

Performance Assessment of Cognitive Radio Adaptation Engine based on Real-coded Genetic Algorithms

Ramy A. Fathy
Digital Services Planning
Telecom Regulatory Authority
(NTRA), Egypt

AbdelhalimZekry
Electronics and Comm. Dept.
Ain Shams University
(ASU), Egypt

Ahmed A. AbdelHafez
Electronics and Comm. Dept. Military
Technical College
(MTC), Egypt

ABSTRACT

Adaptation is one of the fundamental functionalities of Cognitive Radio Systems (CRS). Adaptation refers to the ability of the radio to adapt its operating parameters in response to varying stimuli. Choice of the best parameter set of the radio to achieve certain objectives in shortest time possible remains one of the most challenging tasks in Cognitive Radio (CR) research. One possible approach to adaptation engine design is based on utilizing Genetic Algorithms (GA) which invoke a combination of exploration and exploitation processes to perform random and directed searches for semi-optimal solutions in the possible solution space. However, conventional Binary-coded Genetic Algorithms (BGA) based adaptation engines used frequently in CR research, are criticized for their slow convergence and response times. Accordingly, Real-coded Genetic Algorithms (RGA) – a specific type of GA – have been implemented in our work, to address this problem. RGA alleviates many of the disadvantages of conventional BGA based implementations. This paper focuses on RGA based adaptation engine implementations' performance assessment compared to conventional BGA based implementations. Performance assessment results indicate that RGA based implementation does demonstrate a superior performance over BGA based implementations; in achieving the best configuration to minimize the link BER with minimum possible transmitted EIRP levels; in the shortest time possible.

General Terms

Cognitive Radio, Meta-heuristics, Wireless Communication.

Keywords

Adaptation Engine, Binary-coded Genetic Algorithms, Real-coded Genetic Algorithms.

1. INTRODUCTION

Cognitive Radio Systems (CRS) provide a very promising solution to the vast challenges facing the satisfaction of the ever increasing users' communications needs of the next era. Driven by higher demands for higher speeds, and ubiquitous service delivery; there is a constant pressure on technologists and innovators for developing new techniques for maximizing the usage of the current resources including natural (e.g. frequency spectrum) and artificial (e.g. computing resources) ones.

CRS acquire cognitive behavior by adopting basic elements of cognition; most importantly of which, is the ability of the radio to adapt its operating parameters in response to varying stimuli [1]. Adaptation entails the choice of the best parameter set of the radio that achieves certain objectives.

An Adaptation Engine (AE) is responsible of conducting the adaptation process in the CR. The AE is considered to be a subset, yet a fundamental part, of the Cognitive Radio Engine (CRE); which is responsible of all cognitive aspects of CR like learning, reasoning, and planning. CRE is identified by other additional blocks in Figure 1 like learning, community build-up through cooperation, and goals and targets setting.

One main driver behind the proposed AE design is to develop means for decision-making capability of CR to satisfy emergency and PPDR applications. Accordingly, focus is extended to ensure the ability of the engine to find a solution to the optimization problem, as fast as possible within a more global view of the system itself with all its interacting elements and components; compared to conventional methods.

A novel design and implementation of a Real-coded Genetic Algorithm (RGA) based Adaptation Engine (AE) has been presented in a prior work [2]. Genetic Algorithms (GA) are optimization methods that mimic natural evolution. Optimization is based on the development of the population comprising a certain number of chromosomes. The chromosomes represent a possible solution set for the optimization problem; which could be maximization or a minimization for a specific objective function.

The population size indicates the number of parallel solutions that would be tried in parallel to reach towards the optimum/sub-optimum solution. The development of the population is regulated by means of two genetic operators; namely, crossover and mutation.

GAs have been widely used in the development of AEs. However, they have been criticized for their slow convergence and response times. Accordingly, Real-coded Genetic Algorithms (RGA) – a specific type of GA – has been implemented, to address this problem. RGA alleviates many of the disadvantages of conventional Binary-coded Genetic Algorithm (BGA) based engines, employed often in the CR adaptation literature [3, 4], like the slow convergence and dynamic response times.

