

# OPTIMIZATION OF QoS PARAMETERS IN COGNITIVE RADIO USING ADAPTIVE GENETIC ALGORITHM

Maninder Jeet Kaur<sup>1</sup>, Moin Uddin<sup>2</sup> and Harsh K Verma<sup>3</sup>

<sup>1,3</sup>Department of Computer Engineering, Dr B R Ambedkar National Institute of  
Technology, Jalandhar, India

mani356@gmail.com , vermah@nitj.ac.in

<sup>2</sup>Delhi Technological University, New Delhi, India

prof\_moin@yahoo.com

## **ABSTRACT**

*Genetic algorithm based optimization rely on explicit relationships between parameters, observations and criteria. GA based optimization when done in cognitive radio can provide a criteria to accommodate the secondary users in best possible space in the spectrum by interacting with the dynamic radio environment at real time. In this paper we have proposed adaptive genetic algorithm with adapting crossover and mutation parameters for the reasoning engine in cognitive radio to obtain the optimum radio configurations. This method ensure better controlling of the algorithm parameters and hence the increasing the performance. The main advantage of genetic algorithm over other soft computing techniques is its multi – objective handling capability. We focus on spectrum management with a hypothesis that inputs are provided by either sensing information from the radio environment or the secondary user. Also the QoS requirements condition is also specified in the hypothesis. The cognitive radio will sense the radio frequency parameter from the environment and the reasoning engine in the cognitive radio will take the required decisions in order to provide new spectrum allocation as demanded by the user. The transmission parameters which can be taken into consideration are modulation method, bandwidth, data rate, symbol rate, power consumption etc. We simulated cognitive radio engine which is driven by genetic algorithm to determine the optimal set of radio transmission parameters. We have fitness objectives to guide one system to an optimal state. These objectives are combined to one multi – objective fitness function using weighted sum approach so that each objective can be represented by a rank which represents the importance of each objective. We have transmission parameters as decision variables and environmental parameters are used as inputs to the objective function. We have compared the proposed adaptive genetic algorithm (AGA) with conventional genetic algorithm (CGA) with same set of conditions. MATLAB simulations were used to analyze the scenarios.*

## **KEYWORDS**

*Cognitive Radio, Genetic Algorithm, Optimization*

## **1. INTRODUCTION**

Because of the growth of the number of services, the spectrum which is a limited natural resource is becoming scarce. Various studies and measurements show that the spectrum is not being

utilized efficiently at any given time and space [1]. It has been noticed that the peak traffic of each system occurs during specific times and within specific locations. There have been many studies on the challenges associated with the realization of cognitive radio, some of these are (1) efficient use of the available spectrum (spectrum holes) (2) Spectrum Mobility (3) Energy efficiency (4) Spectrum Decision (5) Accurate Sensing and (6) QoS provisioning. The support of quality of service in optimization of the cognitive radio scenario is also a challenging problem.

The key question is “*How the secondary user would find (in the fastest way) about the free slot of the available spectrum and make necessary decisions to occupy it and at the same time guarantee the QoS of the primary as well as secondary users*”. The answer to this question depends on various factors like the degree of cooperation between the primary and secondary users, the reliability of the spectrum sensing and sharing scheme, the capabilities of the secondary and primary networks etc. This paper focuses on the QoS provisioning of the cognitive radio. Also the integration of real time or practical learning capabilities is also a challenge in cognitive radios. Investigation of applications of lesser-known soft computing techniques can result in several benefits as it enables making decisions through predictions and also when accurate real time information is not available.

In this work, we focus on two methods. Firstly the **Spectrum Sensing**, in which the reliability of the spectrum sensing technique using fuzzy logic is studied and analyzed for QoS provisioning. Secondly, **Spectrum Management**, to control the QoS of the cognitive radio system by optimization of the system using genetic algorithm is studied and evaluated. We can combine the two methods to collectively address it as **Spectrum Optimization**.

## 2. GENETIC ALGORITHM

The concept of genetic algorithm lies on the fact that the measure of success of an individual is its fitness i.e. “survival of fittest”. It states that the best combination of genes and their resulting chromosomes yields the strongest individual, which will survive the longest. The first step of GA requires the solutions to be encoded into chromosomes. In case of cognitive radio scenario, the configurable radio parameters like transmit power, modulation, coding rate, symbol rate, packet size etc. represent genes of a chromosome. We have taken such parameters for the chromosome construction; transmit power (P), bandwidth (BW) and symbol rate (Rs), modulation, bit rate (Rb) and one environmental parameter as noise power (N). These six genes will define the structure of the chromosome. We have considered the values of the parameters as discussed [2], for the implementation of cognitive radio as per IEEE 802.22 standard.

