M.Tech. Thesis

INVESTIGATION ON VARIANT ALGORITHMS OF BIOGEOGRAPHY BASED OPTIMIZATION FOR IMPROVED PERFORMANCE

Submitted in partial fulfillment of the requirements for the degree of Master of Technology in Electronics & Communication Engineering by

Shelja Tayal (1169062)

Under the Supervision of

Dr. Satvir Singh

PUNJAB TECHNICAL UNIVERSITY

Jalandahr-Kapurthala Highway, Jalandhar

SHAHEED BHAGAT SINGH

STATE TECHNICAL CAMPUS

Moga Road (NH-95), Ferozepur-152004

July 2013

CERTIFICATE

I, Shelja Tayal, hereby declare that the work being presented in this thesis on INVESTIGATION ON VARIANT ALGORITHMS OF BIOGEOGRAPHY BASED OPTIMIZATION FOR IMPROVED PERFORMANCE is an authentic record of my own work carried out by me during my course under the supervision of Dr. Satvir Singh. This is submitted to the Department of ECE at Shaheed Bhagat Singh State Technical Campus, Ferozepur (affiliated to Punjab Technical University, Jalandhar) as partial fulfillment of requirements for award of the degree of Master of Technology in Electronics and Communication..

Shelja Tayal (1169062)

To the best of my knowledge, this thesis has not been submitted to Punjab Technical University, Jalandhar or to any other university or institute for award of any other degree or diploma. It is, further, understood that by this certificate the undersigned does not endorse or approve any statement made, opinion expressed or conclusion drawn therein, however, approve the thesis only for the purpose for which it is submitted.

Dr. Satvir Singh [Supervisor]

The thesis Viva-Voce Examination of Shelja Tayal (1169062) is held at Department of ECE, SBS State Technical Campus, Ferozepur on ....

External Examiner
Name: ....

Mr. Sanjeev Dewra
Head, Department of ECE

Thinking is progress. Non-thinking is stagnation of the individual, organisation and the country. Thinking leads to action. Knowledge without action is useless and irrelevant. Knowledge with action, converts adversity into prosperity.

. . . APJ Abdul Kalam

Dedicated to

My Family

Thesis Outcomes

International/National Journal Publications/Submissions

1. S. Tayal, S. Singh and G. Sachdeva, "BBO Algorithm with Graded Emigration for Yagi-Uda Antenna Gain Optimization," in International Journal of Computer Information Systems and Industrial Management Applications. (accepted)
2. S. Singh, S. Tayal, E. Mittal and Shivangna, "Evolutionary Performance of Graded Emigration in BBO for Yagi-Uda Antenna Design Optimization," in Coimbatore Institute of Information Technology International Journals, Pages 134–139, April 2013.
3. S. Singh, Shivangna and S. Tayal, "Analysis of Different Ranges for Wireless Sensor Node Localization using PSO and BBO and its variants," in International Journal of Computer Applications, Volume 63, Pages 31–36, 2013.

International/National Conference Publications

1. S. Singh, S. Tayal and G. Sachdeva, "Evolutionary Performance of BBO and PSO Algorithms for Yagi-Uda Antenna Design Optimization," in proc. of World Congress on Information and Comm. Technologies, Trivandrum, INDIA, Pages 861–865, 30 Oct-2 Nov, 2012 (online at IEEE *Xplore*).
2. S. Singh, E. Mittal and S. Tayal, "Evolutionary Performance Comparison of BBO and PSO Variants for Yagi-Uda Antenna Gain Maximization," in TEQIP Sponsored IEEE National Conference on Contemporary Techniques & Technologies in Electronics at Murthal, Sonipat (Haryana), Pages 178–183, 13-14 March, 2013.

Invited Talk/Tutorial Presentation

1. "Biogeography-Based Optimization" in AICTE-TEQIP Sponsored 2-week FDP on Nature Inspired Computational Intelligence at SBS State Technical Campus, Ferozepur, 6-17 May, 2013.

ACKNOWLEDGEMENTS

Completion of this M-Tech Thesis could be made possible with the support of several people. I would like to express my sincere gratitude to all of them. First of all, I am extremely grateful to my research guide, **Dr. Satvir Singh** Associate Professor, Electronics & Communication Engineering Department, SBS State Technical Campus, Ferozepur, Punjab, India for his valuable guidance and consistent encouragement throughout the research work. This feat is possible with huge support of his unconditional support and his amicable and positive disposition. I am indeed indebted to the time he spared for daily discussions from his extremely tight work schedule. I also want to express my deepest appretiation to him because of the facilities provided by him to carry out my research work.

My sincere thanks to **Dr. T. S. Sidhu**, Director, SBS State Technial Campus, Ferozepur (Punjab) for help and support he provided to me to accomplish this thesis.

I am very grateful to the then Head of Department **Mrs. Rajni** and the present Head of Department **Mr. Sanjeev Dewra** whose support and directions helped me in this thesis.

I would like to thank **Mr. Vishal Sharma**, Assistant Professor, SBS State Technical Campus for encouraging and boosting my confidence level.

The special thank goes to my senior, **Mr. Gagan Sachdeva**. The supervision, encouragement and support that he gives truly help the progression and smoothness of my research work. The big contribution, motivation, cooperation of him during the research period is very great indeed. Besides, this research period makes me realized the value of working together as a team and a new experience in working environment, which challenges us every minute.

I express my gratitude to **Mr. Sureh Kumar**, **Mr. Darshan Singh**, **Mr. Jaswant Singh**, **Mr. Prafful** and **Mr. Lakhwinder Singh**, Lab Staff, SBS State Technical Campus, Ferozepur (Punjab). All of their encouraging guidelines helped me delve into the new venture of swarm intelligence.

I wish to acknowledge the magnificent support I have received from my friends and colleagues **Ms. Etika Mittal**, **Ms. Shivangna Keer**, **Mr. Sumer Singh**, **Ms. Manbir Kaur**, **Ms. Ritika Arora**, **Ms. Ramandeep Kaur**, **Mrs. Gurimandeep Kaur Shergill**, **Ms. Aman Jhand**, **Mrs. Amrita Tyagi** in the form of useful discussion, motivation and appreciation throughout this work.

I also want to thank **Mr. Vivek Garg**, **Mrs. Pallavi Jindal** and **Mr. Ishan Gupta** who encouraged and motivated me to do M-Tech.

Most profound regards to my mother **Mrs. Anchla Tayal** and my father **Sh. Vijay Kumar Tayal**, who confined their needs all for my sake. My heartiest thanks to my younger brothers **Honey and Ankit** for not even touching my laptop and waiting for hours to operate my laptop to access Internet, playing games and watching movies on it. It is these sacrifices and constant blessings that kept me motivated and commited, until I reached this end.

Finally, I must thank GOD for giving me the environment to study, people to help, opportunities to encash and potential to succeed.

Place: SBSSTC Ferozepur
Date: July 9, 2013 Shelja Tayal

ABSTRACT

This thesis is intended to present investigations on Biogeography Based Optimization (BBO) algorithms and introduce a new variant to improve convergence performance. The proposed variant is tested well on testbed of benchmark functions and then applied on a real world problem of evolving optimal design of 6-element Yagi-Uda antenna for gain maximization. Biogeography is the study of the geographic distribution of organisms throughout the landscape over time. It examines how do species migrate among islands (via flotsam, wind, flying and swimming, etc.) due to geographical and environmental conditions. BBO is one of most popular swarm based optimization algorithm that has shown impressive performance over other Evolutionary Algorithms (EAs). BBO consists of two operators (i) Migration Operator: It is a probabilistic operator that improves the solution fitness of poor habitats by receiving features from good habitats. (ii) Mutation Operator: It is second probabilistic operator that modifies the values of some of randomly selected solution features of a few habitats that are intended for exploration of newer solutions within the search-space.

Immigration Refusal Biogeography Based Optimization (IRBBO), Enhanced Biogeography Based Optimization (EBBO), Blended Migration are the most improved version of the BBO. In this thesis, a new variant of BBO (migration operator) is proposed to get faster convergence as compared to other EAs. This proposed migration variant is named as Graded Emigration. Graded Emigration is investigated for comparative study along with other BBO variants as Graded Emigration Biogeography Based Optimization (GE-BBO), Graded Emigration Enhanced Biogeography Based Optimization (GE-EBBO), Graded Emigration Immigration Refusal Biogeography Based Optimization (GE-IRBBO). The proposed variant subjected to evolve solutions for a testbed of benchmark functions having multimodalities and deceptive gradient benchmark performance (i.e., Dejong, Ackley, Griewank, Rastrigin

and Rosenbrock) along with other known EAs for comparison. There after applied to real world problem of designing six-element Yagi-Uda antenna for maximum gain to observe convergence performance.

A Yagi-Uda antenna is one of widely used antenna designs due to high gain capability, low cost and ease of construction. It is simple to construct and has a high gain, typically greater than 10dB at VHF and UHF frequency range. Yagi-Uda consists of three types of elements: (a) Reflector: Biggest among all and is responsible for blocking radiations in one direction. (b) Feeder: That is fed with the signal to be transmitted from transmission line. (c) Directors: These are usually more than one in number and responsible for unidirectional radiations. The physical parameters of Yagi-Uda antenna (element-lengths and spacings between adjacent elements) bear highly complex and non-linear relationship with gain, impedance and Side Lobe Level, etc. This antenna design problem, further, complicates as the number of antenna elements are increased with the objective of achieving higher directional gain. To evaluate Yagi-Uda antenna for gain, impedance, etc., a Method of Moments (MoMs) based antenna modeling free software, called Numerical Electromagnetics Code (NEC), is used along with algorithmic programming in C++.

In this thesis, Graded Emigration algorithms have been investigated to evolve solutions to benchmark optimization functions and to optimize maximum gain for Yagi-Uda antenna design for maximal gain. The comparative analysis of BBO, PSO and combinational BBO-PSO is also presented for antenna design optimization problem for multiple evolutionary runs. From experimental outcomes it has been observed that the GE-EBBO performs overall best among the variants explored in this thesis. The maximum gain achieved using GE-EBBO is 13.84 dB and by using Combined BBO-PSO gain achieved is 13.85 dB for six-element Yagi-Uda antenna.

This thesis is outlined as follow: Chapter 1 is devoted to introduction to M.Tech. thesis as a whole that includes introduction to Research Topic, Motivation, Methodologies, Contributions, Research Findings and outline of Thesis. Chapter 2 starts with the literature survey giving an overview of BBO & PSO algorithms and their most popular variants. It also presents a gentle introduction to Yagi-Uda antenna and AI and non AI based approaches followed to evolve optimal antenna designs. Chapter 3 is dedicated to study of biogeography, BBO & PSO algorithmic flow and their most popular variants reported, till date. It also presents the introduction to the proposed variant of Graded Emigration. Chapter 4 is devoted to introduce testbed of benchmark optimization functions. It also discusses how Yagi-Uda antenna design problem can be formulated as optimization problem. In Chapter 5, Firstly, NEC software is discussed that is used to evaluate wire antennas for gain, impedance, SLL, etc. Secondly, implementation algorithmic flow of BBO, PSO and Combined PSO-BBO in C++ environment, are discussed in detail. Chapter 6 represents average of multiple runs of simulations for convergence performance for benchmark functions and then for optimization of Yagi-Uda antenna. Best results and average for gain maximization in tabulated form are also represented in this chapter. Lastly, conclusion and future scopes of this research are discussed in Chapter 7.

Place: Ferozepur | Shelja Tayal (1169062)

Date: July 9, 2013

ABBREVIATIONS

Abbreviations	Description
ABC	Artificial Bee Colony
ACO	Ant Colony Optimization
AI	Artificial Intelligence
BBO	Biogeography Based Optimization
CDB	Console Debugger
CLPSO	Comprehensive Learning Particle Swarm Optimization
CPU	Central Processing Unit
DE	Differential Evolution
EA	Evolutionary Algorithm
EC	Evolutionary Computation
EBBO	Enhanced Biogeography Based Optimization
GA	Genetic Algorithm
GE-IRBBO	Graded Emigration Immigration Refusal
GE-BBO	Graded Emigration Biogeography Based Optimization
GE-EBBO	Graded Emigration Enhanced Biogeography Based Optimization
GUI	Graphical User Interface
HSI	Habitat Suitability Index
IDE	Integrated Development Environment
IRBBO	Immigration Refusal

Abbreviations	Description
NEC	Numerical Electromagnetics Code
PSO	Particle Swarm Optimization
SA	Simulated Annealing
SGA	Stud Genetic Algorithm
SIV	Suitability Index Variable
SI	Swarm Intelligence
SLL	Side Lobe Level
UHF	Ultra High Frequency
UOD	Universe of Discourse
VHF	Very High Frequency

NOTATIONS

Symbols	Description
μ_k	Emigration Rate of k-th habitat
λ_k	Immigration Rate of k-th habitat
HSI_{max}	Maximum Habitat Suitability Index
HSI_{min}	Minimum Habitat Suitability Index
E	Maximum Emigration Rate
I	Maximum Immigration Rate
NP	Size of Population
H	Habitat
M	Number of SIVs
$mRate$	Mutation Rate
C	Constant
L_s	Length of Yagi-Uda antenna elements
S_s	Spacing of Yagi-Uda antenna elements
$pbest$	Past Best
$gbest$	Global Best
n	Number of Iterations
X_i	i-th particle in the swarm
V_i	Velocity of i-th particle
P_i	Previous Best position of the i-th particle

Symbols	**Description**
P_g	Global Best position of the particle
w	Inertia weight
ψ_1	Cognitive Parameter
ψ_2	Social Parameter
χ	Constriction Factor
r_1, r_2	Random Numbers
$\pm V_{max}$	Maximum Velocity
λ	Wavelength

LIST OF FIGURES

LIST OF TABLES

CONTENTS

CHAPTER 1

INTRODUCTION

This thesis presents investigational studies to improve Biogeography Based Optimization algorithm and their variants. This introductory chapter presents an overview of the thesis, as a whole and introduces to the research topic, motivation, methodologies, objectives, contributions.

1.1 Introduction

Like other Evolutionary Algorithms (EAs), e.g., Particle Swarm Optimization (PSO) [Kennedy and Eberhart, 1995], Genetic Algorithm (GA) [Michalewicz, 1998], Differential Evolution (DE) [Storn and Price, 1997], Evolutionary Programming [Yao et al., 1999], Artificial Bee Colony (ABC) [Karaboga and Basturk, 2007], Ant Colony Optimization (ACO) [Dorigo et al., 2006], Biogeography Based Optimization (BBO) is population based stochastic algorithm. Biogeography is the study of distribution of species over space and time. Robert MacArthur and Edward Wilson first discovered and developed the mathematical models that govern the distribution of species [MacArthur and Wilson, 1967]. This motivates the development of BBO as an application of biogeography to EAs. BBO is a new swarm based optimization technique introduced by Dan Simon in 2008. It is based on science of biogeography where features sharing among various habitats, i.e., potential solution, is accomplished with migration operator and exploration of new features is done with mutation operator.

In [Du et al., 2009], the immigration refusal in BBO is proposed in order to improve its performance. As the modifications have made in BBO by many researches, Pattnaik *et al.* have proposed Enhanced Biogeography Based Optimization (EBBO) in which duplicate habitats created due to migration is modified with random mutation to increase the exploitation ability of BBO [Pattnaik et al., 2010]. Ma and Simon introduced new migration operator, i.e., Blended migration, to solve the constrained optimization problem and make BBO convergence faster [Ma and Simon, 2010, 2011].

In this thesis, a new variant of BBO is proposed by doing migration variations in BBO & its variants to make the convergence faster as compared to other EAs discussed. The proposed variant is tested well on testbed of benchmark functions and applied on a real world problem of evolving optimal design of six-element Yagi-Uda antenna.

In the area of benchmark functions a broad range of published test functions exists, designed to stress different parts of a global optimization algorithm, i.e., Dejong/Sphere, Ackley, Griewank, Rastrigin and Rosenbrock functions. These functions have the strength of an analytical expression with a known global minimum. There are many unimodal and multimodal benchmark functions. which are commonly used to critically test the performance of numeric optimization algorithms. These functions are chosen because of their particularities, which render their optimization difficult.

A Yagi-Uda antenna was invented in 1926 by H. Yagi and S. Uda at Tohoku University in Japan, however, published in English in 1928. Since its invention, continous efforts have been put in optimizing the antenna for gain, impedance, SLL and bandwidth using different optimization techniques based on traditional mathematical approaches and modern heuristic of Artificial Intelligence (AI) techniques.

1.2 Motivation

BBO has shown its ability to solve optimization problems. However, in order to improve this advantage relative to other heuristic algorithms, it is necessary to improve BBO. Several variants of BBO algorithm are introduced to get better results than BBO by adding new features in BBO algorithm like Immigration Refusal, EBBO, Blended Migration. This shows that BBO is an algorithm that has much promise and merits for further development and investigation. This motivates to propose new algorithms by doing migration variations in BBO & its variants to get better results. The name of new algorithms by doing migration variations in BBO, EBBO, IRBBO is Graded Emigration Biogeography Based Optimization (GE-BBO), Graded Emigration Enhanced Biogeography Based Optimization (GE-EBBO), Graded Emigration Immigration Refusal (GE-IRBBO) respectively. It's detail is given in the further chapters.

1.3 Objectives

The primary objectives of this research work are summarized as follow:

1. To introduce a new variant of BBO by doing migration variations in BBO named as Graded Emigration (GE-BBO) to get faster convergence performance as compared to other EAs. Graded Emigration is also investigated with BBO variants as GE-EBBO and GE-IRBBO.
2. To test the Graded Emigration on testbed of benchmark function.
3. To compare the results of various stochastic algorithms by applying it on the application of six-element Yagi-Uda antenna.
4. To compare the results of BBO and PSO and Combined PSO-BBO by applying it on the real world problem of evolving optimal design of six-element Yagi-Uda antenna.

1.4 Methodology

The Methodology followed is:

1. One of the foremost requirement to make these new algorithms is to deep study and understand BBO & its variants.
2. Second difficult requirement is to design programs of all these algorithms and to implement it on some platform to get the output. These algorithms are tested with a well known testbed of benchmark functions on Qt Creator platform.
3. Qt Creator platform is very helpful. Qt Creator is an Integrated Development Environment (IDE) that provides tools to design and develop applications with the Qt application framework. The real strength of Qt is the ability to integrate it into a C++ application.

1.5 Contributions

The main contributions of this report are:

1. Various BBO algorithms & its variants have been investigated for different swarm sizes.

2. Several benchmark functions contribute to test the new proposed graded algorithms to evolve optimal solution.

3. To optimize the wire lengths of Yagi-Uda antenna and spacings in between them, NEC2 (Numerical Electromagnetics Code version 2), is used to evaluate the antenna designs for gain, input impedance, bandwidth and beamwidth, etc.

1.6 Thesis Outline

Chapter 2 starts with the literature survey giving an overview of BBO & PSO algorithms and their most popular variants. It also presents a gentle introduction to Yagi-Uda antenna and AI and non AI based approaches followed to evolve optimal antenna designs.

Chapter 3 is dedicated to study of biogeography, BBO & PSO algorithmic flow and their most popular variants reported, till date. It also presents the introduction to the proposed variant of Graded Emigration.

Chapter 4 is devoted to introduce testbed of benchmark optimization functions. It also discusses how Yagi-Uda antenna design problem can be formulated as optimization problem.