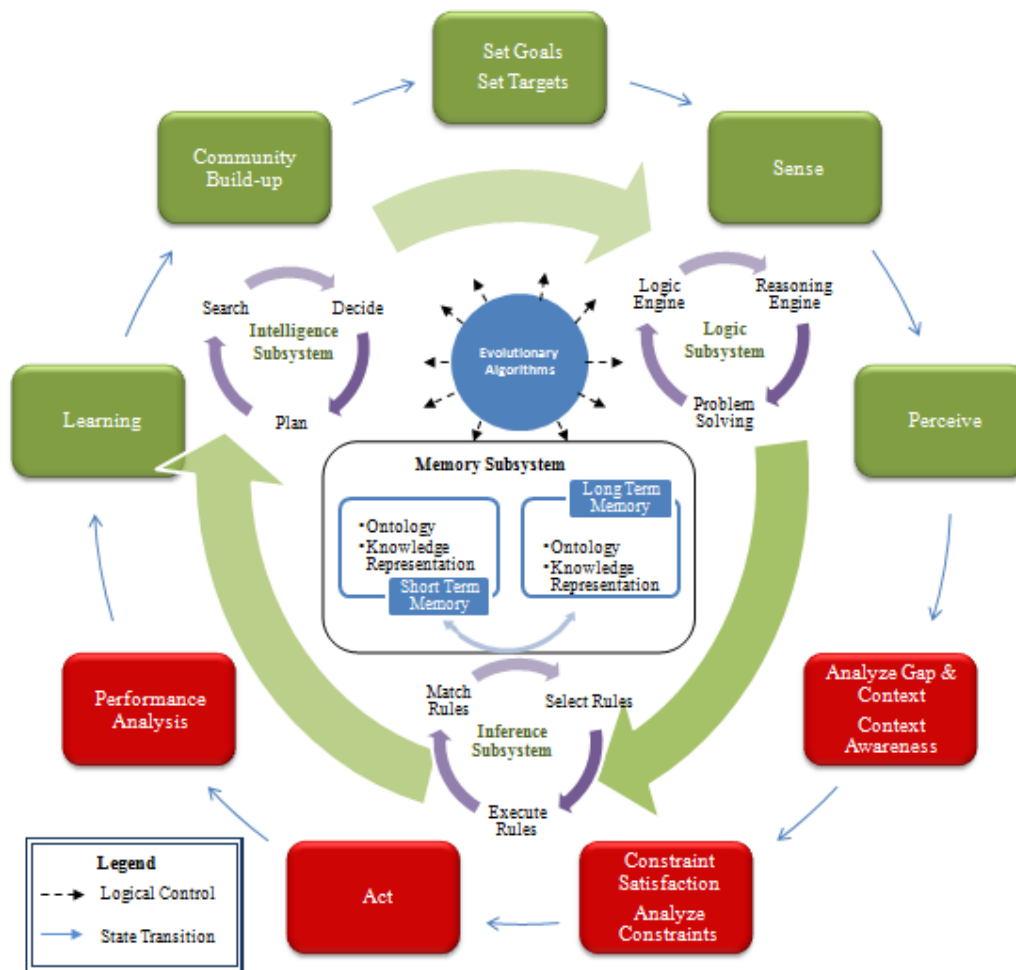


Fig.1 A Conceptual CR Functional Representation based on Formal Theoretical Model of CR
 Red colored blocks demonstrate the functionalities implemented in Adaptation Engine

This paper aims at demonstrating the performance enhancement of RGA based engines over traditional BGA implementations. The rest of this paper is organized as follows:

Section Two presents the objectives of the AE implemented by means of Single Objective Optimization (SOO) setup. Section Three presents the different encoding methods implemented in Genetic Algorithms based AE. Section Four demonstrates the comparative performance assessment conducted for RGA and BGA based AE in addition to the main paper results. Finally, Section Five presents the conclusion and future work that will be addressed in our research, followed by the list of references.

2. Adaptation Engine Objectives

Given is a situation where, a noisy fading channel impairs signals passing through. No equalization techniques are used to compensate for the channel impairments. For a minimization of the BER goal, it is required to transmit the signals through the channel such that the BER is minimized for a particular transmitted power, at a specific data rate (corresponding to a particular service), and at a certain operating frequency.

Based on the channel characteristics, operating frequency, transmitted power, modulation type, and distance of the transmitting node; mathematical formulas for BER and link budget calculations are used to calculate the expected BER for

the established link between the two transceivers. Genetic Algorithms are then used to determine the best combination of the aforementioned variable controlled parameters that could be changed to minimize the BER.