The structure of chromosome comprises of the above-mentioned six parameters or genes. The parameters or genes are concatenated to form a string (chromosome). Table 1 gives the summarized values of the order of the genes, the ranges of operation and the binary bits required to encode the corresponding integer values.

Table 1 Summarized Values of the Chromosome Structure

Order	1	2	3	4	5	6
Gene	Power	Bandwidth	Symbol Rate	Bit Rate	Modulation	Noise Power
Ranges	-30dBm to 30dBm	40 to 840 MHz	1Mbps to 8Mbps	0 to 2 Mbps	-	-20 dBm - -15 dBm
Step Size	1 dBm	8 MHz	1 Mbps	125 Kbps	-	1 dBm
Integer Value Range	0 – 59	0 – 99	0 – 7	0 – 15	0 – 3	0 – 5
Binary Bits Required	6	7	3	4	2	3

The six parameters specified as genes in chromosome structure need 25 bits for its construction. The composition of this bit string is important because the mutation operation performs at bit level. Let us assume the parameters be  $x_1, x_2, x_3, x_4, x_5$  and  $x_6$  corresponding to the power, bandwidth, symbol rate, bit rate, modulation and noise power respectively. Fitness functions for each parameter are generated by the following formula initiated in [3] is given as:

$$f_i = \left[ \frac{w_i \cdot |x_i - x_i^d|}{x_i^d} \right] \text{ if } |x_i - x_i^d| < x_i^d \quad \dots (1)$$

where  $x_i^d$  is the required QoS parameter.  $w_i$  is the weight subject to  $\sum_{i=1}^6 w_i = 1$ , where  $i=1, 2, \dots, 6$ . Overall fitness function  $f_{total}$  is the cumulative sum of individual fitness functions of the parameters given as:

$$f_{total} = \sum_{i=1}^6 f_i \quad \dots (2)$$

The overall fitness value of the chromosomes in percentage is

$$\text{Total fitness value (\%)} = 100 \left[ 1 - \sum_{i=1}^6 f_i \right] \quad \dots (3)$$

Ideally each  $w_i$  should be 16.7 %, that means each gene will have the same weight. But in practical scenario, the weighing factor  $w$  can vary according to QoS specifications. *Selection* - We compare the total fitness of the chromosomes using equation 3 and the best solutions from pool of chromosomes are selected using ‘Roulette wheel selection’. It states that the probability of selection for each individual (chromosome) is given by:

$$p_i = \frac{f_i}{\sum_{i=1}^n f_i} \quad \dots (4)$$

$p_i$  is the probability of selection of individual chromosome.  $f_i$  is the fitness of the individual gene  $i$ .  $n$  is the number of chromosome in the population. The individuals (chromosomes) having the maximum probability value are transferred to the next generation. Subsequently, with each generation the fittest chromosomes are selected and transferred to next generation and made new population until the termination criteria is reached

Table 2 Performance Objective Functions and Objective Functions

S.No.	Objective Function	Equation	Remarks
1	Power Consumption( $P_c$ )	$P_c = PR_s K$ [4]	P=Transmit Power (dBm); R <sub>s</sub> = Symbol Rate(kbit/s); K=Signal Space dimensions(e.g. For PSK, K=1, for QAM, K=2 );
2	Spectral Efficiency( $\eta_B$ )	$\eta_B = \frac{R_b}{BW}$	R <sub>b</sub> =bit rate (kbit/s); BW= Bandwidth;
3	Throughput (T)	$T = \frac{M_i}{Ni.M_{max}}$	M <sub>i</sub> =Modulation index on channel i; Ni = Number of channels; M <sub>max</sub> =Max Modulation Index
4	Signal to Noise Interference Ratio (SINR)	$SINR = \frac{P}{N + \sum_i I_i}$ , where N = kTBW	P=Transmit Power(dBm); I = Interference ; k = Boltzmann constant = 1.38065x10 <sup>-23</sup> J/K; T=Temperature; BW =Bandwidth.

Table 2 shows the performance objectives to be achieved to lead a system to an optimal state. We have specifies power consumption which has to be minimized, spectral efficiency which has to be maximized, throughput has to be maximized and signal to noise ratio to be maximized.

### 3. CONVENTIONAL GENETIC ALGORITHM

The QoS requirements of different parameters are evaluated for different channel bandwidths. This algorithm will have the flexibility to allow tradeoff between power consumption and spectral efficiency according to the specific QoS requirements. The algorithm is tested over a number of different initial GA settings and different adjustable parameters to compare numerical optimization performance.