In Chapter 5, Firstly, NEC software is discussed that is used to evaluate wire antennas for gain, impedance, SLL, etc. Secondly, implementation algorithmic flow of BBO, PSO and Combined PSO-BBO in C++ environment, are discussed in detail.

Chapter 6 represents average of multiple runs of simulations for convergence performance for benchmark functions and then for optimization of Yagi-Uda antenna. Best results and average for gain maximization in tabulated form are also represented in this chapter.

Lastly, conclusion and future scopes of this research are discussed in Chapter 7.

CHAPTER 2

LITERATURE SURVEY

This Chapter contains the overview of various EAs like BBO & its variants, PSO. These alorithms are tested on benchmark functions and investigated on Yagi-Uda antenna to get the optimal solution.

2.1 Introduction

Until the 1960s, the science of biogeography was mainly descriptive and historical [Darwin, 1859, 1964; Wallace, 1876]. In the early 1960s, Robert MacArthur and Edward Wilson began working together on mathematical models of biogeography, their work culminated in a classic publication of *The Theory of Island Biogeography* under arbitrary conditions [MacArthur and Wilson, 1967]. Their interest was primarily focused on the distribution of species among neighboring islands. They were interested in mathematical models for the extinction and migration of species. The application of biogeography to engineering is similar to what has occurred in the past few decades with GAs, ACO, PSO and other areas of computational intelligence. After BBO, BBO variants like Immigration Refusal, EBBO, Blended Migration are introduced. These stochastic algorithms are tested by benchmark functions.

Yagi-Uda antenna was invented by H. Yagi and S. Uda in 1926 at Tohoku University in Japan, but first published in English in 1928 and it has been extensively used as an end-fire antenna [Yagi, 1928]. There are no simple formulas for designing Yagi-Uda antennas due to the complex relationships between physical parameters such as element length, spacing, and diameter. So many researchers have proposed different algorithms for the optimized design of Yagi-uda antenna. Yagi-Uda antenna is designed by stochastic algorithms.

2.2 Biogeography Based Optimization

The study of the geographic distribution of organisms throughout the landscape is known as Biogeography. It examines how the species migrate between islands via flotsam, wind, flying, swimming, due to geographical variation in physical environment. BBO results presented by researchers are better than other optimization techniques, like PSO, GAs, SA, DE, etc. [Baskar et al., 2005; Jones and Joines, 1997; Rattan et al., 2008; Singh et al., 2012a, 2013a; Singh and Sachdeva, 2012a,b; Singh et al., 2013b,c,d,d,e, 2012b; Tayal et al., 2013; Venkatarayalu and Ray, 2003].

The science of biogeography can be traced to the work of nineteenth century naturalists such as Alfred Wallace and Charles Darwin. Until the 1960s, biogeography was mainly descriptive and historical study. In the early 1960s, Robert MacArthur and Edward Wilson began working together on mathematical models of biogeography, their work culminating with the classic 1967 publication *The Theory of Island Biogeography* [MacArthur and Wilson, 1967]. Their interest was primarily focused on the distribution of species among neighboring islands. They were interested in mathematical models for the extinction and migration of species. Since MacArthur and Wilsons work, biogeography has become a major area of research [Hanski and Simberloff, 1997]. The application of biogeography to engineering is similar to what has occurred in the past few decades with GAs, neural networks, fuzzy logic, PSO and other areas of computer intelligence.

BBO has certain features common with other swarm based algorithms. Like GAs and PSO, BBO has a way of sharing information between solutions. GA solutions *die* at the end of each generation, while PSO and BBO solutions survive forever (although their characteristics change as the optimization process progresses). PSO solutions are more likely to clump together in similar groups, while GA and BBO solutions do not necessarily have any built-in tendency to cluster.

Dan Simon introduced yet another swarm based stochastic optimization technique based on science of biogeography where features sharing among various habitats [Simon, 2008], i.e., potential solutions, is accomplished with migration operator and exploration of new features is done with mutation operator. Singh *et al.* have presented BBO as a better optimization technique for Yagi-Uda antenna designs, in [Singh et al., 2010].

2.2.1 Immigration Refusal

Du *et al.* have proposed the immigration refusal in BBO in order to improve its performance [Du et al., 2009]. Features from Evolutionary Strategy (ES) are used for BBO modification.

After the modification of BBO, F-tests and T-tests are used to demonstrate the differences between different implementations of BBO.

2.2.2 Enhanced Biogeography Based Optimization

Pattnaik [Pattnaik et al., 2010] have proposed Enhanced Biogeography Based Optimization (EBBO) in which duplicate habitats created due to migration, is modified with random mutation to increase the exploitation ability of BBO. Experiments have been conducted on unimodal and multimodal benchmark functions. EBBO gives excellent performance when compared with BBO and other versions of BBO.

2.2.3 Blended Migration

Ma and Simon introduced new migration operator, i.e., Blended migration, to solve the constrained optimization problem and make BBO convergence faster [Ma and Simon, 2010, 2011]. Firstly, Blended crossover operator of the GA outperformed standard BBO on a set of benchmark problems. Secondly, Blended BBO algorithm is compared with solutions based on a Stud Genetic Algorithm (SGA) and PSO.

2.3 Particle Swarm Optimization

PSO also belongs to the category of SI (Swarm Intelligence) [Eberhart et al., 2001] useful in solving global optimization problems. It was originally proposed by James Kennedy, as a simulation of social behavior and was introduced as an optimization method in 1995 [Kennedy and Eberhart, 1995; Shi and Eberhart, 1999]. PSO is an EC (Evolutionary computation) technique related to artificial life, specifically to swarming theories as it involves simulation of social behaviors. PSO implementation is easy and computationally inexpensive, since its memory and CPU speed requirements are low [Shi et al., 2001]. Moreover, it does not require gradient information of the fitness function but only its values. PSO has been proved to be an efficient method for many global optimization problems and in some cases it does not suffer from the difficulties experienced by other EAs [Shi and Eberhart, 1999].

Particle swarm algorithm originated from flocking behaviour of birds for getting maximum protection from predators [Heppner and Grenander, 1990]. A simulation program was developed to generate a bird flock for a hollywood film [Reynolds, 1987]. In this simulation, a point on the screen was defined as food, called the cornfield vector, the idea was to allow birds to find food through social learning by observing the behavior of nearby birds, who seemed nearer to the food source. The optimization potential was realized in the initial experiments and the algorithm was modified to incorporate topological rather than Euclidean neighborhoods and multi-dimensional search was attempted successfully.

PSO usually initializes the population by assigning each particle an arbitrary random starting position in the solution space with a randomized velocity. GAs use selection, crossover and mutation to replace less fit individuals by combining the traits of high performing chromosomes/solutions. However, in PSO, members of the particle swarm persist over time, retaining their identities and improving through imitation and interactions of best performing particles/solutions in the swarm.

2.4 Benchmark Functions

Global optimization has a lot of real-world applications, both of discrete and non-discrete nature. Among them are chemical applications such as structure optimization of molecules and clusters, engineering problems such as component design, logistics problems like scheduling and routing and many others. Despite the typical practical finding that a general global optimization algorithm usually is much less efficient than specific versions tuned to the problem at hand, it is still of interest to gauge the baseline performance of a global optimization scheme using benchmark problems. Most recent examples of such tests [Dieterich and Hartke, 2012; Mariani et al., 2011; Pan et al., 2010; Zhao et al., 2010], it is customary to employ certain standard benchmark functions, with the implicit (but untested) assumption that the difficulty of these benchmark functions roughly matches that of real-world applications. Some of these benchmark functions even are advertised as particularly challenging.

EA based global optimization strategies developed in the challenging, real-life area of atomic and molecular cluster structure optimization. Whitley *etal*. argued that many of the standard benchmark functions should be relatively easily solvable due to inherent characteristics like symmetry and separability [Whitley et al., 1996]. Some functions even appeared to get easier as the dimensionality of the function increases.

In the area of benchmark functions a broad range of published test functions exists, designed to stress different parts of a global optimization algorithm. Among the most popular ones are Dejong/Sphere, Ackley, Griewank, Rastrigin, Rosenbrock functions. These functions have the strength of an analytical expression with a known global minimum.

2.5 Yagi-Uda Antenna

A Yagi-Uda antenna was invented in 1926 by H. Yagi and S. Uda at Tohoku University [Uda and Mushiake, 1954] in Japan, however, published in English [Yagi, 1928]. Since its invention, continuous efforts have been put in optimizing the antenna for gain, impedance, SLL and bandwidth using different optimization techniques based on manual, traditional mathematical approaches [Bojsen et al., 1971; Chen and Cheng, 1975; Cheng and Chen, 1973; Cheng, 1971, 1991; Reid, 1946; Shen, 1972] and Artificial Intelligence (AI) based techniques [Baskar et al., 2005; Jones and Joines, 1997; Li, 2007; Singh et al., 2010, 2007; Venkatarayalu and Ray, 2004; Wang et al., 2003].

Yagi aerials approximate design was proposed for maximum gain in [Fishenden and Wiblin, 1949]. Ehrenspeck and Poehler have given a manual approach to maximize the gain of the antenna by varying various lengths and spacings of its elements [Ehrenspeck and Poehler, 1959].

Later on, with the availability of improved computational facilities at affordable prices made it possible to optimize antennas numerically. A numerical optimization technique was proposed to calculate the maximum gain of Yagi-Uda antenna arrays with equal and unequal spacings between adjoining elements. Optimum design of Yagi-Uda antenna where antenna gain function is proved to bear a highly non-linear relationship with its geometric parameters.

In 1975, John Holland introduced Genetic Algorithms (GAs) as a stochastic, swarm based AI technique, inspired from natural evolution of species, to optimize arbitrary systems for certain cost function. Then many researchers investigated GAs to evolve solutions to engineering problems including Yagi-Uda antenna for gain, impedance and bandwidth, separately [Altshuler and Linden, 1997; Correia et al., 1999; Jones and Joines, 1997] and collectively [Kuwahara, 2005; Venkatarayalu and Ray, 2003; Wang et al., 2003]. Baskar *et al.*, have optimized Yagi-Uda antenna using Comprehensive Learning Particle Swarm Optimization (CLPSO) and presented better results than other optimization techniques [Baskar et al., 2005]. Li has used Differential Evolution (DE) to optimize geometrical parameters of a Yagi-Uda antenna and illustrated the capabilities of the proposed method with several Yagi-Uda antenna designs in [Li, 2007]. Singh *et al.* have explored another useful stochastic global search and optimization technique named as Simulated Annealing (SA) for the optimal design of Yagi-Uda antenna [Singh et al., 2007].

In 2008, Dan Simon introduced yet another swarm based stochastic optimization technique based on science of biogeography where features sharing among various habitats (potential solutions) is accomplished with migration operator and exploration of new features is done with mutation operator [Simon, 2008]. Singh *et al.* have presented BBO as a better optimization technique for Yagi-Uda antenna designs [Singh et al., 2010].

2.6 Conclusion

After the evolution in BBO, BBO variants has introduced, i.e., immigration refusal, EBBO and Blended migration and show better convergence results. In this thesis, new algorithms are proposed by doing migration variations in BBO & its variants. All these EAs are tested on various benchmark functions and applied on a real world problem of Yagi-Uda antenna and represented in the further Chapters.

Yagi-Uda antenna has many number of input parameters and have complex relationship between them. After the evolution of Yagi-Uda antenna designing with artificial intelligence and various optimization techniques (GA, PSO, DE, SA and BBO & its variants) to optimize gain, input impedance, SLL, etc. Proposed graded algorithm is also applied for optimum design of 6-element Yagi-Uda antenna for maximum gain.

CHAPTER 3

BBO AND PSO ALGORITHMS

In this chapter, various stochastic algorithms like BBO & its variants and PSO are discussed. BBO has two major operators, viz. migration and mutation. Proposed variant of BBO, viz. Graded Emigration improved performance of BBO & it's variants is discussed in this chapter. This chapter is dedicated to all the variants of BBO and their psuedocodes. It also includes the description of nature inspired PSO algorithm.

3.1 Introduction

Most of AI based EAs are stochastic in nature that uses multiple solutions at a time to evolve better solutions iteratively by imitating one or another natural phenomenon. BBO and PSO are similar EAs those have been developed by imitation of biogeography study and flocking behaviour of birds and fish, etc. BBO is based on science of biogeography and consists of two operators, i.e., migration and mutation operator. Simple migration may leads to same type of habitats. To increase the diversity in the population is the objective of improved performance in BBO. Different BBO variants like Immigration Refusal, EBBO and Blended Migration algorithms are introduced. PSO is based on the social behaviour of birds like the way they move, synchronize without colliding and try to move towards the center of the flock. BBO and PSO solution features survive forever and retain their features from one generation to the next.

3.2 Biogeography

As name suggests, BBO is a population based global optimization technique developed on the basis of the science of biogeography, i.e., study of the distribution of animals and plants among different habitats over time and space.

Originally, biogeography was studied by Alfred Wallace and Charles Darwin mainly as descriptive study [Darwin, 1859, 1964; Wallace, 1876]. However, in 1967, the work carried out by MacAurthur and Wilson changed this view point and proposed a mathematical model for biogeography and made it feasible to predict the number of species in a habitat [MacArthur and Wilson, 1967]. Mathematical models of biogeography describe migration, speciation, and extinction of species in various islands. Habitats that are well suited residences for biological species are referred to have high Habitat Suitability Index (HSI) value.

3.2.1 BBO Terminology

Some of the important terms used in BBO algorithm are:

1. **Island** The term *island* is used for any habitat that is geographically isolated from other habitats.
2. **Habitats** Habitats that are well suited residences for biological species.
3. **Habitat Suitability Index** HSI is analogues to fitness in other EAs whose value depends upon many factors such as rainfall, diversity of vegetation, diversity of topographic features, land area and temperature etc.
4. **SIV** The factors/variables that characterize habitability are termed as Suitability Index Variables (SIVs).
5. **Immigration** Immigration is the arrival of new species into a habitat or population.
6. **Emigration** Emigration is the act of leaving ones native region.
7. **Migration** Migration is a probabilistic operator that improves HSI of poor habitats by sharing features from good habitats. It depends on the immigration and the emigration rate.
8. **Mutation** Mutation is an another probabilistic operator that modifies the values of some randomly selected SIVs of some habitats that are intended for for better solutions by increasing the biological diversity in the population.

The habitats with HSI tend to have a large population of its resident species, that is responsible for more probability of emigration (emigration rate, μ) and less probability of immigration (immigration rate, λ) due to natural random behavior of species. Immigration is the arrival of new species into a habitat or population, while emigration is the act of leaving one's native region. On the other hand, habitats with low HSI tend to have low emigration rate, μ, due to sparse population, however, they will have high immigration rate, λ. Suitability of habitats with low HSI is likely to increase with influx of species from other habitats having high HSI. However, if HSI does not increase and remains low, species in that habitat go extinct that leads to additional immigration.

For sake of simplicity, linear relationship between HSI (or population) and immigration (and emigration) rates are assumed, and maximum values of emigration and immigration rates are made equal, i.e., $E = I$, as depicted graphically in Fig. 3.1

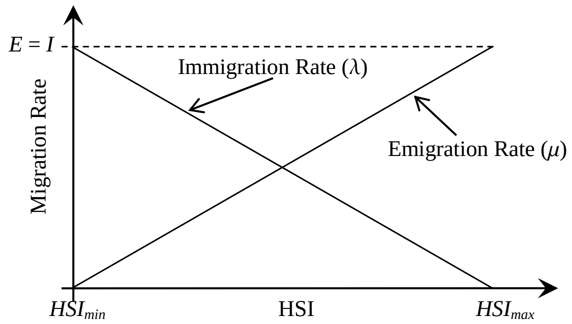

FIGURE 3.1: Migration Curves

For k-th habitat values of emigration rate, μ_k, and immigration rate, λ_k are given by (3.1) and (3.2), respectively.

$$\mu_k = E \cdot \frac{HSI_k}{HSI_{max} - HSI_{min}} \tag{3.1}$$

$$\lambda_k = I \cdot \left(1 - \frac{HSI_k}{HSI_{max} - HSI_{min}}\right) \tag{3.2}$$

The immigration of species from high HSI to low HSI habitats may raise the HSI of poor habitats. Good solutions have more resistance to change than poor solutions whereas poor solutions are more dynamic and accept a lot of features from good solutions.

Each habitat in a population of size NP, is represented by M-dimensional vector as $H = [SIV_1, SIV_2, \ldots, SIV_M]$ where M is the number of SIVs (features) to be evolved for maximal HSI. HSI is the degree of acceptability that is determined by evaluating the cost/objective function, i.e., $HSI = f(H)$. Algorithmic flow of BBO involves two mechanisms, i.e., migration and mutation, and are discussed in the following sections.

3.3 Migration variants

Migration variants are Standard Migration, Blended Migration, Immigration Refusal and Modified Clear Duplicate Operator (EBBO)

3.3.1 Standard BBO Algorithm

Migration is a probabilistic operator that improves HSI of poor habitats by sharing features from good habitats. During migration, i-th habitat, H_i (where $i = 1, 2, \ldots, NP$) use its immigration rate, λ_i given by (3.2), to decide probabilistically whether to immigrate or not. In case immigration is selected, then the emigrating habitat, H_j, is found probabilistically based on emigration rate, μ_j given by (3.1). The process of migration takes place by copying values of SIVs from H_j to H_i at random chosen sites, i.e., $H_i(SIV) \leftarrow H_j(SIV)$. The pseudo code of migration operator is depicted in Algorithm 1.

Algorithm 1 Standard Pseudo Code for Migration

```
for i = 1 to NP do
    Select Hi with probability based on λi
    if Hi is selected then
      for j = 1 to NP do
          Select Hj with probability based on μj
          if Hj is selected
             Randomly select a SIV(s) from Hj
             Copy them SIV(s) in Hi
           end if
      end for
    end if
end for
```

3.3.2 Immigration Refusal BBO Algorithm

In standard BBO, migration locations are decided on the basis of the emigration and immigration rates. If the habitat has a high emigration rate, then the probability of emigrating to other islands is high, whereas, the probability of immigration from other habitats is low. However, the low probability does not mean that immigration from low fit solution will never happen. Once in a while a high fit solution can tend to receive solution features from a low fitness solutions that may ruin the high HSI of the better habitat. So, when the SIVs from habitat which has low fitness try to emigrate to other habitats, the receiving habitats should carefully consider whether to accept these SIVs or not. If the emigration rate of the habitat is less than some threshold, and its fitness is also less than that of the immigrating habitat, then the immigrating island will refuse this migration. This idea of conditional migration is known as immigration refusal [Du et al., 2009]. Immigration Refusal BBO variant is investigated, in this paper, for evolutionary performance here whose pseudo code is depicted in Algorithm 2.

Algorithm 2 Pseudo Code for Immigration Refusal

```
for i = 1 to NP do
    Select H_i with probability based on λ_i
    if H_i is selected then
      for j = 1 to NP do
          Select H_j with probability based on μ_j
        if  H_j is selected
          if ((fitness(H_j)>fitness(H_i))
            apply migration
          end if
        end if
      end for
    end if
end for
```

3.3.3 Enhanced BBO Algorithm

The exploitation ability of BBO is good as migration operator can efficiently share the SIVs between habitats. However, this creates similar habitats which decreases the diversity of the population. To increase diversity in the population so as to increase the exploration ability, clear duplicate operator is used. This variant is named as Enhanced BBO (EBBO) presented in [Pattnaik et al., 2010], the same concept of standard migration and mutation is used. however, modified clear duplicate operator is incorporated to get better results and to make convergence faster. EBBO is investigated, in this paper, for convergence performance whose pseudo code is depicted in Algorithm 3.