The AE routine starts by generating a number of random possible solutions belonging to the solution space; which includes different combinations of: frequency of operation, the transmitter's EIRP, and the modulation/demodulation techniques that are going to be used in the communication process. These three parameters constitute the engines OPS. A typical situation dictates constraining the engine for not to operate beyond a maximum adjustable operating transmitted EIRP; due to the underlying hardware constraints, in addition to the minimization of the radio terminal power consumption to the maximum extent possible. It is required to optimize the OPS to achieve a minimum link BER.

There are mainly two techniques to achieve the required objective given the power levels constraints. One is to use a Single Objective Optimization (SOO) setting where a fitness function representing the minimization of the link BER target, is constructed. This setting however will only generate solutions the yield minimum BER with no regards of the transmitted EIRP levels. Hence, a special Power Limiting Algorithm (PLA) is needed to drive solutions that achieve a minimum BER with the minimum possible transmitted EIRP levels that satisfies the hardware constraints.

The other possibility is to use a Multi-Objective Optimization (MOO) setting where a fitness function representing the objectives of minimizing both the transmitted EIRP levels and the link BER, is constructed.

2.1 Single Objective Optimization (SOO)

Optimization can be defined mathematically by (1) [5]:

$$\min_{\mathbf{X} \in \mathcal{R}^n} f_i(\mathbf{X}), (i = 1, 2, \dots, M),$$

$$\text{subject to } \phi_j(\mathbf{X}) = 0, (j = 1, 2, \dots, J), \tag{1}$$

$$\psi_k(\mathbf{X}) \leq 0, (k = 1, 2, \dots, K),$$

where $f_i(\mathbf{X})$, $\phi_j(\mathbf{X})$, and $\psi_k(\mathbf{X})$ are functions of the decision variable vector, which is denoted by (2) as follows:

$$\mathbf{X} = (x_1, x_2, \dots, x_n)^T \tag{2}$$

The functions $f_i(\mathbf{X})$ where $i = 1, 2, \dots, M$ are called the objective functions, for the case where $M = 1$, this denotes SOO case. The space spanned by the decision variables is called the search space \mathcal{R}^n , while the space spanned by the objective function values is called the solution space. The equalities for $\phi_j(\mathbf{X})$, and inequalities for $\psi_k(\mathbf{X})$ denotes the constraints for the optimization problem.

In a SOO setup, minimization of the BER is targeted. The solutions are also constrained to develop minimum possible transmitted EIRP levels; which entails the existence of an algorithm that ensures that produced solutions are below a certain maximum allowable transmitted EIRP levels, or else these solutions would not be deemed feasible. Applying equations (1) and (2) for the problem under investigation; we get the following mathematical formulation for our optimization problem as shown in (3), (4), and (5):

$$\min_{\mathbf{X} \in \mathcal{R}^n} f(\mathbf{X}),$$

$$\text{subject to } \phi_j(\mathbf{X}) = 0, (j = 1, 2, \dots, J), \tag{3}$$

$$\psi_k(\mathbf{X}) \leq 0, (k = 1, 2, \dots, K),$$

$$\text{where } \mathbf{X} = (x_1, x_2, x_3)^T;$$

$$x_1 = \text{frequency of operation,} \tag{4}$$

$$x_2 = \text{transmitted EIRP,}$$

$$x_3 = \text{modulation/demodulation;}$$

$$\text{and } f(\mathbf{X}) = B_1 + \left(\frac{Z - A_1}{A_2 - A_1} \right) \times (B_2 - B_1),$$

$$A_1 = 1, A_2 = -\log_{10} \text{Target BER,} \tag{5}$$

$$B_1 = 0, B_2 = 10,$$

$$Z = -\log_{10} \text{BER}$$

$$\text{Target BER} = 10^{-6}$$

The equalities for $\phi_j(\mathbf{X})$, and inequalities for $\psi_k(\mathbf{X})$ are defined dynamically at the engine operation. The fitness function $f(\mathbf{X})$, is the linear translation for the logarithmic function of the BER. This scaling is conducted to simplify the optimization procedure and achieve better solutions convergence results, since working on the actual values of the BER, is impractical due to its logarithmic nature.

The PLA is an algorithm developed specifically to cater for the necessity of generating solutions with minimum transmitted EIRP levels. The objective function however is designed to minimize the resulting BER as shown (3), (4), and (5). PLA is developed to use the lowest possible transmitted EIRP while only a single objective is targeted. Possibly, the transmitted EIRP levels constraint could be fed from a *Policy Engine*, another module belonging to the CRE, which is

responsible of identifying among others, the maximum allowable power levels in every band in accordance to regulatory constraints.