*Elitism* – In this dissertation, we have used one more method of selection called elitism. With the process of elitism, some of the best chromosomes are selected from the current population and are transferred directly to the next generation before the crossover and mutation operations. We have set elitism rate of 2%. *Crossover* – Two chromosomes are selected from the current generation of population, arbitrarily and multi – point crossover operation is applied giving new pair of

chromosomes. We have taken crossover rate of 90 %. This means that 90 out of 100 chromosomes will undergo crossover operation. *Mutation* – In this operation some randomly selected bits are made to flip and fit into any of any of the six genes of the chromosome. This operation will provide new solution space to the algorithm. The mutation rate we have taken is 5%. This rate can be increased to increase the fitness of the chromosomes. *Termination criteria* – The steps above selection, crossover and mutation will repeat until termination criteria (best solution) is reached. We have used two termination criteria. Firstly, the number of generations is set to 1000 generations; secondly the convergence of population is set at .95. This means that as the generation progresses and reaches 95% (means 95% of the chromosomes share the same gene), the iterations are stopped. Here the average fitness approaches the best individual and the best solution is selected.

#### *Drawback of conventional genetic algorithm*

While applying genetic algorithm to solve optimization problem, we come across a phenomenon is called **premature convergence** [5]. When premature convergence occurs, the algorithm is converged too fast to a solution, which causes it to be trapped in local minima. The main cause of premature convergence is the lack of diversity because of the ineffectiveness of crossover and mutation operators. Because of premature convergence, the evolutionary algorithm cannot progress and some important genetic solutions are lost. To improve the performance of genetic algorithm we address this issue in our next section. We propose adaptive genetic algorithm with adapting operators and parameters. This is done by controlling the operators and parameters in such a way that they will change the values if the population is not producing enough fit individuals. This in turn will affect diversity and hence the performance of the algorithm will increase. Moreover, it will increase the adaptability of the cognitive radio that is its one of the main characteristic. The performance of the genetic algorithm is calculated by the average fitness convergence values. The higher the fitness score, the fitter the individuals in that generation and therefore the best solution can be achieved.

Table 3 Pseudo Code for a Conventional Genetic Algorithm

<p><b>Inputs</b></p> <ul style="list-style-type: none"> <li>- Size of population</li> <li>- Rate of mutation</li> <li>- Number of Generations    <b>Output</b> X – Solution</li> <li>- Rate of crossover</li> </ul> <p><b>Begin:</b></p> <p>// Initialization</p> <ol style="list-style-type: none"> <li>1. Sense the environment for inputs;</li> <li>2. Convert the inputs into chromosomes by encoding them;</li> <li>3. Generate feasible random solutions;</li> <li>4. Save them in population <i>Pop</i>;</li> </ol> <p>// Loop until the termination criteria is reached</p> <ol style="list-style-type: none"> <li>5. for <math>i = 1</math> to do;</li> </ol> <p>// Selection Based on elitism and fitness function</p> <ol style="list-style-type: none"> <li>6. <math>ne = .</math> ;</li> <li>7. Selection of best <math>ne</math> solutions in <i>Pop</i> and save them in <i>Pop<sub>1</sub></i>;</li> </ol> <p>//Crossover</p> <ol style="list-style-type: none"> <li>8. crossover <math>nc = ( -ne).</math> ;</li> <li>9. for <math>j = 1</math> to <math>nc</math> do;</li> <li>10. Select two solutions randomly from <i>Pop</i>, <math>X_A</math> and <math>X_B</math>;</li> <li>11. Generate <math>X_C</math> and <math>X_D</math> by two point crossover to <math>X_A</math> and <math>X_B</math>;</li> <li>12. Save <math>X_C</math> and <math>X_D</math> to <i>Pop<sub>2</sub></i>;</li> <li>13. end for</li> </ol> <p>//Mutation</p> <ol style="list-style-type: none"> <li>14. for <math>j = 1</math> to <math>nc</math> do;</li> <li>15. Select a solution <math>X_j</math> from <i>Pop<sub>2</sub></i>;</li> <li>16. Mutate each bit of <math>X_j</math> under the rate and generate a new solution <math>X_j'</math>;</li> <li>17. If <math>X_j'</math> is unfeasible ;</li> <li>18. Update <math>X_j'</math> with a new feasible solution by repairing <math>X_j'</math>;</li> <li>19. end if</li> <li>20. update <math>X_j'</math> with <math>X_j'</math> in <i>Pop<sub>2</sub></i>;</li> <li>21. end for</li> </ol> <p>// Updation of Population</p> <ol style="list-style-type: none"> <li>22. update <math>Pop = Pop_1 + Pop_2</math>;</li> <li>23. end for</li> </ol> <p>// Best Solution</p> <ol style="list-style-type: none"> <li>24. Return the best solution <math>X</math> in <i>Pop</i>;</li> </ol>
--