Algorithm 3 Pseudo Code for Enhanced Biogeography Based Optimization

```
for i = 1 to NP do
    Select H_i with probability based on λ_i
    if H_i is selected then
      for j = 1 to NP do
          Select H_j with probability based on μ_j
        if  H_j is selected
          if ((fitness(H_j) == fitness(H_i))
            eliminate duplicates
          end if
        end if
      end for
    end if
end for
```

3.3.4 Blended BBO Algorithm

A new migration operator called blended migration [Ma and Simon, 2011], which is the modification of the standard BBO migration operator, and which is motivated from blended crossover operator of GAs. In blended crossover operator, new genes values are generated by combination of both parental gene values, instead of simple exchange of gene values. In blended migration, SIV of habitat H_i is not simply replaced by SIV of habitat H_j. However, a new SIV value in Blended Migration comprised of SIVs of both participating habitats, as given by equation (3.3). Blended Migration is also investigated here whose pseudo code is depicted in Algorithm 4.

$$H_i(SIV) \leftarrow \alpha \cdot H_i(SIV) + (1 - \alpha) \cdot H_j(SIV) \tag{3.3}$$

Algorithm 4 Pseudo Code for Blended Migration

```
for i = 1 to NP do
    Select Hi with probability based on λi
    if Hi is selected then
      for j = 1 to NP do
          Select Hj with probability based on μj
        if Hj is selected
            Hi (SIV) ← α.Hi + (1-α).Hj
        end if
      end for
    end if
end for
```

Here α is a real number between 0 and 1. It could be random or deterministic. In Blended BBO, exploration of search space for better solution is in built, therefore, may require less mutation rates.

3.4 Proposed Algorithms

In this thesis, a new concept of graded emigration is proposed for furthur improved convergence performance. This graded emigration is also investigated on other BBO variants and described in the following subsections:

3.4.1 Graded Emigration BBO (GE-BBO)

In standard migration and its other variants do decide emigrating and immigrating habitats and their SIVs probabilistically. Graded Emigration (GE) is a new migration variant introduced in this paper, where number of SIVs of each emigrating habitat and their SIV number are predecided where to migrate in accordance to with their fitness ranking. In GE the poorest habitat is completely replaced and the best habitat is preserved as it is, whereas the mediocre habitats are partially modified by sharing fixed number of SIVs from better habitats. The number of migrating SIVs are fixed, however their location is decided randomly. This proposed algorithm is named as Graded Emigration BBO.

Example 3.1 (Graded Emigration among 10 habitats having 10 SIVs in each habitat)**.** *For Graded Emigration in a population of 10 habitats having 10 SIVs in each habitat following steps are required to be followed:*

1. *Sort habitats in ascending order to their fitness values.*
2. *The last poor habitat constitute a new habitat in the ratio of 4:3:2:1 to replace the poorest in the population.*
3. *Next to the poorest is contributed by 90% by first, second, third and fourth best habitats in the 4:3:2:0.*
4. *Subsequently, the other poorer habitats partially modified by the better habitats as per the matrix given in the Algorithm 5.*

For 20 or 30 habitats the algorithm may be extended by doubling or triplicating the rows of the matrix X.

3.4.2 Graded Emigration EBBO (GE-EBBO)

In GE-BBO, there is a large probability of similar habitats. But in case of GE-EBBO, in place of standard migration, the migration process of GE-BBO is applied and then the same process of modified clear duplicate operator is integrated to reduce the similarity of habitats and thereafter named as GE-EBBO. In Short, GE-EBBO is the combination of GE-BBO with modified clear duplicate operator (EBBO) to get better results and to increase the exploration ability.

3.4.3 Graded Emigration IRBBO (GE-IRBBO)

In case of GE-IRBBO, in place of standard migration, the migration process of GE-BBO is applied and then the same process of immigration refusal is integrated in GE-BBO to make the convergence faster. In Short, GE-IRBBO is the combination of migration process of GE-BBO with Immigration Refusal to get better results and to increase the exploration ability.

Algorithm 5 Standard Pseudo Code for Graded Emigration

$$X[i][j] = \begin{bmatrix} 4 & 3 & 2 & 1 \\ 4 & 3 & 2 & 0 \\ 4 & 3 & 1 & 0 \\ 4 & 3 & 0 & 0 \\ 4 & 2 & 0 & 0 \\ 4 & 1 & 0 & 0 \\ 4 & 0 & 0 & 0 \\ 3 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{array}{l} \text{Poor Habitat} \\ \\ \\ \\ \\ \\ \\ \\ \\ \text{Best Habitat} \end{array}$$

for *i to NP* **do**
for *j*=1 to *X*[*i*][1] **do**
Randomly Select SIV(s) from *NP-th* Habitat and
copy to random SIV(*s*) in *i-th* Habitat
end for
for *j*=5 to *X*[*i*][1]+ *X*[*i*][2] **do**
Randomly Select SIV(s) from (*NP-1)-th* Habitat and
copy to random SIV(*s*) in *i-th* Habitat
end for
for *j*=8 to *X*[*i*][1]+ *X*[*i*][2]+ *X*[*i*][3] **do**
Randomly Select SIV(s) from (*NP-2)-th* Habitat and
copy to random SIV(*s*) in *i-th* Habitat
end for
for *j*=10 to *X*[*i*][1]+ *X*[*i*][2]+ *X*[*i*][3]+ *X*[*i*][4] **do**
Randomly Select SIV(s) from (*NP-3)-th* Habitat and
copy to random SIV(*s*) in *i-th* Habitat
end for
end for

3.5 Mutation

Mutation is another probabilistic operator that modifies the values of some randomly selected SIVs of some habitats that are intended for exploration of search space for better solutions by increasing the biological diversity in the population. Here, higher mutation rates are investigated on habitats those are probabilistically participating less in migration process. The mutation rate, $mRate$, for k-th habitat is given as (3.4)

$$mRate_k = C \times \min(\mu_k, \lambda_k) \tag{3.4}$$

where μ_k and λ_k are emigration and immigration rates, respectively, given by (3.1) and (3.2) corresponding to HSI_k. Here C is a constant and kept equal to 1, in this thesis, i.e., mutation rate is much higher as compared to other EAs to maintain high diversity in the population. The pseudo code of mutation operator is depicted in Algorithm 6.

Algorithm 6 Standard Pseudo Code for Mutation

```
mRate = C × min(μ_k, λ_k)
for i = 1 to NP do
   for j = 1 to length(H) do
        Select H_j(SIV) with mRate
        If H_j(SIV) is selected then
           Replace H_j(SIV) with randomly generated SIV
        end if
   end for
end for
```

3.6 Particle Swarm Optimization

The PSO algorithm is one of stochastic swarm intelligence based global search algorithms. The motivation behind the PSO algorithm is the social behavior of animals, viz. flocking of birds and fish schooling. The PSO has its origin in simulation for visualizing the synchronized choreography of a bird flock by incorporating certain features like nearest-neighbor velocity matching and acceleration by distance [Eberhart et al., 2001; Kennedy and Eberhart, 1995; Parsopoulos and Vrahatis, 2002; Shi et al., 2001]. Later on, it was realized that the simulation could be used as an optimizer and resulted in the first simple version of PSO.

1. **Flocks** There is something about the way they move, synchronize, fly-without colliding, and resulting in amazing choreography. In 1987, a very influential simulation of bird-flock was published by Craig Reynolds [Reynolds, 1987]. Reynolds assumed that flocking birds were driven by three concerns:

   (a) Avoid colliding with their neighbors.

   (b) Match with velocities of their neighbors.

   (c) Try to move towards the center of the flock.

   These simple rules resulted in a very realistic flocking behavior that showed coherent clusters of boids (name of simulated birds) whirling through space, splitting to flock around obstacles and rejoining again. His simple non-centralized algorithm was used in many animated cinematic sequences of flocks and herds.

2. **Schools and Social Behaviour** In their book [Eberhart et al., 2001], perfectly described the rationale behind the idea that originated PSO was perfectly described as "Whenever people interact, they become more similar, as they influence and imitate one another. Norms and cultures are the result. Human physical behavior is not flock-like or school-like; the trajectories of human thoughts through high-dimensional cognitive space just might be." The particle swarm algorithm is really a simulation of the way minds think and of human social behavior.

   Regarding concordance they state, "The social phenomenon behind thinking is more complex than the choreographed behaviors of fish and birds. First, thinking takes place in belief space, whose dimensionality is far greater than three. Second, when two minds converge on the same point in cognitive space, we call it agreement, not collision." Each time it agrees, when travels to the same position in belief space (at least in some of the coordinates). When it disagrees, the distance in belief space increases. Imitative behavior is characterized by a velocity vector whose direction aims at another person's place in belief space.

3.6.1 Global-Best PSO Model

In PSO, the particles have adaptable *velocities* that determines their movement in the search space and *memory* which enable them for remembering the best position in the search space ever visited.

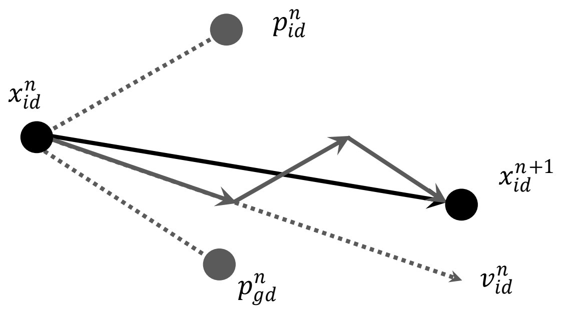

FIGURE 3.2: Movement of i-th particle in 2-dimensional search space

The position corresponding to the best fitness is known as past best, *pbest* and the overall best out of all NP the particles in the population is called global best, *gbest*. Consider that the search space is M-dimensional and i-th particle in the swarm can be represented by $X_i = [x_{i1}, x_{i2}, ...x_{id}..., x_{iM}]$ and its velocity can be represented by another M-dimensional vector $V_i = [v_{i1}, v_{i2}, ...v_{id}.., v_{iM}]$. Let the best previously visited position of i-th particle be denoted by $P_i = [p_{i1}, p_{i2}, ...p_{id}.., p_{iM}]$, whereas, g-th particle, i.e., $P_g = [p_{g1}, p_{g2}, ...p_{gd}.., p_{gM}]$ is globally best particle. Fig. 3.2 depicts the vector movement of particle element from location x_{id}^n to x_{id}^{n+1} in $(n+1)$-th iteration that is being governed by past best location, p_{id}^n, global best location, p_{gd}^n, locations and current velocity v_{id}^n. Alternatively, the whole swarm is updated according to the equations (3.5) and (3.6) suggested by Shi & Eberhart [Shi and Eberhart, 1999].

$$v_{id}^{m+1} = \chi(wv_{id}^m + \psi_1 r_1(p_{id}^m - x_{id}^m) + \psi_2 r_2(p_{gd}^m - x_{id}^m)) \tag{3.5}$$

$$x_{id}^{m+1} = x_{id}^m + v_{id}^{m+1} \tag{3.6}$$

Here, w is inertia weight, ψ_1 is cognitive parameter, ψ_2 is social parameter and constriction factor χ are strategy parameters of PSO algorithm, while r_1, r_2 are random numbers uniformly distributed in the range [0,1]. Generally the inertia weight, w, is not kept fixed and is varied as the algorithm progresses. The particle movements is restricted with maximum velocity, $\pm V_{max}$, to avoid jump over the optimal location as per search space requirements.

3.6.2 PSO Characterization

There are several parameters that need to be defined in order to successfully utilize PSO to solve a given problem [Mendes, 2004]

1. **Solution Encoding**: It is a M-dimensional vector representation of collection problem feature to be evolved for desired fitness function. This usually involves a minimum and a maximum value allowed in each dimension, thus defining a hyperspace.

2. **Fitness Function**: This function is degree of suitability/accepatibility also problem dependent and represents a measurement of a given solution. The function should somehow create a total ranking in the solution space.

3. **Population Size**: This parameter infuences the behavior of the algorithm. A very small population does not create enough interaction for the emergent behavior pertaining to PSO to occur. However, large population size may lead to more computational burden and consequently, take more evolutionary time. So the population size is to be decided as per the problem size and complexity.

4. **Acceleration Coefficients**: The acceleration coefficients ψ_1 and ψ_2 are usually set to the same value. In fact, people usually talk about ψ which sets the other two values $\psi_1 = \psi_2 = \psi/2$. If ψ is too small, the maximum step size becomes quite small and so the algorithm will explore very slowly and degrade its performance. There is a consensus among the researchers that step size is generally optimal if $\psi = 4.1$, however, not for every problem and every time.

5. **Constriction or Inertia coefficient**: It is not necessary to guess its value as given by equation (3.7). If the value of ψ is set to 4.1, then $\chi \approx 0.729$.

$$\chi = \frac{2k}{(2 - \psi - \sqrt{\psi^2 - 4\psi}} \tag{3.7}$$

where $k = [0, 1]$, $\psi = \psi_1 + \psi_2$, $\psi > 4$

6. **Maximum Velocity**: With the advent of the constriction coefficient, most researchers do not bother using this parameter. However, to avoid jump overs maximum velocity is fixed to some value less than unity.

7. **Neighborhood Topology**: If every particle is made to interact with every other in the swarm, then it becomes prove to fall into local optima. However, this may be avoided if swarm is divided into subgroups and every particle is made to interact with all members of its subgroup.

3.7 Conclusion

In this Chapter various nature inspired algorithms like BBO & its variants and PSO are discussed. It also contains the introduction of graded algorithms and these algorithms are experimented on benchmark functions and Yagi-Uda antenna design in further chapters.

CHAPTER 4

BENCHMARK TESTBED FOR OPTIMIZATION ALGORITHMS

This Chapter is dedicated to the evaluation platforms like benchmark functions (Dejong, Ackley, Griewank, Rastrigin, Rosenbrock) and Yagi-Uda Antenna on which various optimization algorithms are applied. Benchmark Functions and Yagi-Uda Antenna are used with an objective to determine and compare the performance of BBO & it's variants and proposed algorithms i.e., (GE-BBO, GE-EBBO, GE-IRBBO).

4.1 Introduction

There are many benchmark functions which are commonly used to critically test the performance of numeric optimization algorithms. These functions are chosen because of their particularities, which render their optimization difficult. These comprise

1. Multi-modality
2. Deceptive gradient information
3. The curse of dimensionality

There are many benchmark test functions like a few of them listed in Table 4.1 and used in this thesis to validate and compare the concept of GE with other variants. The application of Yagi-Uda antenna is also discussed in this chapter.

4.2 DeJong

Dejong/Sphere function is very simple and any algorithm capable of numeric optimization should solve it without any problem. It is unimodal function, with global minima located at $x = (0, \ldots, 0)$, with $f(x) = 0$.

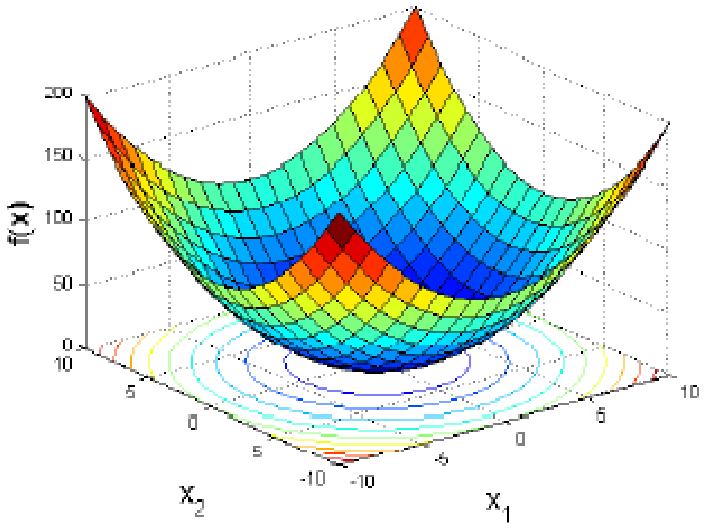

FIGURE 4.1: Dejong function Graph in two dimensions

4.3 Ackley

Ackley is a multimodal function with many local optima, however global minimum is $f(x) = 0$, is located at $x = (0, \ldots, 0)$. This function is difficult because optimization algorithms can easily be trapped in a local minima on it's way to the global minimization.

4.4 Griewank

Griewank function is strongly multimodal function with significant interaction among its variables, caused by the product term. This function has the interesting property that the number of local minima increases with dimensionality. The global minimum, $x = (100, 100, \ldots, 100)$, yields a function value $f(x) = 0$.

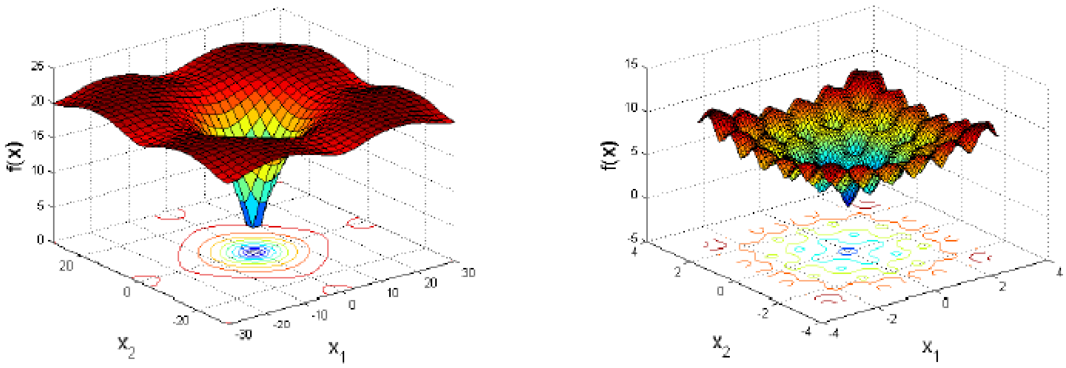

FIGURE 4.2: Ackley function Graph in two dimensions

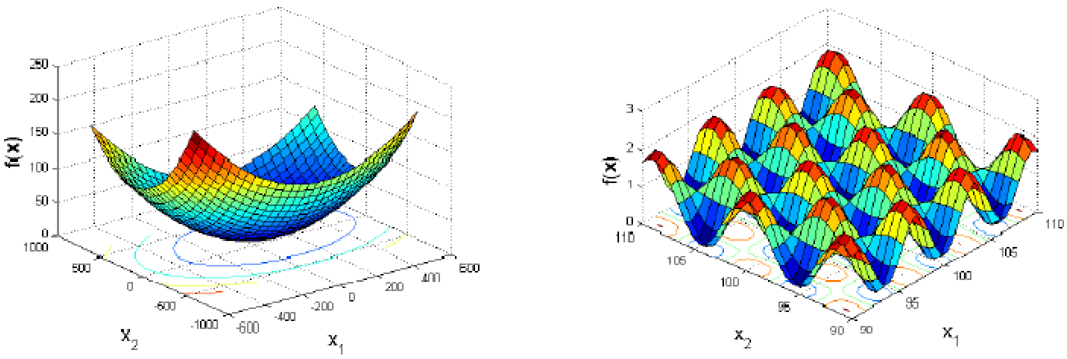

FIGURE 4.3: Griewank function Graph in two dimensions

4.5 Rastrigin

Rastrigin is a multimodal version of the spherical function, characterized by deep local minima arranged as sinusoidal bumps. The global minimum $f(x) = 0$, is located at $x = (0, \ldots, 0)$.