3. Encoding in Genetic Algorithm

GAs operate by invoking a combination of exploration and exploitation processes to perform random and directed searches for semi-optimal solutions in the possible solution space. These techniques have the advantage of avoiding being trapped in non-globally optimal apparently optimal (local optimal in that case) solution sets by achieving the required diversity of the search in addition to necessary search concentration in promising areas of the solution space [6].

According to [7]; exploration refers to the creation of population diversity by exploring new areas of the search space, while exploitation is the reduction of the diversity by focusing on the individuals of higher fitness, or exploiting the fitness information to the maximum represented within the population itself.

Exploration is conducted by means of reproduction of an encoded form of the solutions to a given optimization problem, while exploitation is conducted by means of selection. The encoding process transforms the values of the optimization variables, called ‘genes’, into an encoded form, called a ‘chromosome’. Binary encoding employed in traditional BGA encodes the set of genes into a binary string for further processing. On the other hand, RGA employs real encoding which transforms the set of genes into a real array of variables holding the values of the genes.

Both types of GA start by generating a random population of probable solutions, ‘chromosomes’, where the fitness of each generated chromosome is then determined by evaluating it in a fitness function. Best chromosomes, called ‘parents’, are then selected; where genetic information are being exchanged at a certain probability to produce ‘offspring’ through what is called the crossover operation.

To avoid being trapped in local minima; mutation, a rather rare phenomena is introduced to the chromosome at a specific very small probability to arbitrary change some of the genetic information yielded in the offspring resulting in a probable new form of solutions that isn't a mere exchange of genetic information of the parents.

The offspring solutions are then evaluated in the same fitness function, where superior offspring values replace less fitted individuals in the population through a simulated natural process of generations' evolution. The process is repeated for a large number of generations to obtain a sub-optimum solution.

Generating a large number of individuals in the population enables the algorithms from generating a large number of probable initial solutions to check different areas in the solutions' space in parallel. On the other hand; using large number of bits in the encoding process of BGA, or increasing the precision of the generated solutions in RGA, enhances the resolution of the search which leads to more accurate solutions. Four main parameters have a strong affect on the performance of GAs; namely, population size, number of generations, crossover and mutation rates. Larger population size and large number of generations increase the probability of obtaining a global optimum solution. However this comes at the cost of dramatically increasing the solutions convergence time.

3.1 Binary-coded Genetic Algorithm (BGA)

Encoding solutions of an optimization problem into a chromosome is very important in GA. Binary encoding for function optimization is known to have some drawbacks due to the existence of Hamming cliffs, which denote pairs of encodings having a large Hamming distance while belonging to points of minimal distance in phenotype space [8].

The crossover and mutation genetic operators are relatively simple. However, a key to the performance of the binary encoding operation of the algorithm is the binary to decimal conversion procedure for substituting the values corresponding to the binary-coded chromosomes in the utility function to determine the fitness of the solutions.

The binary to decimal conversion procedures are done for every chromosome in the population; which might cost time and computations. In addition, resulting offspring might not belong to the set of feasible solutions. That necessitates the use of a validity check module – implemented by means of a Power Validation Unit – that repeats the crossover and mutation operation in case the generated offspring doesn't belong to the set of feasible solutions to the optimization problem. This entails additional overhead and affects also the speed of the algorithm. The pseudocode for the structure of BGA procedure is shown below:

Pseudo code of BGA procedure

```

Begin;
Generate real valued random population of P
solutions(chromosomes);
Decimal to Binary conversion of real valued chromosome to
binary valued chromosome;
For each individual  $i \in P$ : calculate  $fitness(i)$ ;
For  $i = 1$  to number of generations;
  Randomly select an operation (crossover or mutation);
  If crossover;
    Select two parents at random  $i_a$  and  $i_b$ ;
    Generate on offspring  $i_c = crossover(i_a, i_b)$ ;
    Validity check if the generated offspring solution
    belongs to the set of feasible solutions;
  Else If mutation;
    Select one chromosome  $i$  at random;
    Generate an offspring  $i_c = mutate(i)$ ;
  End if;
  Binary to Decimal conversion of binary valued
  chromosome to back to real valued chromosome;
  Calculate the fitness of the offspring  $i_c$ ;
  If  $i_c$  is better than the less fit chromosome then
  replace the less fit chromosome by  $i_c$ ;
Next  $i$ ;
Check if termination= true;
End;
```

3.2 Real-coded Genetic Algorithm (RGA)

In order to reduce the amount of computations and the dynamic response time of the engine; the binary to decimal or decimal to binary conversions have been eliminated, by employing RGA. Chromosomes are not expressed as binary strings anymore. They are directly initialized as real values, and special techniques have been applied for conducting crossover and mutation operators using real-coded chromosomes.