#### 4. PROPOSED ADAPTIVE GENETIC ALGORITHM

The crossover and mutation operations are very important in the genetic algorithm for optimization in cognitive radio. Population adaptation was proposed by Newman *et al.* [3] in which the results from previous evolutions were utilized by seeding percentage of the initial generation with chromosomes from final generation of the last cognition cycle. This method improved the convergence rate of the algorithm to quite an extent. Subsequently, in [6] parameter adaptation in multicarrier environment was proposed by cognitive radio scenario. It used binary ant colony optimization method to optimize the cognitive radio parameters given a set of objectives. The automation of the parameter and operator settings in genetic algorithm for optimization in cognitive radio still remains an unexplored area so far. We have focused our research on this issue. Therefore, we propose a generic scheme for adaptive crossover and mutation rates.

In conventional genetic algorithm, the genetic operators such as crossover and mutation work with a constant probability. However if we have different crossover and mutation rates, the algorithm traverses different search directions in the search space thus improving the performance. The essential design criterion of a cognitive radio is to adapt itself to the surrounding changing environment. Due to this dynamic changing scenario, the proposed method depends on maintaining an acceptable level of productivity throughout the process of evolution and thus it is essential that the designed algorithm adapt itself to the appropriate crossover and mutation rates, and thus optimizing out cognitive radio scenario.

Let the probability of crossover be  $p_c$  and probability of mutation be  $p_m$ . We use the progress value concept as in [7] for calculating the progress (adjustment rates) for the probabilities. Let  $X_A$  and  $X_B$  be two offsprings after crossover operation. Let  $F_{Sum_S}$  be the fitness sum of two offsprings and  $F_{Sum_P}$  denotes the sum of the parent individuals.

$$Value\ CP = F_{Sum_S} + F_{Sum_P} \quad \dots (5)$$

The generation undergoes  $n_c$  crossover operations. Average crossover progress is given by  $\overline{CP}$ :

$$\overline{CP} = \frac{1}{n_c} \sum CP \quad \dots (6)$$

$\overline{CP}$  measures the overall performance of crossover operator in a generation. Similarly, the overall performance of mutation operator in a generation is calculated as follows:

$$MP = f_{new} + f_{old} \quad \dots (7)$$

where  $f_{new}$  is the fitness of new offspring and  $f_{old}$  is the fitness of the old offspring after mutation.

$$\overline{MP} = \frac{1}{n_m} \sum MP \quad \dots (8)$$

The crossover and mutation operator performances are calculated each generation and adjusted using the average progress values. The adjusted values of operators are as follows:

$$p_c = p_c + \mu_1 \quad \text{if} \quad \overline{CP} > \overline{MP} \quad \dots (9)$$

$$p_c = p_c - \mu_1 \quad \text{if} \quad \overline{CP} < \overline{MP} \quad \dots (10)$$

$$\text{and} \quad p_m = p_m + \mu_2 \quad \text{if} \quad \overline{CP} > \overline{MP} \quad \dots (11)$$

$$p_m = p_m - \mu_2 \quad \text{if} \quad \overline{CP} < \overline{MP} \quad \dots (12)$$

where  $\mu_1$  and  $\mu_2$  represent the amount of adjustments of  $p_c$  and  $p_m$ . Experimental results in [8] gives the values of the adjustments as:

$$\mu_1 = \mu_2 = 0.01 \frac{f_{max} - f_{avg}}{f_{max} - f_{min}} \quad \text{if} \quad f_{max} > f_{min} \quad \dots (13)$$