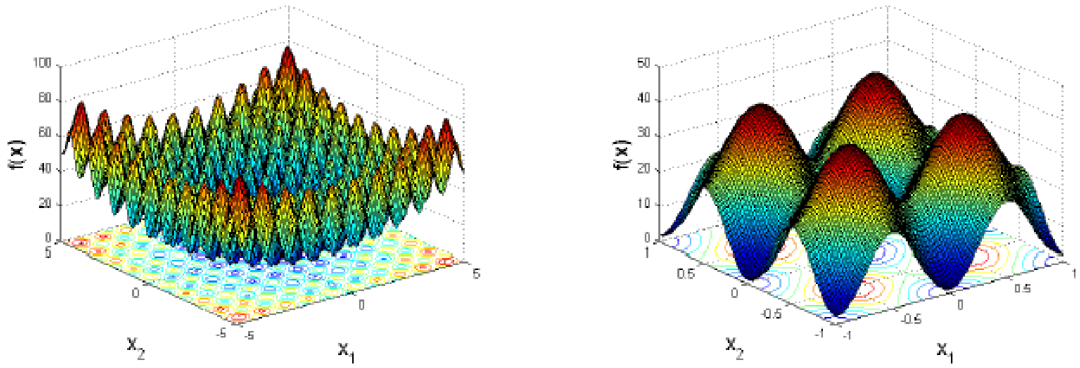

FIGURE 4.4: Rastrigin function Graph in two dimensions

4.6 Rosenbrock

Rosenbrock function variables are strongly dependent and gradient information often misleads algorithms. It's global minimum of $f(x) = 0$ is located at $x = (1, \ldots, 1)$.

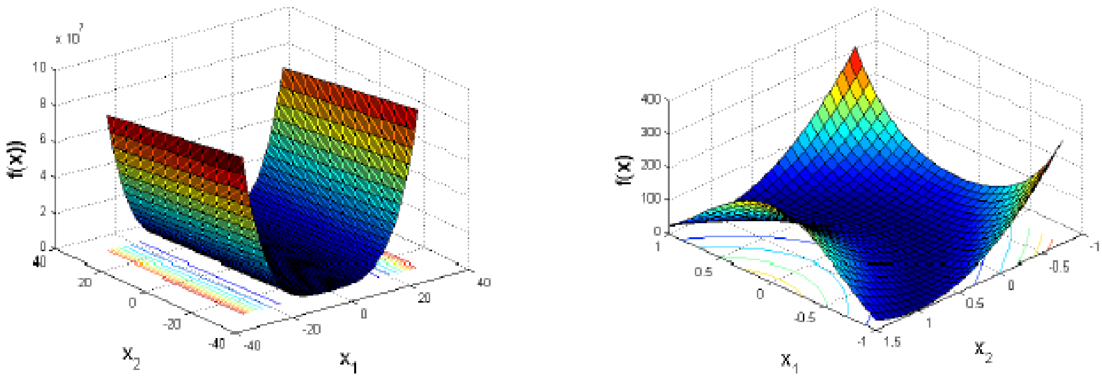

FIGURE 4.5: Rosenbrock function graph in two dimensions

4.7 Yagi-Uda Antenna

Antenna is an electrical device which forms an interface between free space radiations and transmitter or receiver. The choice of an antenna depends on various factors such as gain,

Function Name	Function (*f(x)*)	Search Space	Maximum
Dejong/Sphere	$\sum_{i=1}^{n} x_i^2$	$-100 \leq x_i \leq 100$	3.10^5
Ackley	$20 + e - 20e^{-0.2\sqrt{\frac{\sum_{i=1}^{n} x_i^2}{n}}} - e^{\frac{\sum_{i=1}^{n} \cos 2\pi x_i}{n}}$	$-30 \leq x_i \leq 30$	22.35040
Griewank	$1 + \frac{\sum_{i=1}^{n} (x_i - 100)^2}{4000} - \prod_{i=1}^{n} \frac{\cos(x_i - 100)}{\sqrt{i}}$	$-600 \leq x_i \leq 600$	1666.839 *and* 3676.839
Rastrigin	$\sum_{i=1}^{n} x_i^2 - 10\cos 2\pi x_i + 10$	$-5.12 \leq x_i \leq 5.12$	1210.6
Rosenbrock	$\sum_{i=1}^{n-1} 100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2$	$-30 \leq x_i \leq 30$	2508237869

Table 4.1: Benchmark Functions

impedance, bandwidth, frequency of operation, SLL, etc. A Yagi-Uda antenna is a widely used antenna design due to high forward gain capability, low cost and ease of construction. It is a parasitic linear array of parallel dipoles, one of which is energized directly by transmission line while the others act as a parasitic radiators whose currents are induced by mutual coupling. The characteristics of Yagi-Uda antenna are affected by all of the geometric parameters of array. It is simple to construct and has a high gain, typically greater than 10dB at VHF and UHF frequency range, i.e., 3 MHz to 3 GHz.

Yagi-Uda antenna is shown in the Fig. 4.6. Yagi-Uda antenna consists of three types of wire elements:

1. **Feeder or Driven Element**: Driven Element or Feeder is fed with the signal from transmission line to be transmitted. There is usually just one driven element. A dipole driven element will be resonant when its electrical length is half of the wavelength of the frequency applied to its feed point.

2. **Reflector**: Reflector is biggest among all and is responsible for blocking radiations in one direction. The reflector element is 5 percent is longer than the feed element. There is typically only one reflector; adding more reflectors improves performance very slightly. This element is important in determining the front-to-back ratio of the antenna.

3. **Directors**: Directors are usually more than one in number and responsible of unidirectional radiations. The lengths of directors reduces in the direction of radiations and depends upon the director spacing, the number of directors used in the antenna, the desired pattern, pattern bandwidth and element diameter.

Fig. 4.7 depicts a typical six-wire Yagi-Uda antenna all wires placed parallel to x-axis and along y-axis. Middle segment of the reflector element is placed at origin, $x = y = z = 0$, and excitation is applied to the middle segment of the feeder element. An incoming field sets up resonant currents on all the antenna elements which re-radiate signals. These re-radiated fields are then picked up by the feeder element, that leads to total current induced in the feeder equivalent to combination of the direct field input and the re-radiated contributions from the director and reflector elements.

The radiation or antenna pattern describes the relative strength of radiated field in various directions from the antenna at a constant distance. The radiation pattern is also called reception pattern as well, since it also describes the receiving properties of the antenna. The radiation pattern is three-dimensional, however, usually the measured radiation patterns are a two dimensional slice of the three-dimensional pattern in the horizontal and vertical planes. These pattern measurements are presented in either a rectangular or a polar format. A polar format of the gain versus orientation (radiation pattern) is useful when characterizing antennas. Some other important features of antenna that appears on plot are:

1. **Forward Gain**: Forward gain is the ability of an antenna to focus energy in a particular direction while transmitting/receiving energy better to/from a particular direction. To determine the gain or directivity of an antenna, a reference antenna is used to compare antenna performance. Forward gain is expressed in decibel (dB) relative to an isotropic source or a standard dipole in direction of maximum gain. Typically, higher the gain, more the efficient antenna performance and longer the range of the antenna will operate. Radiation pattern of a typical six-elements Yagi-Uda antenna is depicted in Fig. 4.7.

2. **Front to Back ratio**: The Front to Back ratio is used in describing directional radiation patterns for antennas. If an antenna radiates maximum in one direction, the F/B ratio is the ratio of the gain in the maximum direction to that in the opposite direction (180 degrees from the specified maximum direction) ans is also expressed in dB.

3. **Beamwidth**: Beamwidth is the angle between directions where the power is half the value at the direction of maximum gain which is -3dB. It gives the measure of directivity of antenna

4. **Sidelobes**: Antenna is not able to radiate all the energy in one preferred direction because some part of energy is inevitably radiated in other directions. Sidelobes are unwanted peaks in the gain at angles other than in forward direction, they reduce the amount of useful energy contained in the forward direction. The peaks are referred to as side lobes, as shown in Fig. 4.7, and commonly specified in dB down from the main lobe.

Other characteristics that do not appear on the polar plot but which are equally important are:

1. **Bandwidth**: Bandwidth is the range of frequency over which the antenna exhibits acceptable characteristics.

2. **Radiative impedance**: For an efficient transfer of energy, the radiative impedance of the antenna and transmission cable connecting them must be the same. Transceivers and their transmission lines are typically designed for 50Ω resistive impedance. If the antenna has an impedance different from 50Ω then there is a mismatch and an impedance matching circuit is required to avoid signal loss.

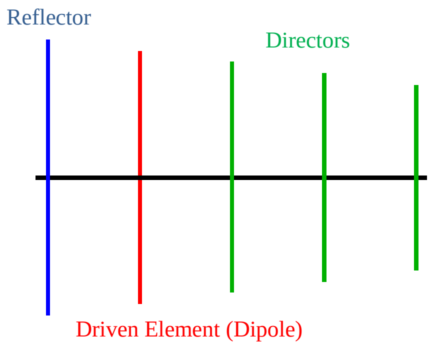

FIGURE 4.6: Yagi-Uda Antenna

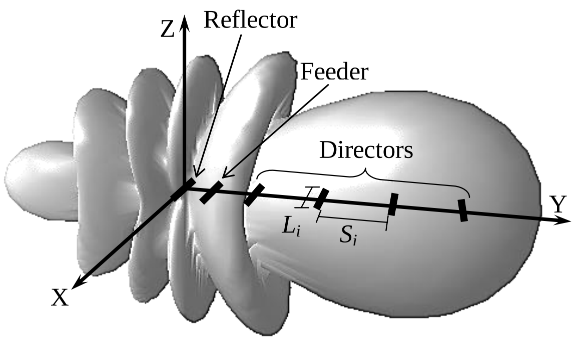

FIGURE 4.7: Radiation Pattern of a typical 6-wire Yagi-Uda Antenna

Designing a Yagi-Uda antenna involves determination of wire-lengths and wire-spacings in between to get maximum gain, desired impedance and minimum SLL at an arbitrary frequency of operation. An antenna with N elements requires $2N-1$ parameters, i.e., N wire lengths and $N-1$ spacings, that are to be determined. These $2N-1$ parameters, collectively, are represented as a string referred as a *habitat* in BBO given as (4.1).

$$H = [L_1, L_2, \ldots, L_N, S_1, S_2, \ldots, S_{N-1}] \tag{4.1}$$

where L_s are the lengths and S_s are the spacing of antenna elements. An incoming field sets up resonant currents on all the antenna elements which re-radiate signals. These re-radiated signals are then picked up by the feeder element, that leads to total current induced in the feeder equivalent to combination of the direct field input and the re-radiated contributions from the director and reflector elements.

4.8 Conclusion

In this Chapter, various evaluation platforms, i.e., benchmark functions and Yagi-Uda antenna are discussed. In this thesis, designing graded algorithm and proving it on benchmark functions, moreover, gain maximization of six-element Yagi-Uda antenna will be the target in the coming chapters. Algorithmic flow of BBO and PSO algorithms are discussed in the next chapter.

CHAPTER 5

IMPLEMENTATION

In this chapter, Firstly, NEC software developed for antenna design parameters evaluation, i.e., gain, input impedance, SLL, etc. Secondly, implementation algorithmic flow of BBO, PSO, Combined PSO-BBO in C++ environment, are discussed in detail.

5.1 Introduction

Swarm based algorithms like BBO & it's variants, PSO are nature inspired optimization techniques. In this Chapter, Combined PSO-BBO Algorithm is also introduced. All these Algorithms are experimented on various benchmark functions (Dejong, Ackley, Griewank, Rastrigin, Rosenbrock) and 6-element Yagi-Uda antenna to evolve near optimal solution for faster convergence. Designing a Yagi-Uda antenna is a complex optimization problem. The goal of the design process is to determine constructional detail of the antenna that meets some desired performance characteristics. A few of the characteristics that define an antenna performance are SLL, beamwidth, bandwidth, F/B ratio, size, gain, and input impedance. Qt Creator is used as to create BBO algorithms in C++, whereas NEC2 is used to evaluate all antenna designs for gain, impedance, etc.

5.2 Implementation Requirements

To design Yagi-Uda antenna and to test Optimization techniques on benchmark functions we require a programming platform of Qt Creator for C++ programming and NEC2 to evaluate antenna design based on method of moments. Their brief introduction is present in following subsections :

5.2.1 Qt Creator

Qt Creator is a cross-platform integrated development environment tailored to the needs of Qt developers. It provides:

1. C++ and JavaScript code editor
2. Integrated UI designer
3. Project and build management tools
4. gdb and CDB debuggers
5. Support for version control
6. Simulator for mobile UIs
7. Support for desktop and mobile targets

Qt Creator is part of Qt Quick, which allows designers and developers to create the kind of intuitive, modern-looking, fluid user interfaces that are increasingly used on mobile phones, media players, set-top boxes and other portable devices. Qt Creator enables collaboration between designers and developers. Designers work in a visual environment, while developers work in a full featured IDE and Qt Creator supports round-trip iteration from design to code, test and back to design.

5.2.2 C++ Development with Qt

Qt provides an intuitive C++ class library with a rich set of application build blocks for C++ development. Qt goes beyond C++ in the areas of inter-object communication and flexibility for advanced GUI development. Qt adds the following features to C++:

1. Powerful mechanism for inter-object communication called signals and slots.
2. Queryable and designable object properties.
3. Powerful events and events filters.
4. Contextual string translation for internationalization.

5. Sophisticated interval driven timers that make it possible to elegantly integrate many tasks in an event-driven GUI.

6. Hierarchical and queryable object trees that organize object ownership in a natural way.

7. Guarded pointers that are automatically set to 0 when the referenced object is destroyed, unlike normal C++ pointers which become dangling pointers when their objects are destroyed.

8. A dynamic cast that works across library boundaries.

5.2.3 Numerical Electromagnetics Code (NEC)

The old version of Numerical Electromagnetics code, i.e., NEC-2 is a computer code that runs through command line for analyzing the electromagnetic response of an arbitrary structure consisting of wires and surfaces in free space or over a ground plane. The analysis is accomplished by the numerical solution of integral equations for induced currents. The excitation may be an incident plane wave or a voltage source on a wire, while the output may include current and charge density, electric or magnetic field in the vicinity of the structure, and radiated fields.

The Numerical Electromagnetics Code (NEC-2) is a user-oriented computer code for analysis of the electromagnetic response of antennas and other metal structures. It is built around the numerical solution of integral equations for the currents induced on the structure by sources or incident fields. This approach avoids many of the simplifying assumptions required by other solution methods and provides a highly accurate and versatile tool for electromagnetic analysis.

5.2.4 How to use NEC

First of all, create a text file with *.nec* extension and write commands with parameters to create geometry and radiation pattern of antenna. Commands to create geometry and radiation pattern of antenna are as follow :

1. **Comment Cards (CM, CE)**: The data-card deck for a run must begin with one or more comment cards which can contain a brief description and structure parameters for the run. The cards are printed at the beginning of the output of the run for identification only and have no effect on the computation. Any alphabetic and numeric characters can be punched on these cards.

```
CM EXAMPLE
CM YAGI-UDA ANTENNA
CE Generated by Shelja Tayal
GW  1  7  -0.2389  0.0000  0.0000  0.2389  0.0000  0.0000  0.0034
GW  2  7  -0.2462  0.1942  0.0000  0.2462  0.1942  0.0000  0.0034
GW  3  7  -0.2196  0.4454  0.0000  0.2196  0.4454  0.0000  0.0034
GW  4  7  -0.2002  0.7596  0.0000  0.2002  0.7596  0.0000  0.0034
GW  5  7  -0.2083  1.2371  0.0000  0.2083  1.2371  0.0000  0.0034
GW  6  7  -0.2108  1.5546  0.0000  0.2108  1.5546  0.0000  0.0034
GE
GN  -1
EX  0  2  4  0  1.0000  0.0000
FR  0  1  0  0  300.0000  0.0000
RP  0  73  37 1110  0.0000  0.0000  5.0000  5.0000
EN
```

Figure 5.1: Input NEC File Format

2. **Wire Specification (GW)**: It is used to a string of segments to represent a straight wire.

3. **End of Run (EN)**: It is used to indicate to the program the end of all execution.

4. **Excitation (EX)**: It is used to specify the excitation for the structure. The excitation can be voltage sources on the structure, an elementary current source, or a plane wave incident on the structure.

5. **Frequency (FR)**: specify the frequency (frequencies) in Mega Hertz (MHZ).

6. **Ground Parameters (GN)**: It is used to specify the relative dielectric constant and conductivity of ground in the vicinity of the antenna. In addition, a second set of ground parameters for a second medium can be specified, or a radial wire ground screen can be modeled using a reflection coefficient approximation.

7. **Radiation Pattern (RP)**: It is used to specify radiation pattern sampling parameters and to cause program execution. Options for a field computation include a radial wire ground screen, a cliff, or surface-wave fields.

These all commands write into a text file in particular defined format and create text file with .nec extension, as depicted in Fig. 5.1.

After create a text file, it passes through the NEC2.exe as a input file. Then it create a output text file with .OUT extension contains all characteristics of antenna like frequency, wavelength, input impedance, gain and run time.

5.3 Implementation Algorithm

Design of Yagi-Uda antenna is done in four algorithms, first is Fitness Algorithm to design an antenna without any optimization technique then BBO, PSO and Combined PSO-BBO algorithm for optimized design of Yagi-Uda antenna.

5.3.1 Fitness Algorithm of Yagi-Uda Antenna

Followings are to step for fitness evaluation in NEC and C++ programming environment.

1. In first step, create a input text file with .nec extension.
2. In second step, add all commands and parameters to design particular antenna with specific parameters.
3. If file is created, then input file with .nec extension pass to nec2.exe, otherwise create correct input file as shown in fitness algorithm flow chart Fig 5.2 .
4. In next step, output text file is generated with .out extension.
5. Read all characteristics of an antenna that is required for optimization.

5.3.2 BBO Algorithm

Algorithmic flow for BBO is depicted in Fig. 5.3, and explained stepwise as follows:

1. In first step, identify SIVs and their universe of discourse (UODs).
2. In next step, create a habitat (string).
3. Then generate a random population.
4. Check for maximum iteration number arrived or not. If yes, select the best habitat and stop the BBO algorithm. If no, then evaluate fitness.
5. Check for fitness if achieved then select the best habitat and stop the BBO algorithm. If no, then apply migration.
6. After Migration, apply mutation.
7. If fitness is achieved then select the best habitat and stop the BBO algorithm. If no, then repeat the processes as shown in Fig 5.3.