The engine is constrained not to operate beyond a maximum adjustable operating transmitted EIRP, so as to minimize the power consumption of the radio terminal to the maximum

extent possible. A special *Power Limiting Algorithm* (PLA) is developed to use the lowest possible transmitted EIRP while only a single goal is targeted; that's minimizing the BER. A possible power level constraint could be fed from a *Policy Engine*, a module which belongs to the CRE; responsible of identifying among others, the maximum allowable power levels in every band in accordance to regulatory constraints. The pseudocode for the structure RGA procedure is shown below:

Pseudo code of RGA procedure

```

Begin;
Generate real valued random population of P
solutions(chromosomes);
For each individual  $i \in P$ : calculate  $fitness(i)$ ;
For  $i = 1$  to number of generations;
  Randomly select an operation (crossover or mutation);
  If crossover;
    Select two parents at random  $i_a$  and  $i_b$ ;
    Generate on offspring  $i_c = crossover(i_a, i_b)$ ;
  Else If mutation;
    Select one chromosome  $i$  at random;
    Generate an offspring  $i_c = mutate(i)$ ;
  End if;
  Calculate the fitness of the offspring  $i_c$ ;
  If  $i_c$  is better than the less fit chromosome then
  replace the less fit chromosome by  $i_c$ ;
Next  $i$ ;
Apply Power Limiting Algorithm;
Check if termination= true;
End;
```

The pseudocode of the structure *Power Limiting Algorithm* is shown below:

Pseudo Code of Power Limiting Algorithm

```

Begin;
Input: Value of Converging Threshold, Power Step, and
Max_EIRP

If current Generation of the population > Convergence
Threshold;
  If error between the resulting BER corresponding to the so
  far optimized OPS and the targeted BER is  $\geq$  permissible
  error;
    Initialize the Generation of the population;
    Increase the maximum allowable EIRP (Max_EIRP) by
    the Power_step;
    Generating random EIRP levels between the minimum
    EIRP preset levels and the new maximum allowable
    EIRP levels;
    Creating a new initial population by replacing the old
    randomly generated EIRP set with the modified one;
    Restart the whole optimization process;
  End if;
End if;
End;
```

4. Comparative Performance Assessment

The AE is implemented and tested using MATLAB version 7.0.0.19920 (R14). The experimentation and assessment procedures are conducted for a total of 6000 runs to ensure statistical confidence, especially since GA is of a stochastic nature in its core building blocks, e.g. selection, and genetic operators.

Figure 2 presents six Test Cases that were conducted in the experimentation procedure to simulate the dynamic behavior of the channel in disaster communication incidents and

emergency communications events. These Test Cases are used as a basis for evaluation [2].

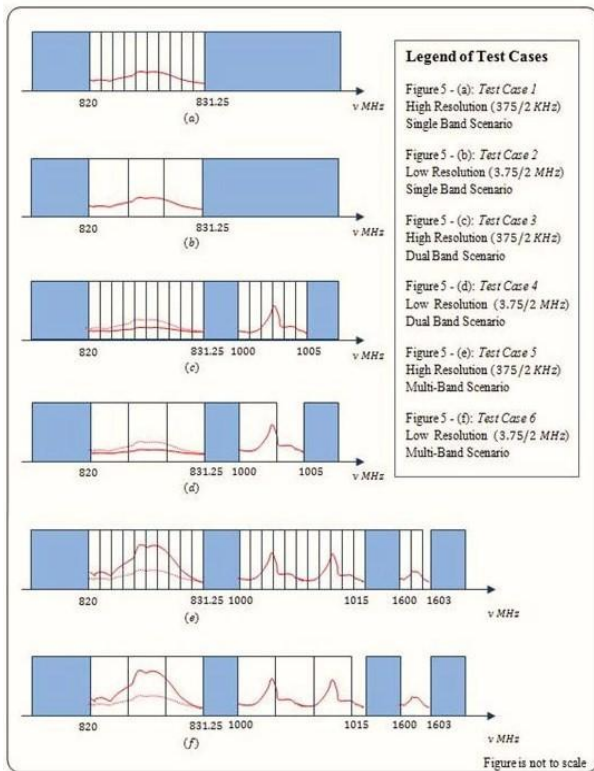


Fig. 2. Test Cases Schematic Diagram for the AE Accuracy Validating [2]

Comparative performance assessment refers to the comparative assessment of the reliability and dynamic

response time of a specific engine implementation compared to other implementations, for the sake of demonstrating better performance assessments. This section demonstrates the levels of performance of RGA based AE versus BGA based implementations.