$$\mu_1 = \mu_2 = 0.01 \quad \text{if } f_{max} = f_{avg} \quad \dots (14)$$

Table 2 Pseudo Code for Proposed Adaptive Genetic Algorithm

<p><b>Inputs</b></p> <ul style="list-style-type: none"> <li>- Size of Population</li> <li>- Rate of Mutation</li> <li>- Number of Iterations</li> <li>- Rate of Crossover</li> </ul> <p><b>Output X – Solutions</b></p> <p><b>Begin</b></p> <p><i>// Initialization</i></p> <ol style="list-style-type: none"> <li>1. Sense the environment for inputs;</li> <li>2. Convert the inputs into chromosomes by encoding them;</li> <li>3. Generate feasible random solution;</li> <li>4. Save them in Pop;</li> </ol> <p><i>// Loop1 until termination criteria is reached</i></p> <ol style="list-style-type: none"> <li>5. for i = 1 to do;</li> </ol> <p><i>// Selection based on fitness function</i></p> <ol style="list-style-type: none"> <li>6. Selection of best individuals in Pop;</li> <li>7. Save them in Pop<sub>1</sub>;</li> </ol> <p><i>// Loop2 Start</i></p> <p><i>// Crossover and Mutation</i></p> <ol style="list-style-type: none"> <li>8. Select two parents from Pop<sub>1</sub>;</li> <li>9. Perform Crossover operation;</li> <li>10. Save the offspring values in Pop<sub>2</sub>;</li> <li>10. Accumulate the crossover progress value CP;</li> <li>11. Perform Mutation Operation;</li> <li>12. Mutate each bit of X<sub>j</sub> in Pop<sub>2</sub> under the rate and generate a new solution X<sub>j</sub>';</li> <li>13. Update the offspring value X<sub>j</sub>' in Pop<sub>2</sub>;</li> <li>14. Accumulate the mutation progress value MP;</li> </ol> <p><i>// Loop2 end</i></p> <p><i>// Progress Rates and Adjustments</i></p> <ol style="list-style-type: none"> <li>15. Calculate <math>\overline{CP}</math>;</li> <li>16. Adjust crossover rate ;</li> <li>17. Calculate <math>\overline{MP}</math>;</li> <li>18. Adjust Mutation rate ;</li> </ol> <p><i>// Updation of Population</i></p> <ol style="list-style-type: none"> <li>19. Update Pop= Pop<sub>1</sub>+Pop<sub>2</sub>;</li> </ol> <p><i>// End Loop1</i></p> <p><i>// Best Solution</i></p> <ol style="list-style-type: none"> <li>20. Return the best solution X in Pop;</li> </ol>
---

## 5. SIMULATION RESULTS

Our simulation is targeted to determine the convergence speed of the fitness function using conventional genetic algorithm. We also analyze the fitness value the system converges to. Weights are defined in Table 6, which are used to create four different search modes for the cognitive radio. Mode 1 is the low power mode (for applications e.g. text messaging), Mode 2 is the high efficiency mode (e.g. ECG data), Mode 3 is the high throughput mode (e.g. broadband video download) and Mode 4 is specifies for high reliability (e.g. defense services).



Table 5 Genetic Algorithm Settings

S. No.	Parameter	Value
1.	Population Size	50
2.	Length of Chromosomes	25 bits
3.	Number of Generations	1000
4.	Crossover Rate	90%
5.	Mutation Rate	5%
6	Elitism Rate	2%

We aimed at targeting the speed of convergence of the fitness function using CGA(Conventional Genetic Algorithm) and AGA (Adaptive Genetic Algorithm). Figure 1 shows the comparison of the fitness convergence for CGA and AGA in lower power mode (Mode 1), Figure 2 shows the comparison of the fitness convergence scenario for CGA and AGA in high efficiency mode (Mode 2). Figure 3 shows the comparison of the fitness convergence scenario for CGA and AGA in high throughput mode (Mode 3). And Figure 4 shows the comparison of fitness convergence scenarios for CGA and AGA in high reliability mode (Mode 4).

Table 6 Weight Values

Objective	w1	w2	w3	w4
Mode 1	0.65	0.15	0.08	0.12
Mode 2	0.12	0.65	0.15	0.08
Mode 3	0.08	0.12	0.65	0.15
Mode 4	0.15	0.08	0.12	0.65

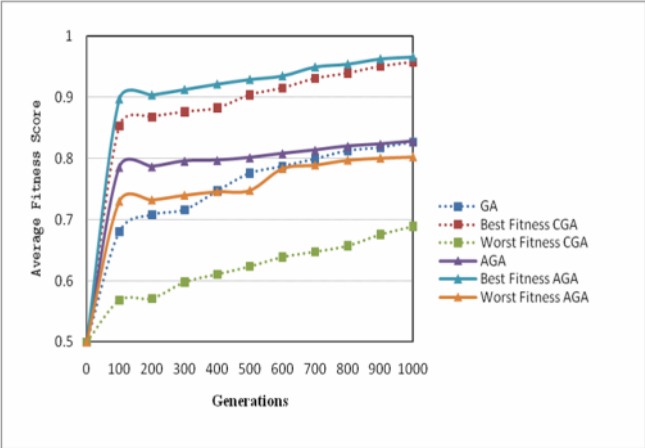


Figure 1 Comparison of Average Fitness Scores of CGA and AGA for Mode 1 (Low Power Mode)

