthias Wellens for the fruitful discussions and the insights during the implementation work, and Luis Simões for help on earlier implementation work, especially with the TCP feedback channel. A part of this work is done for the fulfillment of M.Sc. thesis of one of us (CS). This work has been in part financed by DFG, RWTH Aachen University through UMIC, and European Commission (ARAGORN project).

References

[1] J. Mitola III and G. Maguire, Jr., "Cognitive radio:making Software Radios More Personal," *IEEE Personal Communications*, pp. 13–18, Aug. 1999.
[2] C. Rieser, *Biologically inspired cognitive radio engine model utilizing distributed genetic algorithms for secure and robust wireless comunications and networking*. PhD thesis, Virginia Polytechnic Institute and State University, April 2004.
[3] M. Petrova, P. Mähönen, J. Riihijärvi, and M. Wellens, "Cognitive Wireless Networks: Your Network Just Became a Teenager," in *IEEE INFOCOM (Poster session)*, April 2006.
[4] M. Petrova and P. Mähönen, *Cognitive Resource Manager: A cross-layer architecture for implementing Cognitive Radio Networks*. Cognitive Wireless Networks (eds. Fittzek F. and Katz M.), Springer, 2007.
[5] P. Sutton, L. Doyle, and K. Nolan, "A Reconfigurable Platform for Cognitive Networks," in *Proc. of the Int. Conf. on Cognitive Radio Oriented Wireless Networks and Communications*, pp. 1–5, June 2006.
[6] T. Newman, B. Barker, A. Wyglinski, A. Agah, J. Evans, and G. Minden, *Cognitive Engine Implementation for Wireless Multicarrier Transceivers*. Wiley Wireless Communications and Mobile Computing ed., May 2007.
[7] GNU Radio Project Site, "The GNU Software radio," Latest update June 9, 2007. Available: www.gnu.org/software/gnuradio.
[8] C. Fonseca and P. Fleming, "Multi-objective optimization and multiple constraint handling with evolutionary algorithms. I. A unified formulation," *IEEE Transactions on Systems, Man and Cybernetics, Part A*, vol. 28, pp. 26–37, Jan. 1998.
[9] T. W. Rondeau, C. J. Rieser, B. Le, and C. W. Bostian, "Cognitive Radios with Genetic Algorithms: Intelligent Control of Software Defined Radios," *SDR Forum Technical Conference*, vol. C, pp. 3–8, 2004.
[10] D. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, 1989.
[11] K. Kärkkäinen, "Phase optimized PN code sets for numerical analysis and simulation of DS-CDMA systems," cited on September 3, 2007. Available: www.ee.oulu.fi/∼kk/optim_codes_info.html.
[12] J. G. Proakis, *Digital Communications*. McGraw-Hill, Fourth ed., 2000.
[13] E. Hagemann, "The Costas Loop- Setting the Loop," 2001. Available: http://www.i.cmpnet.com/chipcenter/dsp/images/dspsourced/DSP010531F1.pdf.
[14] C. J. Rieser, T. W. Rondeau, C. W. Bostian, and T. Gallagher, "Cognitive Radios Testbed: Further details and testing of a distributed genetic algorithm based cohnitive engine for programmable radios," in *Proc. of the IEEE Military Communications Conference*, vol. 3, pp. 1437–1443, Nov. 2004.
[15] K. Mueller and M. Müller, "Timing Recovery in Digital Synchronous Data Receivers," *IEEE Transactions on Communications*, vol. 24, pp. 516–531, May 1976.
[16] C. Fonseca and P. Fleming, "Multi-objective optimization and multiple constraint handling with evolutionary algorithms. II. Application example," *IEEE Transactions on Systems, Man and Cybernetics, Part A*, vol. 28, pp. 38–47, Jan. 1998.
[17] IEEE 802.11, "IEEE 802.11-1999," tech. rep., Institute of Electrical and Electronics Engineers, 1999. Available: http://standards.ieee.org/getieee802/802.11.html.
[18] I.F. Akyildiz and W.-Y. Lee and M.C. Vuran and S. Mohanty, "Next generation/dynamic spectrum access/cognitive radio wireless networks: a survey," *Computer Networks*, vol. 50, no. 13, pp. 2127–2159, 2006.