Conventional meta-heuristic algorithms, like BGA was usually used for optimizing the radio internal parameters to satisfy the radio objectives. Repeating the same experimentation procedures conducted to test feasibility and effectiveness of the implemented RGA based AE; a comparative assessment of the operation of the newly developed engine could be undergone.

Table 1 presents the BGA based AE reliability, performance, and the transmitted EIRP levels resulting from running the algorithm when *Power Limiting Algorithm (PLA)* is used. The assessment is conducted for the different Test Cases presented in Figure 3, at an inter-CR terminals distance of 500 m. Table 2 presents the results for the RGA based AE case.

Results from Tables 1, and 2 assist in the comparative analysis between the performance metrics identified by the Engine Dynamic Response Time and the Engine Reliability for BGA and RGA based AE implementations. In addition, the feasibility of the generated transmitted EIRP levels can be compared for the two implementations; to highlight any deficiencies and/or possible remedies and enhancements.

Analysis of the BGA based AE reliability indicates misleading high reliability values of 100% as compared to that reported in RGA based implementations; where the latter resulted in a reliability ranging from 74% (Test Case 4) to 97% (Test Case 1). This denotes the dependence of the reliability of the engine on the engine's operating environment, services to be transmitted and ambient levels of noise and interference.

Table1: BGA Based Reliability and Performance Assessment of the AE (with PLA)

Test Case Number	KPI	Dynamic Response Time (Sec)	Maximum Allowable EIRP (dB)	Transmitted EIRP (dB)	Reliability
# 1	Mean	14.27	10.00	14.82	100%
	Median	14.22	10.00	15.00	
	Std. Dev.	1.92	0.00	0.44	
	Min.	9.58	10.00	13.00	
	Max.	21.61	10.00	15.00	
# 2	Mean	13.78	10.00	30.76	100%
	Median	13.60	10.00	31.00	
	Std. Dev.	2.04	0.00	0.55	
	Min.	10.19	10.00	29.00	
	Max.	18.86	10.00	31.00	
# 3	Mean	10.76	10.00	29.23	100%
	Median	10.76	10.00	30.00	
	Std. Dev.	1.57	0.00	1.81	
	Min.	7.22	10.00	25.00	
	Max.	15.65	10.00	31.00	

<i>Test Case Number</i>	KPI	Dynamic Response Time (Sec)	Maximum Allowable EIRP (dB)	Transmitted EIRP (dB)	Reliability
# 4	Mean	9.81	10.00	30.68	100%
	Median	9.73	10.00	31.00	
	Std. Dev.	1.37	0.00	0.74	
	Min.	6.79	10.00	27.00	
	Max.	13.87	10.00	31.00	
# 5	Mean	13.27	10.00	30.11	100%
	Median	13.21	10.00	30.00	
	Std. Dev.	1.43	0.00	0.96	
	Min.	10.45	10.00	27.00	
	Max.	17.13	10.00	31.00	
# 6	Mean	11.62	10.00	30.50	100%
	Median	11.31	10.00	31.00	
	Std. Dev.	1.88	0.00	0.78	
	Min.	8.60	10.00	27.00	
	Max.	16.90	10.00	31.00	

Table 42: RGA Based Engine Reliability and Performance Assessment of the AE (with PLA)

<i>Test Case Number</i>	KPI	Dynamic Response Time (Sec)	Maximum Allowable EIRP (dB)	Transmitted EIRP (dB)	Reliability
# 1	Mean	0.94	10.03	8.13	97%
	Median	0.92	10.00	8.36	
	Std. Dev.	0.10	0.30	1.38	
	Min.	0.86	10.00	4.51	
	Max.	1.87	13.00	11.48	
# 2	Mean	2.60	21.43	20.13	84%
	Median	2.65	22.00	20.45	
	Std. Dev.	0.34	1.79	1.93	
	Min.	2.03	19.00	17.10	
	Max.	3.48	25.00	24.91	
# 3	Mean	1.14	10.00	7.83	94%
	Median	1.12	10.00	7.99	
	Std. Dev.	0.06	0.00	1.32	
	Min.	1.03	10.00	4.58	
	Max.	1.50	10.00	9.86	
# 4	Mean	2.69	21.07	19.44	74%
	Median	2.66	22.00	18.69	
	Std. Dev.	0.43	2.08	2.02	
	Min.	2.08	19.00	17.14	
	Max.	3.67	25.00	24.86	