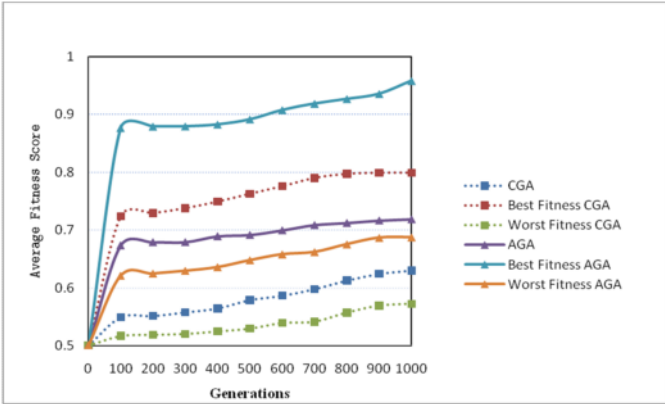


Figure 2 Comparison of Average Fitness Scores of CGA and AGA for Mode 2 (High Efficiency Mode)

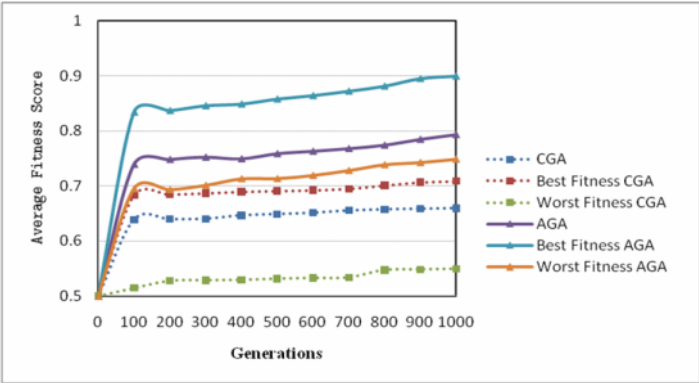


Figure 3 Comparison of Average Fitness Scores of CGA and AGA for Mode 3 (High Throughput Mode)

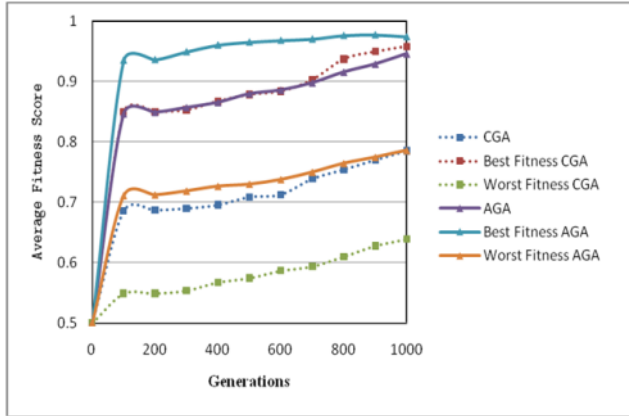


Figure 4 Comparison of Average Fitness Scores of CGA and AGA for Mode 4 (High Reliability Mode)

The results clearly show that the AGA (Adaptive Genetic Algorithm) outperforms the CGA (Conventional Genetic Algorithm) in all the four modes of operation. The fitness convergence values for the two algorithms are compared in Table 7.

Table 7 Comparison of Fitness Convergence Values

Generation Number	Mode 1		Mode 2		Mode 3		Mode 4	
	CGA	AGA	CGA	AGA	CGA	AGA	CGA	AGA
100	.8526	.8967	.7236	.8765	.6833	.8345	.8491	.9345
500	.9033	.9287	.7625	.8914	.6912	.8582	.8782	.9647
1000	.9567	.9654	.7989	.9581	.7087	.8999	.9579	.9737

It was analyzed that the best solution with the fitness value of the gene converging to zero, but in our result the converging point is 0.2016 as shown in Figure 5.