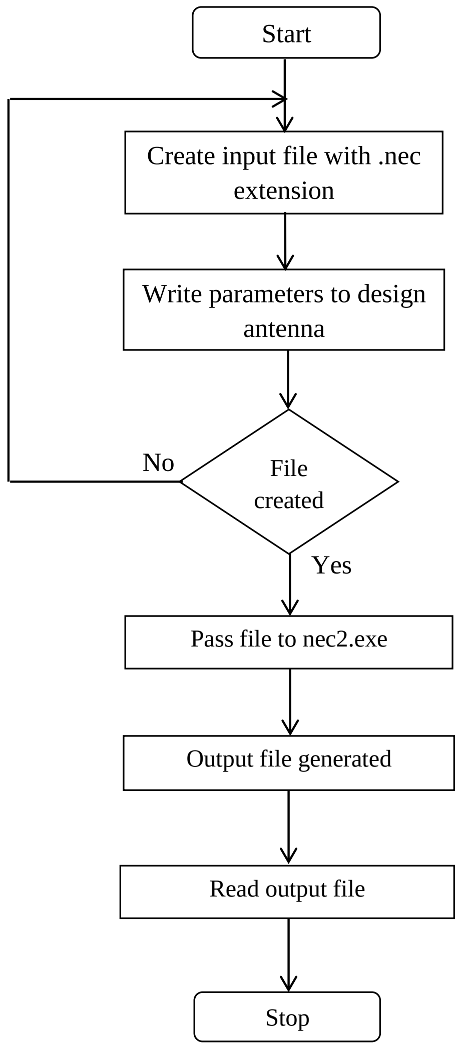

FIGURE 5.2: Fitness Evaluation Algorithm Flow Chart

5.3.3 PSO Algorithm

Algorithmic flow for PSO described below step wise and depicted in Fig. 5.4 for design optimization of Yagi-Uda antenna.

1. Initialize the population of particles at random positions and velocities. Assign present location and fitness as p_{id} and *pbest* to every particle as starting position and fitness, respectively.

2. For each particle, evaluate its fitness at the present position, x_i.

3. Compare the particle's fitness with *pbest*. If the current fitness value is better, copy it to *pbest* and set p_{id} equal to the current position, x_{id}.

4. Identify the most successful particle in the swarm and store it as p_{gd}.

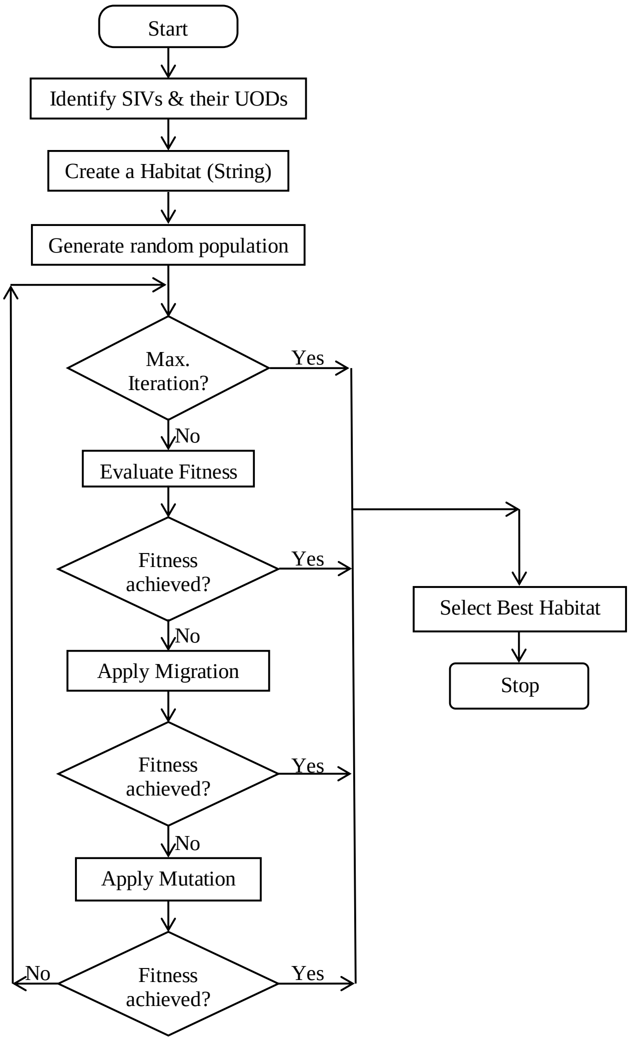

FIGURE 5.3: BBO Algorithm Flow Chart

5. Update the velocity and position of the particle using equations (5.1) and (5.2) [Shi and Eberhart, 1999]:

$$v_{id}^{m+1} = \chi(wv_{id}^{m} + \psi_1 r_1(p_{id}^{m} - x_{id}^{m}) + \psi_2 r_2(p_{gd}^{m} - x_{id}^{m})) \tag{5.1}$$

$$x_{id}^{m+1} = x_{id}^{m} + v_{id}^{m+1} \tag{5.2}$$

Here, w is inertia weight, ψ_1 is cognitive parameter, ψ_2 is social parameter and constriction factor χ are strategy parameters of PSO algorithm, while r_1, r_2 are random numbers uniformly distributed in the range [0,1].

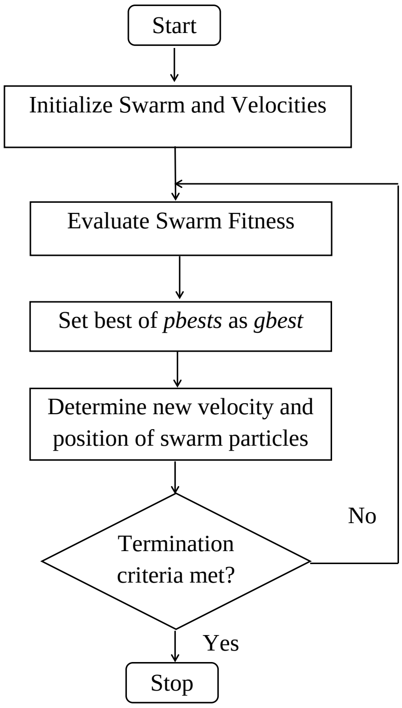

FIGURE 5.4: PSO Algorithm Flow Chart

5.3.4 Combined PSO-BBO Algorithm

To investigate faster convergence and evolve best results, PSO and BBO are experimented together to optimize same problem of antenna design. Here, PSO is made to run for initial pre-specified number of iterations and then BBO runs till end, however, number of maximum iterations is kept same. Algorithmic flow is depicted in Fig. 5.5 for design optimization of Yagi-Uda antenna.

1. In first step, identify SIVs and their universe of discourse (UODs). Here SIV's are $[L_1, L_2, \ldots, L_N, S_1, S_2, \ldots, S_{N-1}]$, where L is the length of N element yagi-uda antenna and S is the spacing between them.

2. Create habitats and initialize the random population of size NP.

3. Evaluate fitness of each habitat.

4. Update the population using PSO and BBO. Then check maximum iteration is done or not. If yes, select a best habitat and stop the BBO algorithm. If no, then evaluate fitness.

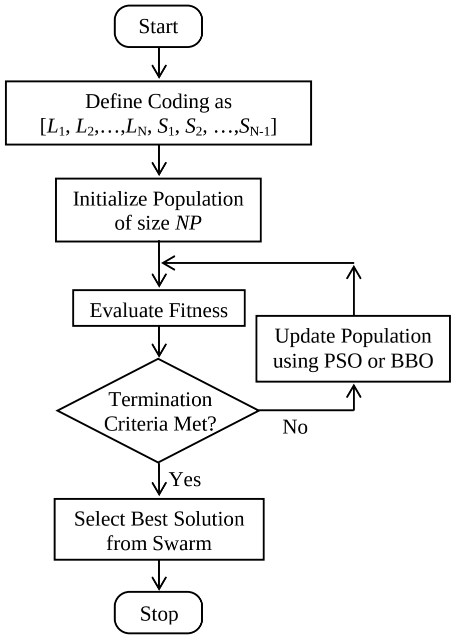

Figure 5.5: Combined PSO-BBO Algorithm Flow Chart

5.4 Conclusion

In this chapter, various implementation steps of algorithmic flow of fitness algorithm of Yagi-Uda antenna with NEC2 software and algorithmic flow of BBO, PSO and combined PSO-BBO are discussed for better understanding of work on both softwares (Qt Creator and NEC2). Simulation results of convergence performance of various stochastic algorithms using benchmark functions and 6-element Yagi-Uda antenna are represented in next chapter.

CHAPTER 6

SIMULATION RESULTS

This chapter presents average simulation results of 10 monte-carlo evolutionary runs to conclude convergence performance of various stochastic algorithms. Many simulation results are discussed in this chapter by using benchmark functions (Ackley, Dejong, Griewank, Rastrigin, Rosenbrock) and Yagi-Uda Antenna. Here, we use C++ programming platform for coding of various stochastic algorithms and NEC2, antenna modeling software, to determine antenna characteristics like gain and impedance etc..

6.1 Introduction

BBO is one of stochastic search algorithms therefore, require multiple run to present fair analysis. To improve the performance of BBO algorithm, various variants of BBO have been introduced like IRBBO, EBBO, Blended Migration. In this thesis, We introduced some new algorithms to get the faster results, i.e., the proposed algorithms are GE-BBO, GE-EBBO, GE-IRBBO. When GE-BBO is integrated with EBBO, it becomes GE-EBBO. When GE-BBO is integrated with Immigration Refusal then it becomes GE-IRBBO. Here, BBO & its variants, GE-BBO, GE-EBBO, GE-IRBBO are made to run for 10000 iterations using 20 habitats in each case. These algorithms are also experimented to design Yagi-Uda Antenna by optimizing gain, impedance etc. BBO and GE-BBO are made to run for 200 iterations using 20 and 30 habitats in each case. GE-EBBO is applied on Yagi-Uda antenna to optimize its wire lengths and spacings in between them to present better choice for faster convergence and made to run for 50 iterations with 30 habitats and their performances are presented in the ending sections of this chapter. PSO, BBO and Combined PSO-BBO are also made to run for 200 iterations with 30 habitats in each case. Every habitats involves 11 SIVs for evolution of six-wire Yagi-Uda antenna.

6.2 Simulation Platform

Six-wire Yagi-Uda antenna designs are optimized for gain using PSO, BBO, GE-BBO and combined PSO-BBO algorithms are investigated. Average of 10 monte-carlo evolutionary runs for each algorithm are plotted here for investigation. The C++ programming platform is used for coding of optimization algorithms, whereas, a method of moments based software named as NEC [Burke and Poggio, 1981] is used for evaluation of antenna designs. Each potential solution in BBO is encoded as vector with 11 SIVs as shown in Fig. 4.7. The universe of discourse for the search of optimum values of wire lengths and wire spacings are $0.40\lambda - 0.50\lambda$ and $0.10\lambda - 0.45\lambda$, respectively, however, cross sectional radius and segment sizes are kept same for all elements, i.e., 0.003397λ and 0.1λ respectively, where λ is the wavelength corresponding to frequency of operation, i.e, 300 MHz. Excitation is applied to the middle segment of driven element and location of middle segment of the reflector element is always kept at $x = 0$.

In this chapter, Firstly, Convergence comparison of various stochastic algorithms like BBO, IRBBO, EBBO, Blended Migration, GE-BBO, GE-EBBO, GE-IRBBO is done by testing on various benchmark functions. Secondly, Convergence Performance of gain optimization of six-element Yagi-Uda Antenna is experimented by BBO, GE-BBO. Thirdly, gain optimization of six-element Yagi-Uda Antenna is performed by BBO, EBBO, PSO and GE-EBBO. Fourthly, gain optimization of six-element Yagi-Uda Antenna is performed by PSO, BBO, Combined PSO-BBO.

6.3 Simulations on Benchmark Testbed

Various stochastic algorithms are tested by using benchmark functions. Stochastic algorithms are BBO, IRBBO, EBBO, Blended Migration, GE-BBO, GE-EBBO, GE-IRBBO. Convergence performance of various stochastic algorithms is investigated with high mutation on mediocre habitats, i.e., C = 1. Simulation parameters used here are:

1. Population size: 20
2. Number of SIV's: 10
3. Search space of f(x): $-2 \leqslant x \leqslant 2$

4. Number of Iterations: 10000

5. Mutation probability: 1%

6. Number of Monte-Carlo simulations per experiment: 10

7. No Elitism in Mutation

Benchmark Functions used are Ackley, Dejong, Griewank, Rastrigin, Rosenbrock. In almost all the cases GE-EBBO gives better results than BBO and its variants. But the Blended migration gives very poor and exceptional results, i.e., out of the range of graph. Thats why it is not shown in the graph. The evolutionary simulation results for convergence performance for each benchmark function are presented, systematically, one by one as follows:

6.3.1 Ackley

In case of ackley function, All stochastic algorithms are tested by using the ackley function from the Table 4.1. The performance of various algorithms is shown in Fig. 6.1 and is given in Table 6.1. By using ackley function, The best results are given by GE-EBBO. Rest other optimization techniques give results closer to the best one. EBBO gives poor results followed by IRBBO, BBO and Blended Migration.

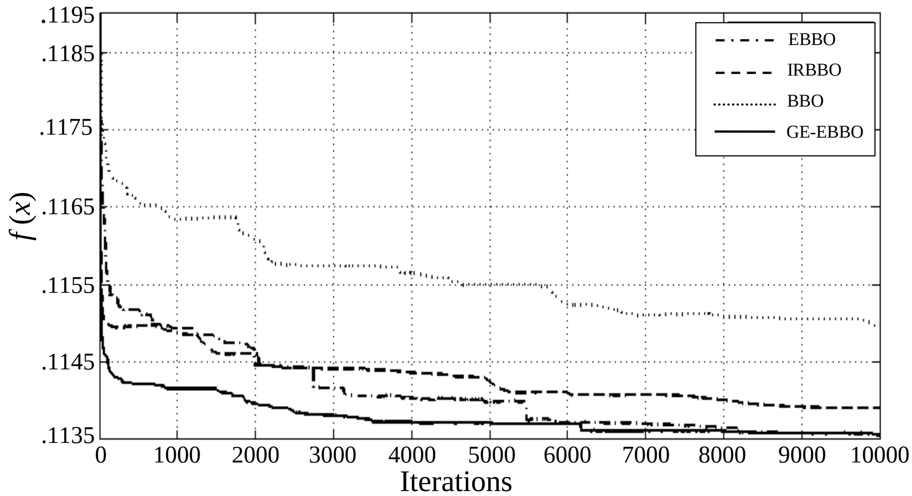

FIGURE 6.1: Convergence Comparison using Ackley Function

Fig. 6.2 shows the convergence comparison between BBO and GE-BBO using ackley function. Initially, GE-BBO converges faster, After then BBO gives the overall best results. The reason of bad performance of GE-BBO is the large possibilty of similar solutions. Because of high exploitation in GE-BBO.

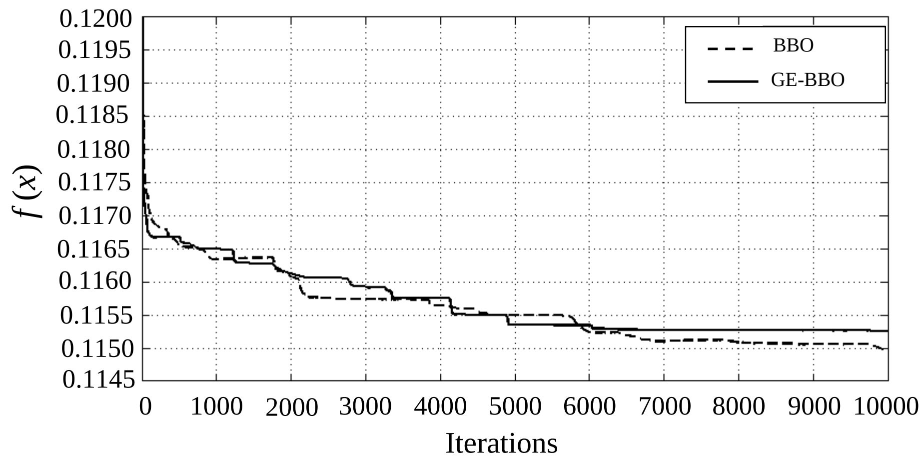

FIGURE 6.2: BBO versus GE-BBO using Ackley Function

Fig. 6.3 shows the convergence comparison between EBBO and GE-EBBO using ackley function. GE-EBBO performs overall best in this case. GE-EBBO performs better because of high exploitation on less fit habitats, whereas, less exploitation on high fit habitats and then the modified clear duplicate operator is further integrated to increase the diversity and exploration of the solution.

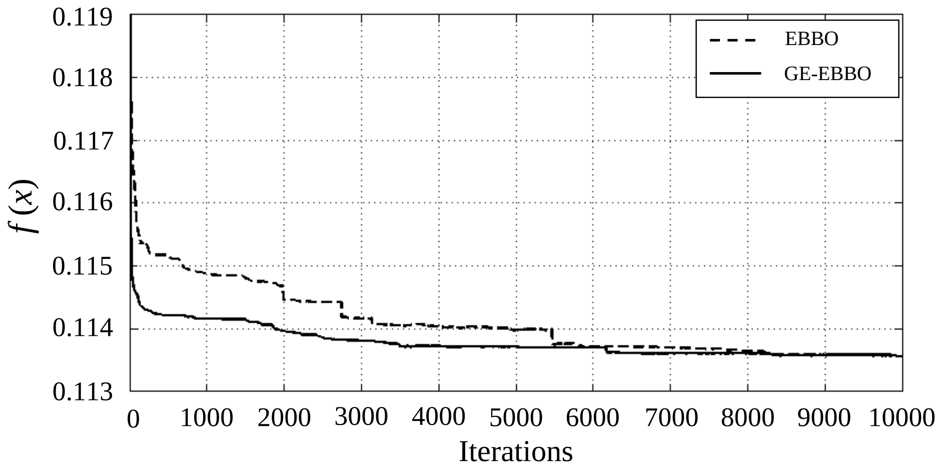

FIGURE 6.3: EBBO versus GE-EBBO using Ackley Function

Fig. 6.4 shows the convergence comparison between IRBBO and GE-IRBBO using ackley function. IRBBO performs better in this case, whereas, high exploitation in GE-IRBBO gives bad results instead of increasing the fitness of habitats.

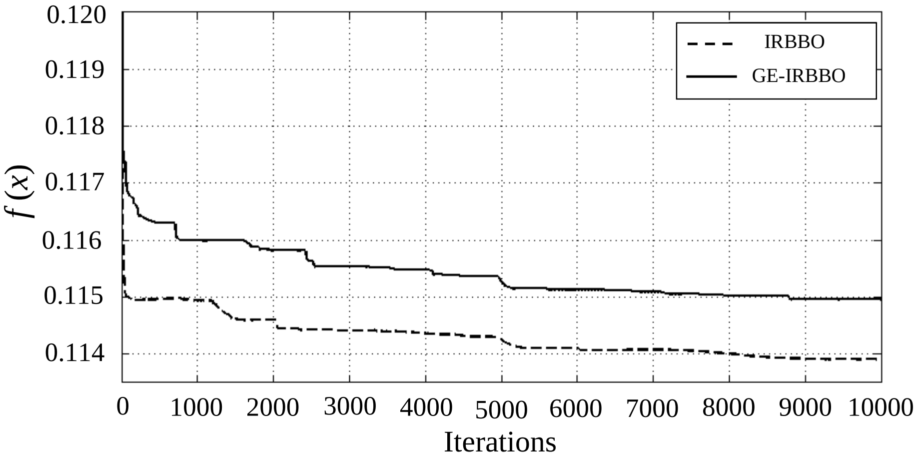

Figure 6.4: IRBBO versus GE-IRBBO using Ackley Function

6.3.2 Dejong

In case of dejong function, All stochastic algorithms are tested by using the dejong function from the Table 4.1. The performance of various algorithms is shown in Fig. 6.5 and is given in Table 6.1. By using dejong function, The best results are given by GE-EBBO. Rest other optimization techniques give results closer to the best one.