<i>Test Case Number</i>	KPI	Dynamic Response Time (Sec)	Maximum Allowable EIRP (dB)	Transmitted EIRP (dB)	Reliability
# 5	Mean	2.61	12.19	9.99	80%
	Median	2.96	13.00	9.66	
	Std. Dev.	1.04	2.13	1.58	
	Min.	1.50	10.00	6.63	
	Max.	4.63	16.00	13.47	
# 6	Mean	4.15	26.77	25.14	81%
	Median	4.03	28.00	24.44	
	Std. Dev.	1.18	1.91	1.95	
	Min.	3.31	25.00	22.39	
	Max.	14.93	31.00	30.35	

The reason for the false reliability figures produced by BGA based implementation could be attributed to the lack of BGA based implementations to the inherent value limiting capabilities of the genetic operators found in the RGA based approaches. This is mainly due the incapability of the crossover genetic operator applied in BGA to limit the resulting offspring solutions to be within some feasible values, as is the case in RGA.

The implemented crossover technique in RGA approach, the AMXO [9, 10] inherently generates offspring solutions that are bounded by the values of their parents. While this approach might have its limitations, the process of mutation alleviate the possible entrapment in local minima by introducing controlled probabilistic mutations that potentially help the engine to explore for new solutions in other ranges of the solution space.

To mitigate the misleading resulting transmitted EIRP levels in BGA based engine implementations, two possibilities could be applied: one is to apply the optimization routine of the BGA in a MOO setting where a minimization of the transmitted EIRP levels is targeted in addition to minimizing transmitted EIRP levels. Additional objectives can also be included in the objective function like maximizing the throughput and/or maximizing the spectral efficiency. Another possibility is to implement a Power Validation Unit inside the engine with the responsibility of checking the feasibility of the resulting EIRP solutions. However, both approaches dictate higher complexity and slower performance since more calculations in either way needs to be conducted.

Using PLA with RGA based implementations has a similar effect of MOO where both BER minimization and EIRP minimization objectives are met. However, in the former case, only a SOO setup is required; thus eliminating much of the complexity associated with implementing the MOO problem, in addition to the problematic issue of selecting the values of the weights associated with the combined fitness function used in MOO problems [11].

5. CONCLUSION AND FUTURE WORK

In this paper, an assessment of the performance of a RGA based AE in CRS that adapts its physical layer parameters to its environment is presented. It is the first time to our knowledge that RGA is used in the adaptation function of

cognitive radio AE. The purpose of this novel implementation is to achieve minimum link BER and minimum transmitted EIRP levels by optimizing specific physical layer radio parameters in a SOO setting.

Similar targets could be achieved by using MOO with conventional BGA based implementations. However, this paper demonstrated that equipping the algorithm with a simple PLA in a SOO has much faster engine response compared to the traditional BGA based approaches; with no need to increase the engine complexity by applying MOO implementations. *Power Limiting Algorithm* (PLA) operates by incrementing the maximum allowable transmitted EIRP levels during the engine's operation, in case if the engine experienced a slow convergence towards the optimal required solution.

Accordingly RGA based implementation does demonstrate a superior performance over BGA based implementations; in achieving the best configuration to minimize the link BER with minimum possible transmitted EIRP levels; in the shortest time possible.

This paper also demonstrated that the only possibility for implementing BGA based AE, is to either apply the engine in a MOO that includes minimizing the generated transmitted EIRP levels, or to use Power Validation Unit that validates the resulting EIRP levels and inhibits non-feasible solutions to be generated from the engine.

Employing RGA based AE, as a core meta-heuristic technique; enables the radio to provide a multitude of applications and services including high data rate and bandwidth hungry applications like video calls and VOD services.

Further investigation is needed to analyze the sensitivity of the engine resulting solutions due to variations in RGA mutation and crossover control parameters in addition to the probability of mutation and crossover. A quantitative assessment is still needed to accurately assess the performance and complexity associated with SOO and MOO based implementations.

While it is apparent from the implemented SOO approach that the engine employing this method indeed succeeded in attaining a similar behavior of MOO setups by achieving

more than one objective. However it is still necessary to quantify the exact engine's dynamic response time and complexity associated with MOO setups for further comparative assessments.