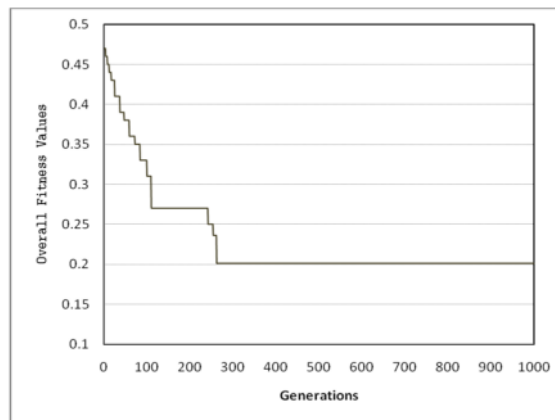


Figure 5 Overall Fitness Value Plot for AGA

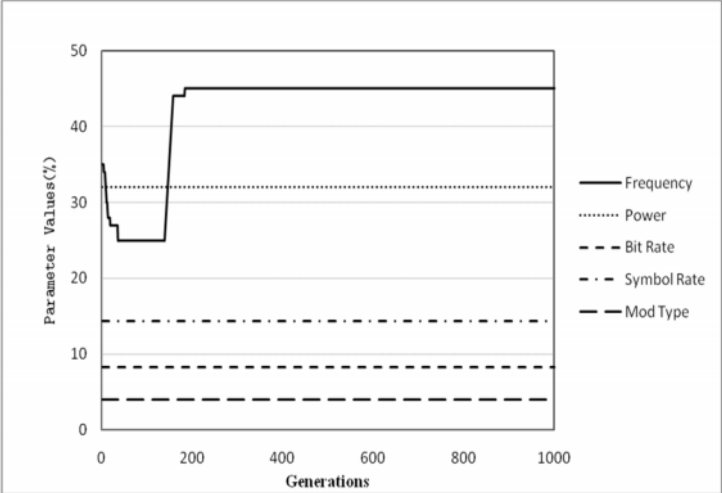


Figure 6 Generation Vs. Parameter Value (%) for AGA

With the proposed algorithm, it is found that the best value of the parameter to obtain the required QoS specification for the cognitive radios system is for the frequency as shown in Figure 6. The percentage of the best total fitness values against the average fitness values for AGA is shown in Figure 7. It shows that the proposed algorithm has 84 % of best fitness values, which is more than the previous algorithms proposed by the researchers.

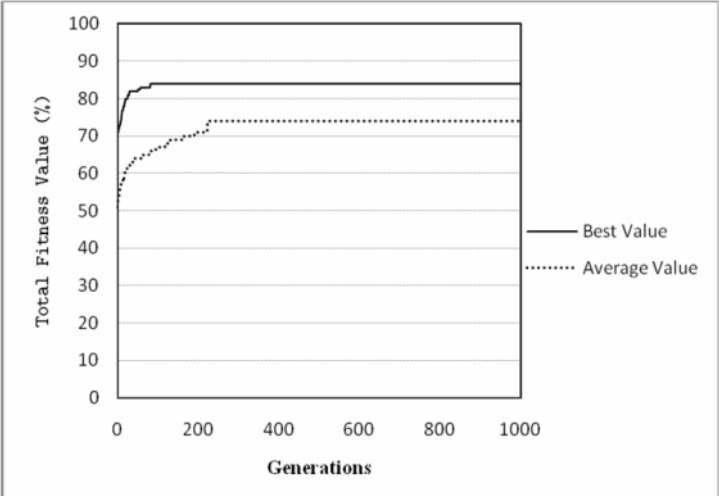


Figure 7 Plot of the Fitness Values in (%) for AGA

Table 8 Comparison for the transmission parameter values after 1000 iterations.

Objective	Average Power (Pt) (dBm)		Bit Rate (Rb) (Mbps)		Modulation index (Mi)	
	CGA	AGA	CGA	AGA	CGA	AGA
Low Power Mode	3.267	2.763	0.687	0.834	4.768	2.76
High Efficiency Mode	10.45	9.76	0.852	0.966	4.891	3.48
High Throughput Mode	12.678	10.456	1.783	1.968	8.873	7.12
High Reliability Mode	5.463	4.113	1.569	1.683	6.873	5.76

Finally, the values of the transmission parameters were calculated and analyzed for both the algorithms as summarized in Table 8. It shows that the goal was achieved in every mode of operation. After 1000 iterations, the cognitive radio was able to decrease the power to an average of 2.763 dBm. For the second goal to increase the efficiency the power had to be increased but the modulation index was reduced. For high throughput mode, the power is increase to increase the bit rate for high throughput transmission purposes. And for the fourth mode to increase the reliability the balanced approach is applied, the power is kept not too high or not too low for ensuring the reliability of the transmission.

## 6. CONCLUSION

In this paper, we developed performance objective fitness functions representing the relationship between the transmission and environmental parameters that were identified in the literature survey. For the implementation to be simple, we have not considered all the parameters, which contribute to the cognitive radios system but a set of parameters that contribute to good extent. The parameters were encoded using binary encoded method into chromosomes. The chromosome of 25-bit length string was used. We first carried out the simulation scenario on conventional genetic algorithm (CGA). The pseudo code was developed for step-by-step implementation of the genetic algorithm implementation. The same set of conditions were tested upon the proposed adaptive genetic algorithm (ADA) in which the crossover and mutation rates were automated. This automation was based upon the previous fitness values and previous crossover and mutation rates. The *progress\_value* function was initialized, which measures the overall performance of the crossover and mutation operators in a generation. The required adjustments are made for the crossover and mutation rates that are decided by the *progress\_value* function. The results of the proposed algorithm show that the optimal solutions give the values for individual genes that accommodate the secondary user with a specific QoS requirement. But the tradeoff between the parameters has to be managed as the behavior is not linear. The simulation results shows that the proposed method gives better solution to the cognitive system with adapting capabilities and

converges faster than the conventional methods of optimization of cognitive radio parameters. The QoS parameters optimized by the proposed algorithm have higher fitness values and also have the capability to manage the tradeoff between multiple objectives more effectively. The convergence plots illustrated the expected trends but the best convergence fitness value reaches stability at 95% in case of the proposed adaptive genetic algorithm. The problem of premature convergence was also solved as the proposed adaptive genetic algorithm adjusts to the suitable crossover and mutation rates to reduce the effort of searching for appropriate crossover and mutation rates.