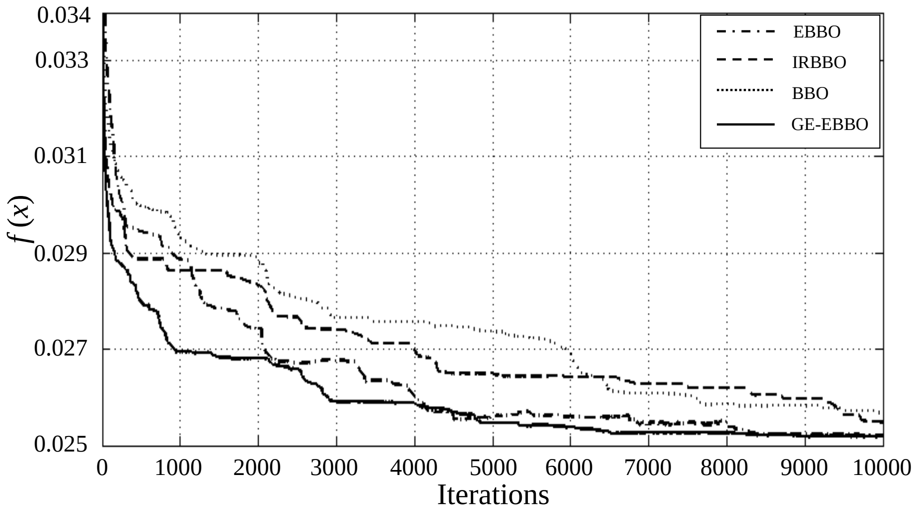

Figure 6.5: Convergence Comparison using Dejong Function

Fig. 6.6 shows the convergence comparison between BBO and GE-BBO using dejong function. BBO performs better in this case. GE-BBO gives bad results because of the large possibility of similar solutions.

Fig. 6.7 shows the convergence comparison between EBBO and GE-EBBO using dejong function. GE-EBBO performs better in this case. The possibility of similar solution is removed in EBBO and GE-EBBO by integrating modified clear duplicate operator.

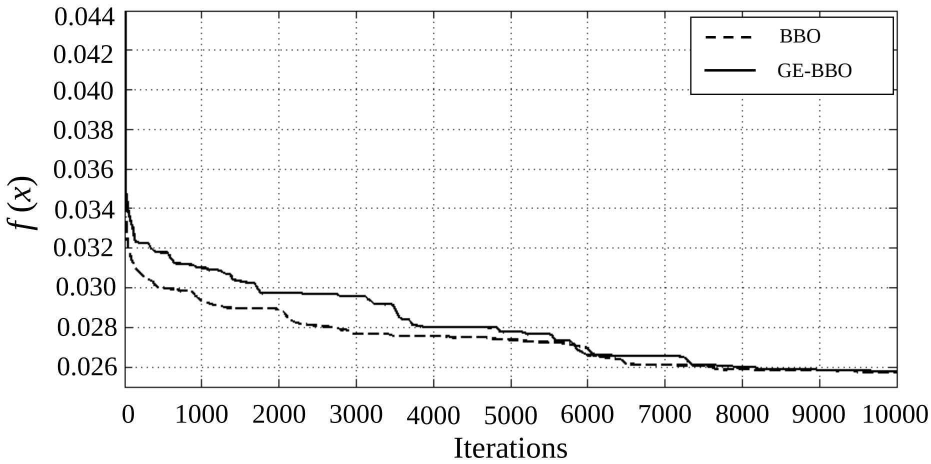

FIGURE 6.6: BBO versus GE-BBO using Dejong Function

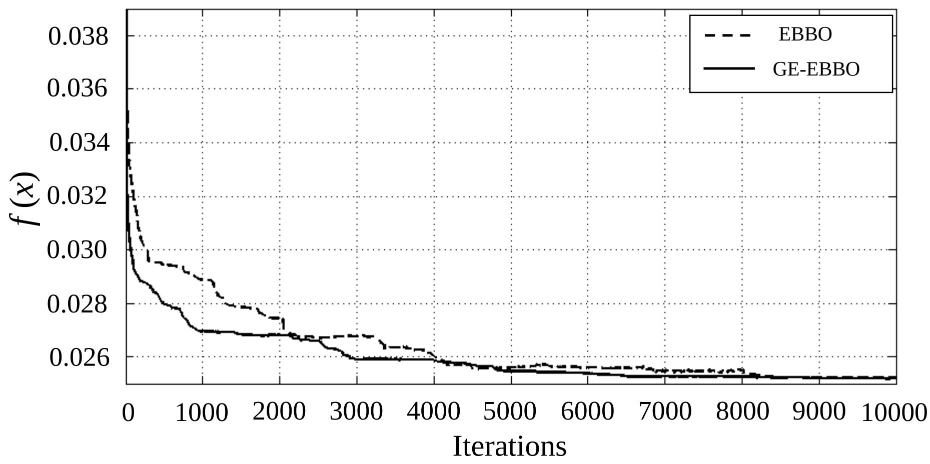

FIGURE 6.7: EBBO versus GE-EBBO using Dejong Function

Fig. 6.8 shows the convergence comparison between IRBBO and GE-IRBBO using dejong function. GE-IRBBO performs almost same like IRBBO in this case.

6.3.3 Griewank

In case of griewank function, All stochastic algorithms are tested by using the griewank function from the Table 4.1. The performance of various Algorithms is shown in Fig. 6.9 and is given in Table 6.1. By using griewank function, GE-EBBO and EBBO gives almost same results. Initially, EBBO converges faster. But with increase in iterations, GE-EBBO performs better than all other algorithms.

Fig. 6.10 shows the convergence comparison between BBO and GE-BBO using griewank function. BBO performs better in this case.

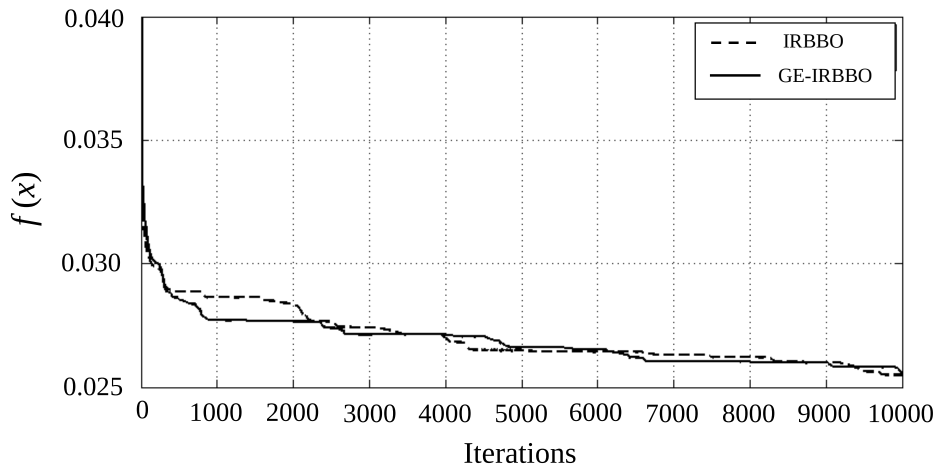

Figure 6.8: IRBBO versus GE-IRBBO using Dejong Function

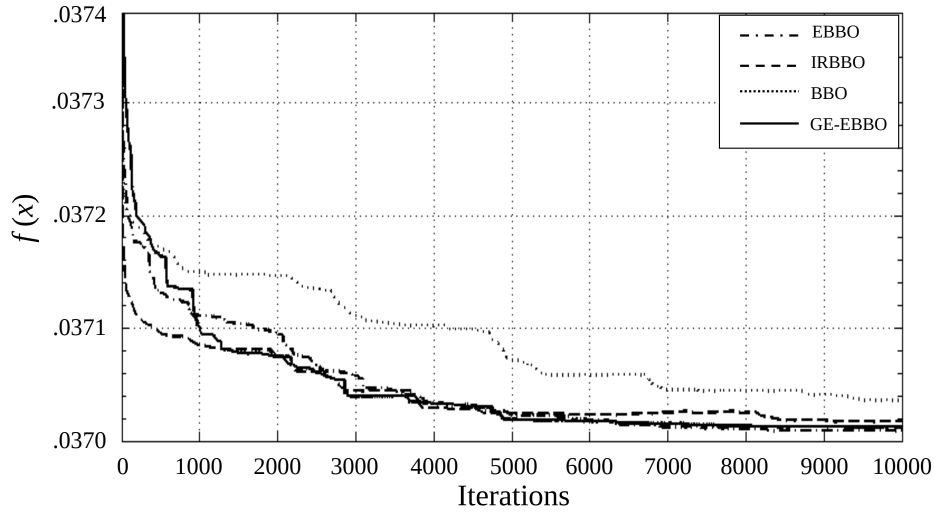

Figure 6.9: Convergence Comparison using Griewank Function

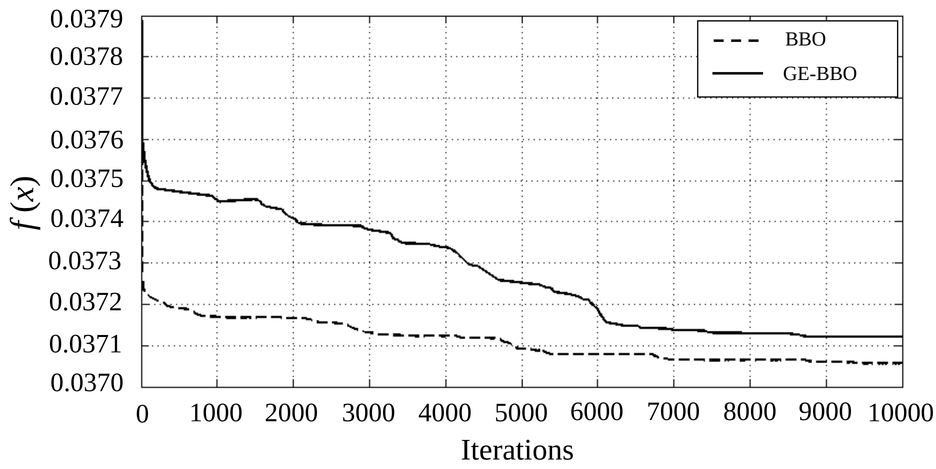

Figure 6.10: BBO versus GE-BBO using Griewank Function

Fig. 6.11 shows the convergence comparison between EBBO and GE-EBBO using griewank function. GE-EBBO performs better in this case.

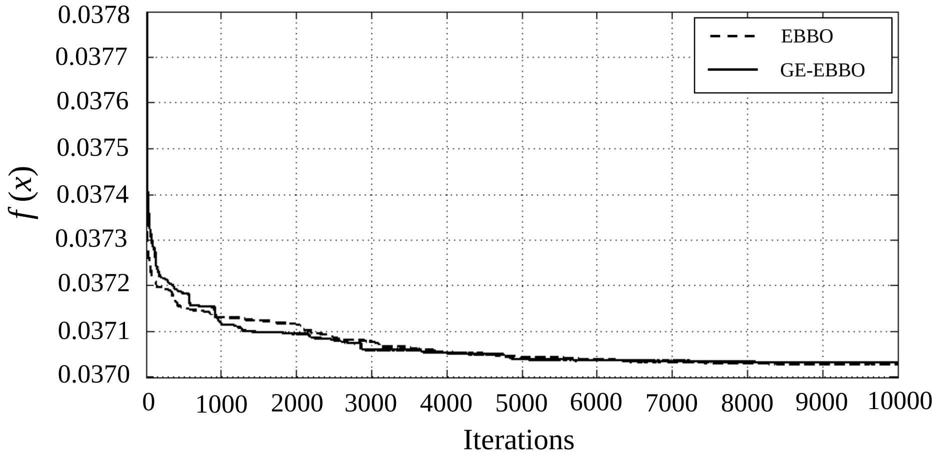

FIGURE 6.11: EBBO versus GE-EBBO using Griewank Function

Fig. 6.12 shows the convergence comparison between IRBBO and GE-IRBBO using griewank function. IRBBO performs better in this case.

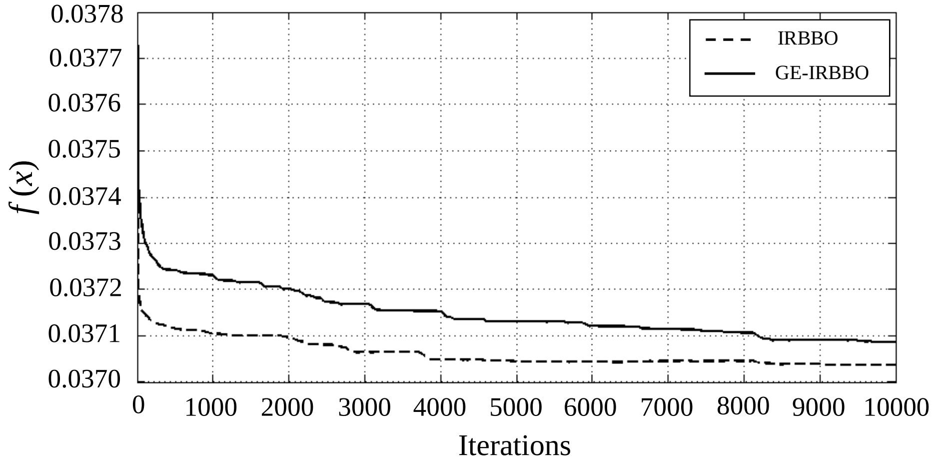

FIGURE 6.12: IRBBO versus GE-IRBBO using Griewank Function

6.3.4 Rastrigin

In case of rastrigin function, All stochastic algorithms are tested by using the rastrigin function from the Table 4.1. The performance of various Algorithms is shown in Fig. 6.13 and is given in Table 6.1. By using Rastrigin Function, The best results are given by GE-EBBO. initially, GE-EBBO converges slowly than other algorithms. But with increase in number of iterations, GE-EBBO performs better than all other algorithms.

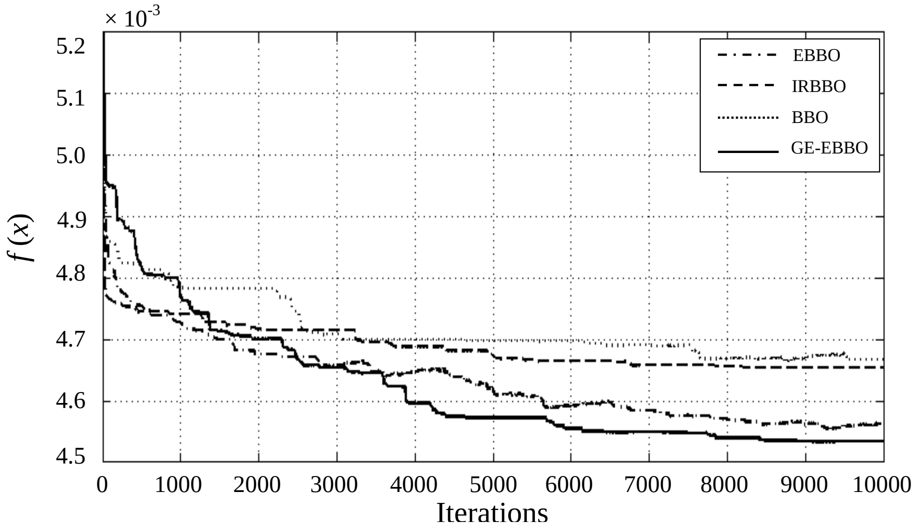

FIGURE 6.13: Convergence Comparison using Rastrigin Function

Fig. 6.14 shows the convergence comparison between BBO and GE-BBO using rastrigin function. BBO performs better in this case.

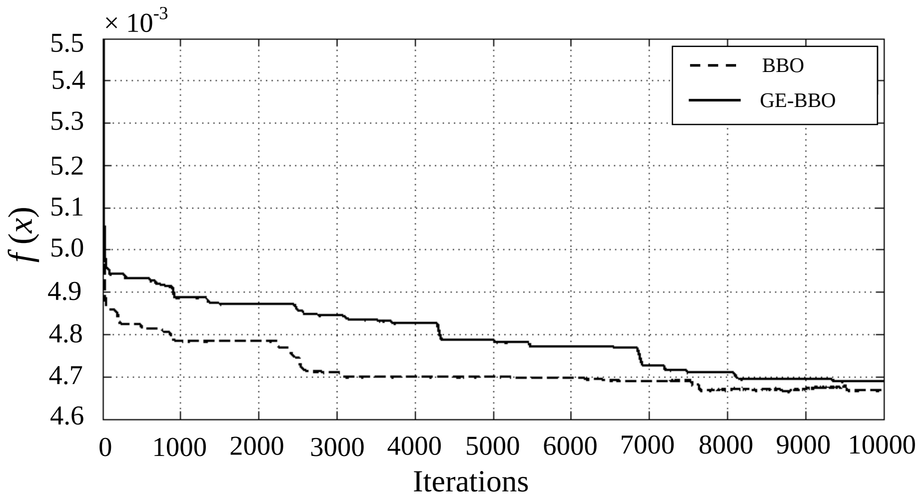

FIGURE 6.14: BBO versus GE-BBO using Rastrigin Function

Fig. 6.15 shows the convergence comparison between EBBO and GE-EBBO using rastrigin function. GE-EBBO performs better in this case.

Fig. 6.16 shows the convergence comparison between IRBBO and GE-IRBBO using rastrigin function. IRBBO performs better in this case.