6. REFERENCES

- [1] J. Mitola, III, "Cognitive Radio for Flexible Multimedia Communications", Mobile Multimedia Communications, 1999. (MoMuC '99) 1999 IEEE International Workshop on, pp. 3 –10, 1999.
- [2] R. A. Fathy, A. A. Abdelhafez and A. Zekry, "Meta-Heuristic based Adaptation Engine for Cognitive Radio Systems," International Journal of Computer Applications Vol. 64, No. 18 pp:53-60, February 2013. Published by Foundation of Computer Science, New York, USA.
- [3] T. W. Rondeau, Application of Artificial Intelligence to Wireless Communications. PhD thesis, Virginia Tech, 2007.
- [4] T. R. Newman, B. A. Barker, A. M. Wyglinski, A. Agah, J. B. Evans, and G. J. Minden, "Cognitive Engine Implementation for Wireless Multicarrier Transceivers," Wiley Journal on Wireless Communications and Mobile Computing, vol. 7, no. 9, pp. 1129–1142, 2007.
- [5] X. Yang. Engineering Optimization: An Introduction with Metaheuristic Applications. John Wiley & Sons, Inc. 2010.
- [6] A. H. A. Ahmed, "Studies on Metaheuristics for Continuous Global Optimization Problems," Kyoto University, Kyoto, Japan, Doctor of Informatics. June 2004.
- [7] L.Hansheng, and K.Lishan, "Balance between exploration and exploitation in genetic search," *Wuhan University Journal of Natural Sciences*, Vol. 4, No. 1 PP. 28-32, 1999.
- [8] J. Ludvig, J. Hesser, and R. Manner, Tackling the representation problem by stochastic averaging, in Proceedings of the 7th International Conference on Genetic Algorithms, Morgan Kaufmann Publishers, San Francisco, PP. 196-203, 1996.
- [9] Z. Michalewicz, "Genetic Algorithms, Numerical Optimization and Constraints," Proceedings of the Sixth International Conference on Genetic Algorithms, pp. 151-158, 1995.
- [10] P. Kaelo and M. M. Ali, "Integrated Crossover Rules in Real Coded Genetic Algorithms," European Journal of Operational Research, vol. 176, no. 1, pp. 60-76, Jan. 2007.
- [11] T. R. Newman, B. A. Barker, A. M. Wyglinski, A. Agah, J. B. Evans, and G. J. Minden, "Cognitive engine implementation for wireless multicarrier transceivers," Wiley Journal on Wireless Communications and Mobile Computing, vol. 7, no. 9, pp. 1129–1142, 2007.

Ramy A. Fathy; received the M.Sc. Degree in Electrical Engineering from Ain Shams University, Cairo, Egypt in 2007. Currently he is with the Egyptian National Telecommunication Regulatory Authority (NTRA) responsible responsible of setting the digital services policies in the Egyptian market. Eng. Fathy is a candidate for the Ph.D. degree in Electrical Engineering, Electronics and Communications Dept., Ain Shams University, Cairo, Egypt. His technical experience covers signal processing, video compression, and DSP related applications specifically those of real time systems, and cryptography. He published several papers in specialized conferences and periodicals in addition to one book in Electronics. His current research interests include cognitive radio, AI, and meta-heuristic techniques.

Abdelhalim Zekry; graduated from Cairo University in Egypt in 1969. He was offered the M.Sc. degree in 1973 from the same university. He worked as a scientific coworker at TU Berlin, where he got his PhD at 1981. He worked as an assistant prof. at Ain-Shams University (ASU), Egypt in 1982. From 1998 till 2004, he moved to King Soud University where he became a professor of Electronics. Now, he is a professor of Electronics at the faculty of Engineering, ASU, Egypt. Prof. Zekry published 133 papers in specialized conferences and periodicals in addition to three books in Electronics. Now, he is driving research on electronics for communication especially the implementation of advanced communication standards using DSP platforms

Ahmed Ali AbdelHafez; received the B.S. and M.Sc. in Electrical Engineering from Military Technical College (MTC) in 1990, 1997 respectively, and his Ph.D from School of Information Technology and Engineering (SITE), University of Ottawa, Ottawa, Canada in 2003. Dr. AbdelHafez is the head of the Cryptography Research Center (CRC), Egypt where he is leading many applied researches in communication security field. He is a visiting lecturer in Communication Dept. MTC, and other universities in Egypt. His research interests include wireless networks and data security, mathematical cryptography and provable security.