## REFERENCES

- [1] Federal Communications Commission - First Report, and Order and Further Notice of Proposed Rulemaking, "Unlicensed operation in the TV broadcast bands," *FCC 06-156*, Oct. 2006.
- [2] V. Blaschke, T. Renk, F. K. Jondral, "A Cognitive Radio Receiver Supporting Wide-Band Sensing," *IEEE Conference on Communications, ICC '08*, pp. 499-503, 2008.
- [3] T. R. Newman, R. Rajbanshi, A. M. Wyglinski, J. B. Evans, and G. J. Minden, "Population adaptation for genetic algorithm-based cognitive radios," in *IEEE Proceedings of Cognitive Radio Oriented Wireless Networks and Communications*, Orlando, FL, August 2007.
- [4] T. W. Rondeau, B. Le, D. Maldonado and C.W. Bostian, "Cognitive Radio formulation and implementation" in *Proceedings of the First International Conference on Cognitive Radio Oriented Wireless Networks and Communications, (CROWNCOM'06)*, 2004.
- [5] Y. Leung, Y. Gao and Z. Hu, "Degree of population diversity – a perspective on premature convergence in genetic algorithm and its markov chain analysis," *IEEE Transactions on Neural Networks*, Vol. 8, pp. 1165 – 1176, 1997.
- [6] M. Waheed and C. Cai, "Cognitive Radio Parameter Adaptation in Multicarrier Environment," *IEEE Fifth International Conference on Wireless and Mobile Communications*, pp. 391 – 395, 2009.
- [7] T. P. Hong, H. S. Wang, W, Y. Lee, "Evolution of appropriate crossover and mutation operators in a genetic process," *Applied Intelligence*, Vol. 16, pp. 7 – 17, 2002.
- [8] W.Y. Lin, W.Y. Lee, T.P. Hong, "Adapting Crossover and Mutation Rates in Genetic Algorithm," *Journal of Information Science and Engineering*, pp. 1- 17, 2003.

## Authors

Maninder Jeet Kaur is working as a Research Scholar in Department of Computer Science Engineering at Dr B R Ambedkar National Institute of Technology, Jalandhar, India. She has completed her B.Tech in Electronics and Communication Engineering from Punjab Technical University in 2005. She completed her M.Tech in Computer Science Engineering from Punjab Agricultural University, Ludhiana, Punjab, India in 2007. She has published and presented many papers in International Journals and Conferences. She is a member of International Association of Engineering (IAENG) and International Association of Engineering and Scientists (IAEST). She was selected for Commonwealth Split Site Doctoral Fellowship-2010 for doing research work at University of York, United Kingdom for a period of 1 year. Her current research interests include Cognitive Radio, Artificial Intelligence, Information Communication etc.



Moin Uddin , presently Pro – Vice Chancellor of Delhi Technological University and Former Director Dr B R Ambedkar National Institute of Technology, Jalandhar (India). He obtained his B.Sc. Engineering and M.Sc. Engineering (Electrical) from AMU, Aligarh in 1972 and 1978 respectively. He obtained his Ph. D degree from University of Roorkee, Roorkee in 1994. Before joining NIT, Jalandhar, he has worked as Head Electrical Engineering Department and Dean Faculty of Engineering and Technology at Jamia Millia Islamia (Central University) New Delhi. He supervised 14 Ph. D thesis and more than 30 M.Tech dissertations. He has published more than 40 research papers in reputed journals and conferences. Prof. Moin Uddin holds membership of many professional bodies. He is a Senior Member of IEEE.



Harsh K. Verma received his PhD degree in Computer Science and Engineering from Punjab Technical University, Jalandhar and Master's degree from Birla Institute of Technology, Pilani. He is presently working as Associate Professor in the Department of Computer Science and Engineering at Dr B R Ambedkar National Institute of Technology, Jalandhar, Punjab, India. He has published more than 20 research papers in various Journals and Conferences of International repute. His teaching and research activities include Scientific Computing, Information Security, Soft Computing and Software Engineering.