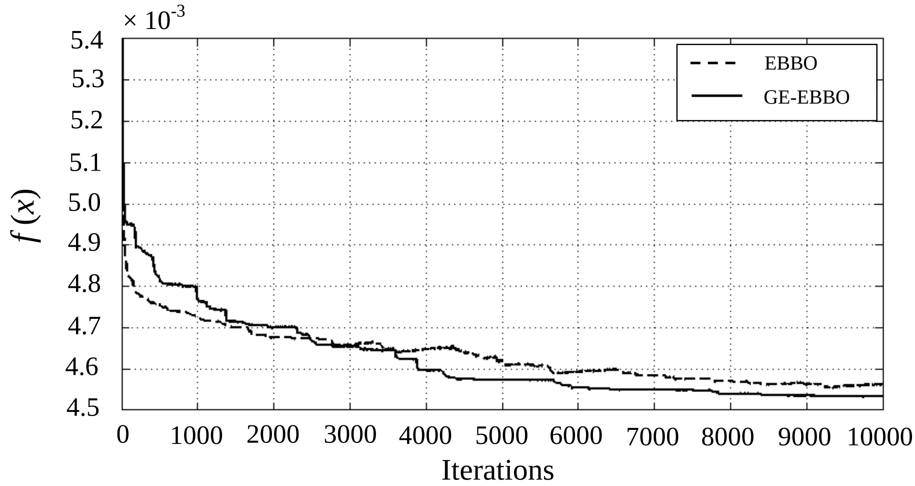

FIGURE 6.15: EBBO versus GE-EBBO using Rastrigin Function

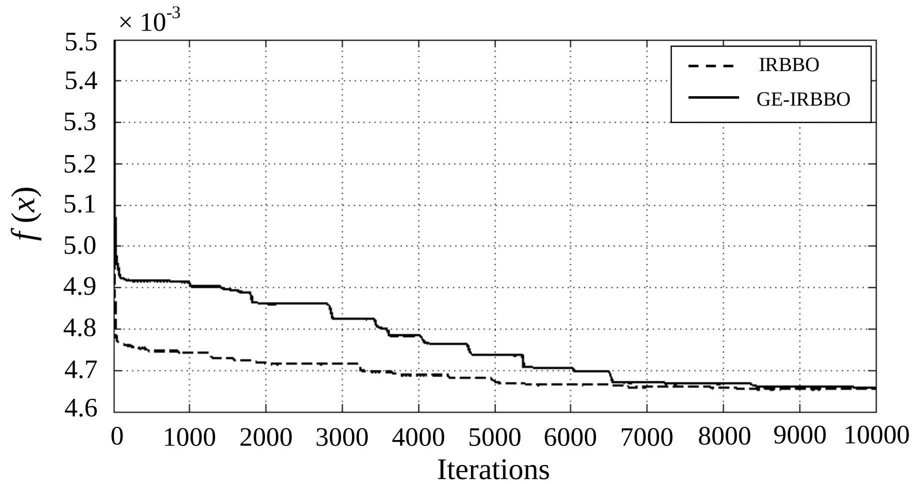

FIGURE 6.16: IRBBO versus GE-IRBBO using Rastrigin Function

6.3.5 Rosenbrock

In case of rosenbrock function, All stochastic algorithms are tested by using the rosenbrock function from the Table 4.1. The performance of various algorithms is shown in Fig. 6.17 and is given in Table 6.1. By using this function, The best results are given by IRBBO. At initial stage GE-EBBO does not perform upto the mark, but with increase in iterations at 3000-4000, the performance increases and approaches almost to the performance of IRBBO which gives best results.

Fig. 6.18 shows the convergence comparison between BBO and GE-BBO using rosenbrock function. BBO performs better in this case.

Fig. 6.19 shows the convergence comparison between EBBO and GE-EBBO using rosenbrock function. Performance of EBBO and GE-EBBO is almost same.

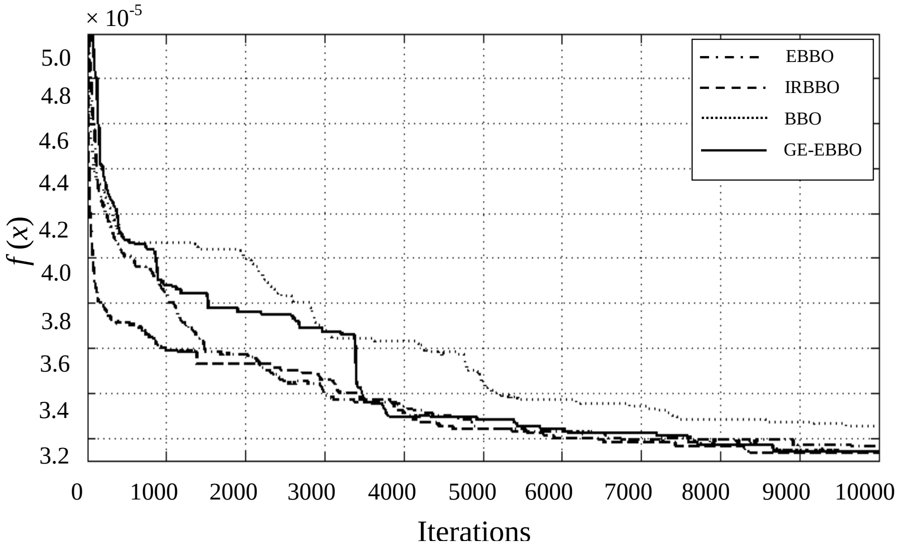

Figure 6.17: Convergence Comparison using Rosenbrock Function

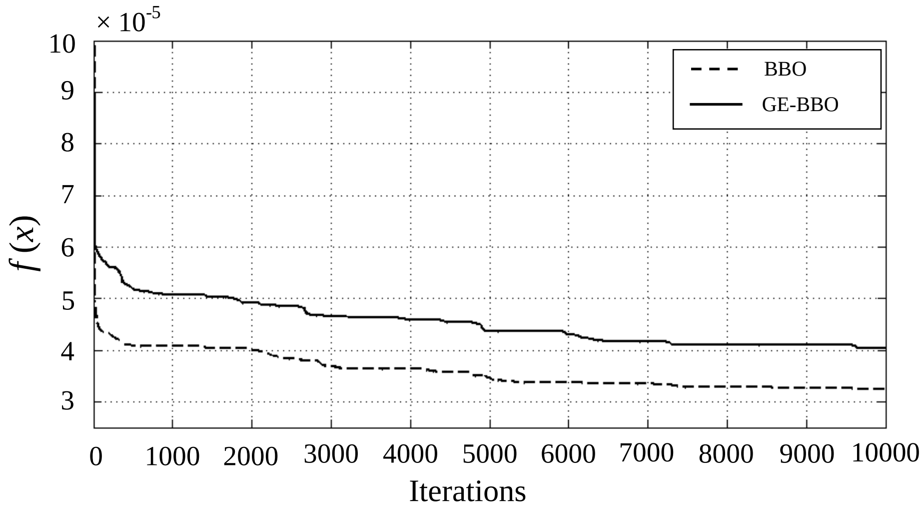

Figure 6.18: BBO versus GE-BBO using Rosenbrock Function

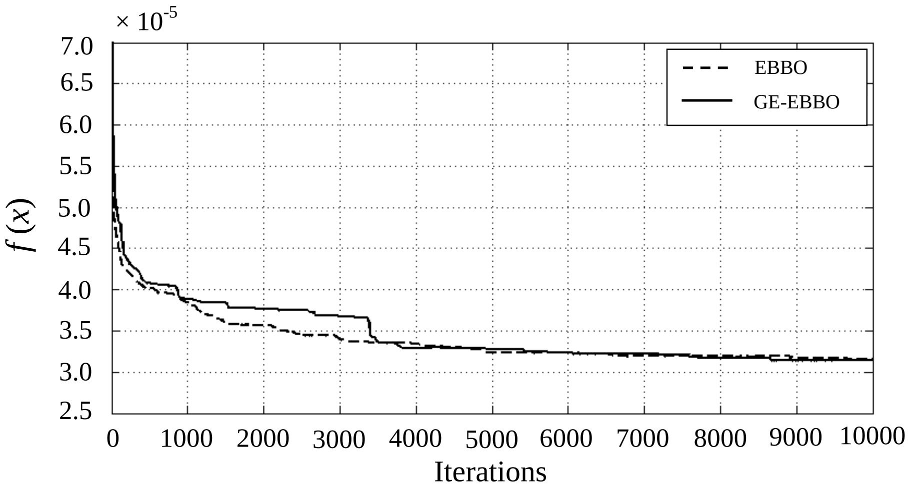

Figure 6.19: EBBO versus GE-EBBO using Rosenbrock Function

Fig. 6.20 shows the convergence comparison between IRBBO and GE-IRBBO using Rosenbrock function. IRBBO performs better in this case.

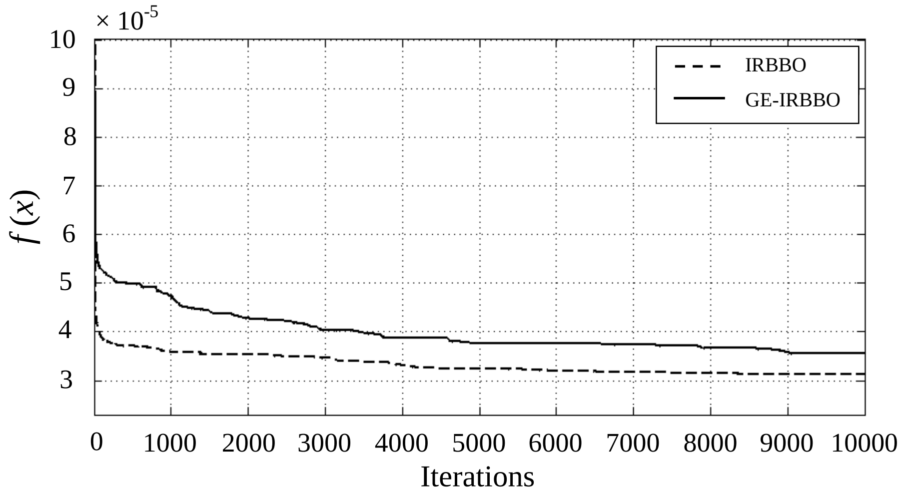

FIGURE 6.20: IRBBO versus GE-IRBBO using Rosenbrock Function

Test Function	BBO	GE-BBO	IRBBO	GE-IRBBO	EBBO	GE-EBBO	Blended
Ackley	0.1149567	0.1152590	0.1139057	0.1149593	0.1135618	**0.1135563**	0.1211875
Dejong	0.0256786	0.0257821	0.0254857	0.0255463	0.0252104	**0.0251785**	0.0286977
Griewank	0.0370556	0.0371213	0.0370368	0.0370861	**0.0370287**	0.0370322	0.0371918
Rastrigin	0.0046668	0.0046874	0.0046538	0.0046580	0.0045540	**0.0045334**	0.0052622
Rosenbrock	0.0000325	0.0000404	**0.0000313**	0.0000355	0.0000316	0.0000314	0.0000414
Average	0.0364780	0.0365779	0.0362226	0.0364569	0.0360773	**0.0360663**	0.0384761

TABLE 6.1: Comparison of various Stochastic Algorithms Convergence Performance using Benchmark Testbed

6.4 Simulation results using six-element Yagi-Uda Antenna

The evolutionary simulation results for convergence performance are presented, systematically, one by one as follows:

6.4.1 BBO versus GE-BBO

In case of BBO, standard migration is experimented for the gain optimization of six-wire Yagi-Uda Antenna. In case of GE-BBO, the good solution features contribute more towards the worse solution features to make them more fit and to make the convergence faster. GE-BBO is also experimented for the gain optimization of Yagi-Uda Antenna. Convergence performance of BBO and GE-BBO algorithms is investigated with high mutation on mediocre habitats, i.e., C = 3. Simulation parameters used here are:

1. Population size: 20 and 30
2. Number of SIV's: 11
3. UOD for wire length elements: $0.40\lambda - 0.50\lambda$
4. UOD for wire length spacings: $0.10\lambda - 0.45\lambda$
5. cross sectional radius: 0.003397λ
6. cross sectional segment size: 0.1λ
7. Mutation probability: 1%
8. Number of Iterations: 200
9. Number of Monte-Carlo simulations per experiment: 10
10. Elitism in Mutation

Experiments are done on both by using 20 habitats and 30 habitats. In both the cases as shown in the Fig. 6.21, 6.22 the convergence is faster in case of GE-BBO. In case of 20 habitat, The performance of GE-BBO is overall better than BBO, whereas, in case of 30 habitats, the convergence performance of GE-BBO is almost as BBO. The best gain optimization results obtained are given in the Tables 6.2, 6.3.

	GE-BBO		**BBO**	
Element	**Length**	**Spacing**	**Length**	**Spacing**
1(λ)	0.4851	-	0.4849	-
2(λ)	0.4955	0.1629	0.4625	0.1657
3(λ)	0.4380	0.2684	0.4408	0.2473
4(λ)	0.4249	0.3820	0.4256	0.3877
5(λ)	0.4181	0.4384	0.4200	0.4190
6(λ)	0.4295	0.3616	0.4272	0.3876
Gain(dBi)	**13.84**		**13.85**	
Z(Ω)	**5.37+j67.9**		**3.55+j15.48**	

Table 6.2: Convergence comparison of GE-BBO versus BBO using 20 habitat on six-element Yagi-Uda Antenna

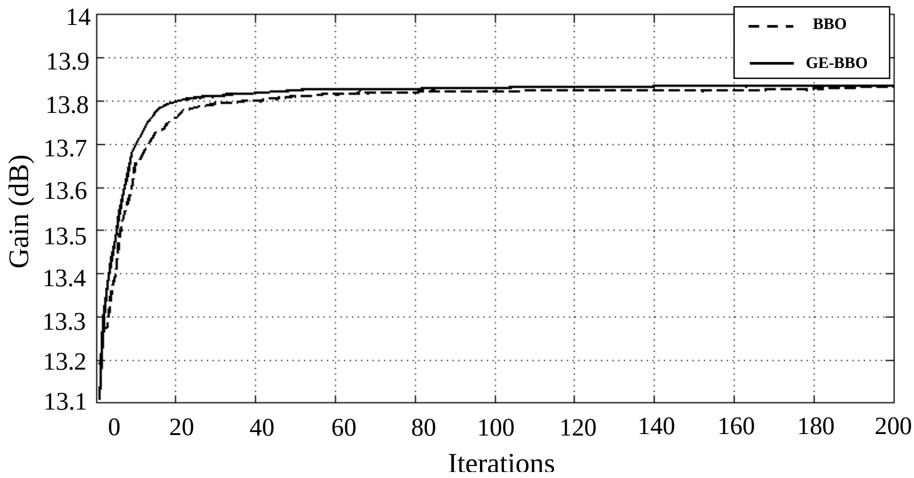

FIGURE 6.21: BBO versus GE-BBO using 20 habitats

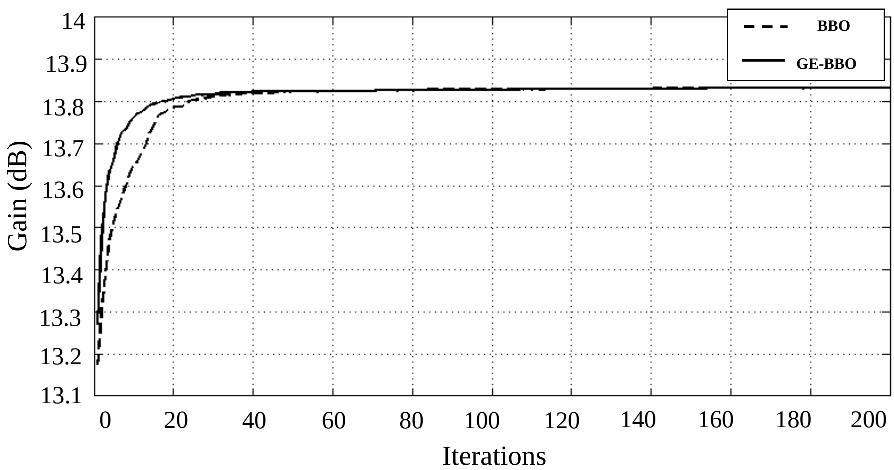

FIGURE 6.22: BBO versus GE-BBO using 30 habitats

	GE-BBO		**BBO**	
Element	**Length**	**Spacing**	**Length**	**Spacing**
1(λ)	0.4852	-	0.4841	-
2(λ)	0.4847	0.1808	0.4731	0.1745
3(λ)	0.4431	0.2256	0.4391	0.2561
4(λ)	0.4225	0.4181	0.4247	0.3986
5(λ)	0.4205	0.4071	0.4201	0.4060
6(λ)	0.4288	0.3769	0.4292	0.3786
Gain(dBi)	**13.84**		**13.84**	
Z(Ω)	**4.32+j48.54**		**4.81+j34.24**	

TABLE 6.3: Convergence comparison of GE-BBO versus BBO using 30 habitat on six-element Yagi-Uda Antenna

6.4.2 BBO, EBBO, PSO and GE-EBBO

In case of BBO, standard migration is experimented for the gain optimization of six-wire Yagi-Uda antenna. In case of EBBO, clear duplicate operator is used to increase the diversity over similarity in the population and to increase the exploration ability. GE-EBBO is based upon grading the potential habitats for migration and the concept of EBBO is incorporated to prevent similar solutions and to increase the diversity of newly generated solutions. The proposed algorithm GE-EBBO is applied on a six-element Yagi-Uda antenna to optimize its wire lengths and spacings in between them to present the better choice for faster convergence. Convergence performance of BBO, EBBO and GE-EBBO algorithms is investigated with high mutation on mediocre habitats, i.e., C=1. Simulation parameters used here are:

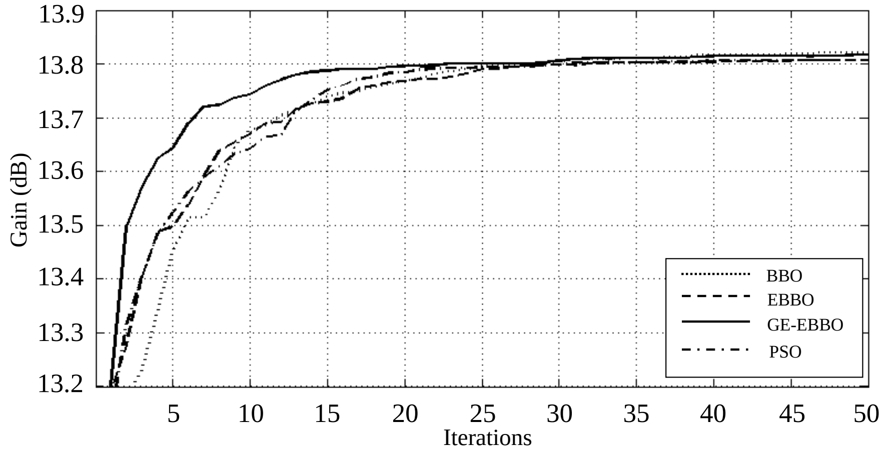

FIGURE 6.23: Convergence Performance of BBO, EBBO, PSO and GE-EBBO

1. Population size: 30
2. Number of SIV's: 11
3. UOD for wire length elements: $0.40\lambda - 0.50\lambda$
4. UOD for wire length spacings: $0.10\lambda - 0.45\lambda$
5. cross sectional radius: 0.003397λ
6. cross sectional segment size: 0.1λ
7. Mutation probability: 1%
8. Number of Iterations: 50
9. Number of Monte-Carlo simulations per experiment: 10
10. Elitism in Mutation

Experiments are done on 30 habitats. Fig. 6.23 depicts the convergence performance of BBO, EBBO, PSO and GE-EBBO algorithms. It can be observed that GE-EBBO is performing best among all optimization algorithms. EBBO also converges faster than BBO, but almost same like PSO's performance. The best gain optimization results are shown in the Table 6.4.

Stochastic algorithms	Element	1(λ)	2(λ)	3(λ)	4(λ)	5(λ)	6(λ)	Gain(dBi)	Z(Ω)
BBO	Length	0.4875	0.4884	0.4400	0.4233	0.4217	0.4233	**13.84**	**4.09+j54.53**
	Spacing	-	0.1503	0.2571	0.4087	0.3932	0.4095		
EBBO	Length	0.4827	0.4735	0.4424	0.4259	0.4201	0.4256	**13.83**	**5.23+j31.59**
	Spacing	-	0.2255	0.2160	0.3889	0.4181	0.3911		
GE-EBBO	Length	0.4842	0.4910	0.4425	0.4253	0.4181	0.4270	**13.84**	**4.47+j59.63**
	Spacing	-	0.1778	0.2381	0.3918	0.4212	0.3870		
PSO	Length	0.4872	0.4944	0.4423	0.4272	0.4194	0.4276	**13.85**	**3.83+j62.16**
	spacing	-	0.1597	0.2420	0.3857	0.4190	0.3841		

Table 6.4: The Best Results obtained using Gain Optimization by BBO, EBBO and GE-EBBO

6.4.3 PSO, BBO and Combined PSO-BBO

In case of BBO, standard migration, PSO, Combined PSO-BBO is experimented for the gain optimization of Yagi-Uda Antenna. To investigate faster convergence and evolve best results, PSO and BBO are experimented together to optimize same problem of antenna design. Here, PSO is made to run for initial pre-specified number of iterations and then BBO runs till end, however, number of maximum iterations is kept same, i.e., 200. Convergence performance of BBO, PSO and Combined PSO-BBO algorithms is investigated with high mutation on mediocre habitats, i.e., C = 3. Simulation parameters used here are:

1. Population size: 30
2. Number of SIV's: 11
3. UOD for wire length elements: $0.40\lambda - 0.50\lambda$
4. UOD for wire length spacings: $0.10\lambda - 0.45\lambda$
5. cross sectional radius: 0.003397λ
6. cross sectional segment size: 0.1λ
7. Mutation probability: 1%
8. Number of Iterations: 200
9. Number of Monte-Carlo simulations per experiment: 10
10. Elitism in Mutation

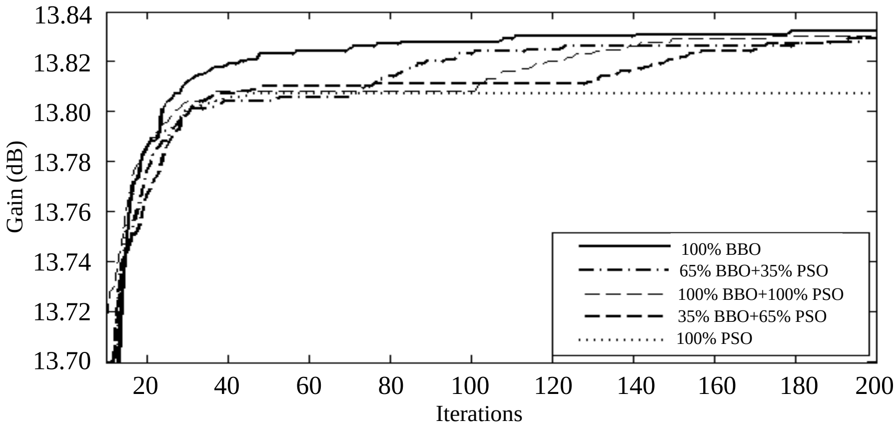

FIGURE 6.24: Convergence Performance of PSO and BBO

	PSO		**BBO**		**PSO-BBO**	
Element	**Length**	**Spacing**	**Length**	**Spacing**	**Length**	**Spacing**
1(λ)	0.4861	-	0.4838	-	0.4869	-
2(λ)	0.4791	0.1650	0.4728	0.1745	0.4941	0.1597
3(λ)	0.4426	0.2337	0.4388	0.2561	0.4421	0.2420
4(λ)	0.4239	0.4009	0.4244	0.3986	0.4269	0.3857
5(λ)	0.4203	0.4097	0.4198	0.4060	0.4191	0.4190
6(λ)	0.4252	0.3918	0.4289	0.3786	0.4273	0.3841
Gain(dBi)	**13.85**		**13.84**		**13.85**	
Z(Ω)	**3.58+j31.86**		**4.82+j34.24**		**3.83+j62.17**	

TABLE 6.5: The Best Results obtained using Gain Optimization by PSO and BBO

Fig. 6.24 depicts the convergence performance of PSO, BBO and combined PSO-BBO algorithms. It can be observed that BBO is performing best among all optimization algorithms. Further, PSO gives poor evolutionary results that are improved suddenly as and when BBO iterations start during combined PSO-BBO flow. The best gain optimization results are shown in the Table 6.5.

6.5 Conclusion

In this chapter, BBO & its variants are experimented and discussed with their graded versions by applying on various evaluation platforms to improve the performance of BBO algorithms. In the ending of section average results and best results are tabulated for overall evolution.

CHAPTER 7

CONCLUSION AND FUTURE SCOPE

Research is an iterative process very similar to BBO where researchers keep testing ideas based on their previous successes and the successes observed by other researchers in the area. The work in this thesis is no exception. Various research observations are presented at the end of each chapter as conclusions but limited to the scope of that chapter only. This chapter aims to conclude the thesis, as a whole, and to aggregate all the offshoots found throughout the work.

7.1 Introduction

The highlights of this thesis are:

1. We explore and test proposed algorithms on testbed of benchmark functions and compared them with various stochastic algorithms.
2. We explore GE-BBO and BBO to optimize gain of six-element Yagi-Uda antenna as single objective problem by varying element lengths and spacings between among elements.
3. We investigate GE-EBBO, BBO, EBBO and PSO for better convergence performance for design optimization of six-element Yagi-Uda antenna.

4. We investigate BBO, PSO and combined PSO-BBO for faster convergence performance as this technique is used for design optimization of six-element Yagi-Uda antenna.

Section 7.1 presents the concluding remarks about what has been investigated, developed, and contribution througout the work. In Section 7.3, various offshoots of the work are discussed as future research agenda.

7.2 Conclusion

In this thesis, various stochastic algorithms are investigated. Firstly on benchmark functions then on Yagi-Uda antenna for better convergence and gain optimization. Conclusions of this investigational study, as a whole are discussed as follows:

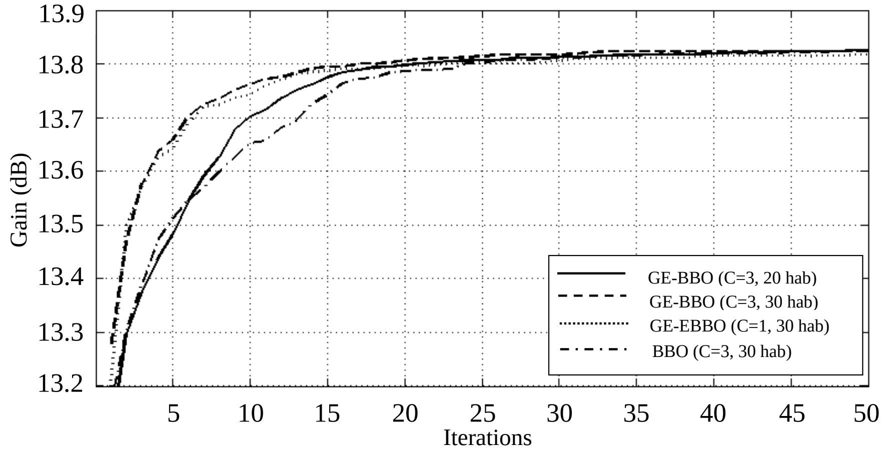

Figure 7.1: Best Convergence Performance using different Stochastic Algorithms

1. Among all the proposed algorithms, it can be observed that GE-EBBO performs best on almost all the benchmark functions. Because of high exploitation during migration, there is a large possibility of similar solutions, the concept of modified clear duplicate operator is incorporated in GE-EBBO to increase the diversity of newly generated solutions. Thats why GE-EBBO performs better.

2. From Yagi-Uda antenna simulation results, it can be observed that GE-BBO with C=3 on 20 and 30 habitats, GE-EBBO with C=1 on 30 habitats and BBO with C=3 on 30 habitats gives the best convergence performance by comparing with various stochastic algorithms as depicted in Fig. 7.1.

3. Maximum gain of Yagi-Uda antenna acheived during optimization using various stochastic algorithms is 13.85 dB that is better than that of reported in [Singh et al., 2010], i.e., 13.84 dB.

7.3 Future Agenda

Most of the times, a solution to a problem gives many issues to be investigated. The following remains on our future agenda.

1. In this thesis, We target single objective as gain of Yagi-Uda antenna using proposed algorithms. Here, Multiobjective optimization algorithms can be targeted for Gain, Impedance, etc.

2. All these stochastic algorithms can also be investigated to design other types of antenna like helical antenna, spida antenna, microstrip antenna, etc.

3. Investigations of some another real time problem using GE-EBBO can also be targeted.

4. Proposed graded algorithm can also be investigated by influencing the population size and search space to get better results.

5. Investigations of some problem using variant of PSO and comparing the performance with proposed algorithms can also be targeted.

REFERENCES

Altshuler, E. and Linden, D. (1997). Wire-antenna Designs using Genetic Algorithms. *Antennas and Propagation Magazine, IEEE*, 39(2):33–43.

Baskar, S., Alphones, A., Suganthan, P. N., and Liang, J. J. (2005). Design of Yagi-Uda Antennas using Comprehensive Learning Particle Swarm Optimisation. *IEEE*, 152(5):340–346.

Bojsen, J., Schjaer-Jacobsen, H., Nilsson, E., and Bach Andersen, J. (1971). Maximum Gain of Yagi–Uda Arrays. *Electronics Letters*, 7(18):531–532.

Burke, G. J. and Poggio, A. J. (1981). Numerical Electromagnetics Code (NEC) method of moments. *NOSC Tech. DocLawrence Livermore National Laboratory, Livermore, Calif, USA*, 116:1–131.

Chen, C. and Cheng, D. (1975). Optimum Element Lengths for Yagi-Uda Arrays. *IEEE Transactions on Antennas and Propagation,*, 23(1):8–15.

Cheng, D. and Chen, C. (1973). Optimum Element Spacings for Yagi-Uda Arrays. *IEEE Transactions on Antennas and Propagation,*, 21(5):615–623.

Cheng, D. K. (1971). Optimization Techniques for Antenna Arrays. *Proceedings of the IEEE*, 59(12):1664–1674.

Cheng, D. K. (1991). Gain Optimization for Yagi-Uda Arrays. *Antennas and Propagation Magazine, IEEE*, 33(3):42–46.

Correia, D., Soares, A. J. M., and Terada, M. A. B. (1999). Optimization of gain, impedance and bandwidth in Yagi-Uda Antennas using Genetic Algorithm. *IEEE*, 1:41–44.

Darwin, C. (1859). On the origins of species by means of natural selection. *London: Murray.*

Darwin, C. (1964). *On the Origin of Species: A Facsimile.* Harvard University Press.

Dieterich, J. and Hartke, B. (2012). Empirical review of standard benchmark functions using evolutionary global optimization. *arXiv preprint arXiv:1207.4318*.

Dorigo, M., Birattari, M., and Stutzle, T. (2006). Ant colony optimization. *Computational Intelligence Magazine, IEEE*, 1(4):28–39.

Du, D., Simon, D., and Ergezer, M. (2009). Biogeography-based Optimization Combined with Evolutionary Strategy and Immigration Refusal. *IEEE*, 1:997–1002.

Eberhart, R., Shi, Y., and Kennedy, J. (2001). *Swarm Intelligence*. Morgan Kaufmann Publisher.

Ehrenspeck, H. and Poehler, H. (1959). A New Method for Obtaining Maximum Gain from Yagi Antennas. *IRE Transactions on Antennas and Propagation,*, 7(4):379–386.

Fishenden, R. M. and Wiblin, E. R. (1949). Design of Yagi Aerials. *Proceedings of the IEE-Part III: Radio and Communication Engineering*, 96(39):5.

Hanski, I. and Simberloff, D. (1997). The metapopulation approach, its history, conceptual domain, and application to conservation. *Metapopulation biology*, (1):5–26.

Heppner, F. and Grenander, U. (1990). A stochastic nonlinear model for coordinated bird flocks. *AMERICAN ASSOCIATION FOR THE ADVANCEMENT OF SCIENCE, WASHINGTON, DC(USA). 1990.*

Jones, E. A. and Joines, W. T. (1997). Design of Yagi-Uda Antennas using Genetic Algorithms. *IEEE Transactions on Antennas and Propagation,*, 45(9):1386–1392.

Karaboga, D. and Basturk, B. (2007). A powerful and efficient algorithm for numerical function optimization: artificial bee colony (abc) algorithm. *Journal of Global Optimization*, 39(3):459–471.

Kennedy, J. and Eberhart, R. (1995). Particle swarm optimization. In *Neural Networks, 1995. Proceedings., IEEE International Conference on*, volume 4, pages 1942–1948. IEEE.

Kuwahara, Y. (2005). Multiobjective Optimization Design of Yagi-Uda Antenna. *IEEE Transactions on Antennas and Propagation,*, 53(6):1984–1992.

Li, J. Y. (2007). Optimizing Design of Antenna using Differential Evolution. *IEEE*, 1:1–4.

Ma, H. and Simon, D. (2010). Biogeography-based optimization with blended migration for constrained optimization problems. pages 417–418.

Ma, H. and Simon, D. (2011). Blended Biogeography-based Optimization for Constrained Optimization. *Engineering Applications of Artificial Intelligence*, 24(3):517–525.

MacArthur, R. and Wilson, E. (1967). *The Theory of Island Biogeography.* Princeton Univ Pr.

Mariani, V., Coelho, L., and Sahoo, P. (2011). Modified differential evolution approaches applied in exergoeconomic analysis and optimization of a cogeneration system. *Expert Systems with Applications*, 38(11):13886–13893.

Mendes, R. (2004). Population topologies and their influence in particle swarm performance.

Michalewicz, Z. (1998). Genetic algorithms+ data structures= evolution programs.

Pan, Q., Suganthan, P., Tasgetiren, M., and Liang, J. (2010). A self-adaptive global best harmony search algorithm for continuous optimization problems. *Applied Mathematics and Computation*, 216(3):830–848.

Parsopoulos, K. and Vrahatis, M. (2002). Recent Approaches to Global Optimization Problems through Particle Swarm Optimization. *Natural computing*, 1(2):235–306.

Pattnaik, S. S., Lohokare, M. R., and Devi, S. (2010). Enhanced Biogeography-Based Optimization using Modified Clear Duplicate Operator. *IEEE*, 1:715–720.

Rattan, M., Patterh, M. S., and Sohi, B. S. (2008). Optimization of Yagi-Uda Antenna using Simulated Annealing. *Journal of Electromagnetic Waves and Applications, 22*, 2(3):291–299.

Reid, D. G. (1946). The Gain of an Idealized Yagi Array. *Journal of the Institution of Electrical Engineers-Part IIIA: Radiolocation,*, 93(3):564–566.

Reynolds, C. W. (1987). Flocks, herds and schools: A distributed behavioral model. 21(4):25–34.

Shen, L. C. (1972). Directivity and Bandwidth of Single-band and Double-band Yagi Arrays. *IEEE Transactions on Antennas and Propagation,*, 20(6):778–780.

Shi, Y. and Eberhart, R. (1999). Empirical Study of Particle Swarm Optimization. In *Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on*, volume 3. IEEE.

Shi, Y. et al. (2001). Particle Swarm Optimization: Developments, Applications and Resources. In *Evolutionary Computation, 2001. Proceedings of the 2001 Congress on*, volume 1, pages 81–86. Ieee.

Simon, D. (2008). Biogeography-based Optimization. *IEEE Transactions on Evolutionary Computation,*, 12(6):702–713.

Singh, S., Mittal, E., and Sachdeva, G. (2012a). Nsbbo for gain-impedance optimization of yagi-uda antenna design. In *Information and Communication Technologies (WICT), 2012 World Congress on*, pages 856–860. IEEE.

Singh, S., Mittal, E., and Tayal, S. (2013a). Evolutionary performance comparison of bbo and pso variants for yagi-uda antenna gain maximization. In *National Conference on Contemporary Techniques & Technologies in Electronics Engineering.*

Singh, S. and Sachdeva, G. (2012a). Mutation effects on bbo evolution in optimizing yagi-uda antenna design. In *Third International Conference on Emerging Applications of Information Technology (EAIT 2012)*, Kolkata, India.

Singh, S. and Sachdeva, G. (2012b). Yagi-uda antenna design optimization for maximum gain using different bbo migration variants. *International Journal of Computer Applications*, 58(5):1–5.

Singh, S., Shivangna, and Mittal, E. (2013b). Performance of pso with different ranges for wireless sensor node localization. In *National Conference on Contemporary Techniques & Technologies in Electronics Engineering*, page Accepted, Murthal, Sonepat, India.

Singh, S., Shivangna, and Mittal, E. (2013c). Range based wireless sensor node localization using pso and bbo and its variants. In *International Conference on Communication Systems and Network Technologies*, pages 309–315. IEEE.

Singh, S., Shivangna, and Tayal, S. (2013d). Analysis of different ranges for wireless sensor node localization using pso and bbo and its variants. *International Journal of Computer Applications*, 63(22):31–37. Published by Foundation of Computer Science, New York, USA.

Singh, S., Tayal, S., Mittal, E., and Shivangna (2013e). Evolutionary performance of graded emigration in bbo for yagi-uda antenna design optimization. *CiiT International Journal of Programmable Device Circuit and Systems.* CiiT International Journal.

Singh, S., Tayal, S., and Sachdeva, G. (2012b). Evolutionary performance of bbo and pso algorithms for yagi-uda antenna design optimization. In *2012 World Congress on Information and Communication Technologies (WICT)*, pages 861–865. IEEE.

Singh, U., Kumar, H., and Kamal, T. S. (2010). Design of Yagi-Uda Antenna Using Biogeography Based Optimization. *IEEE Transactions on Antennas and Propagation,*, 58(10):3375–3379.

Singh, U., Rattan, M., Singh, N., and Patterh, M. S. (2007). Design of a Yagi-Uda Antenna by Simulated Annealing for Gain, Impedance and FBR. *IEEE*, 1:974–979.

Storn, R. and Price, K. (1997). Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization*, 11(4):341–359.

Tayal, S., Singh, S., and Sachdeva, G. (2013). Bbo algorithm with graded emigration for yagi-uda antenna gain optimization. *International Journal of Computer Information Systems and Industrial Management Applications*. IJCISIM.

Uda, S. and Mushiake, Y. (1954). *Yagi-Uda Antenna*. Research Institute of Electrical Communication, Tohoku University.

Venkatarayalu, N. and Ray, T. (2004). Optimum Design of Yagi-Uda Antennas Using Computational Intelligence. *IEEE Transactions on Antennas and Propagation,*, 52(7):1811–1818.

Venkatarayalu, N. V. and Ray, T. (2003). Single and Multi-Objective Design of Yagi-Uda Antennas using Computational Intelligence. *IEEE*, 2:1237–1242.

Wallace, A. R. (1876). *The geographical distribution of animals: With a study of the relations of living and extinct faunas as elucidating the past changes of the earth's surface*, volume 1. Cambridge University Press.

Wang, H. J., Man, K. F., Chan, C. H., and Luk, K. M. (2003). Optimization of Yagi array by Hierarchical Genetic Algorithms. *IEEE*, 1:91–94.

Whitley, D., Rana, S., Dzubera, J., and Mathias, K. (1996). Evaluating evolutionary algorithms. *Artificial intelligence*, 85(1):245–276.

Yagi, H. (1928). Beam Transmission of Ultra Short Waves. *Proceedings of the Institute of Radio Engineers*, 16(6):715–740.

Yao, X., Liu, Y., and Lin, G. (1999). Evolutionary programming made faster. *Evolutionary Computation, IEEE Transactions on*, 3(2):82–102.

Zhao, L., Qian, F., Yang, Y., Zeng, Y., and Su, H. (2010). Automatically extracting t–s fuzzy models using cooperative random learning particle swarm optimization. *Applied Soft Computing*, 10(3):938–944.

INDEX