M.Tech. Thesis

MULTIOBJECTIVE GAIN-IMPEDANCE OPTIMIZATION OF 6-WIRE YAGI-UDA ANTENNA

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

by

Etika Mittal (1169038)

Under the Supervision of

Dr. Satvir Singh

PUNJAB TECHNICAL UNIVERSITY

Jalandhar-Kapurthala Highway, Jalandhar

SHAHEED BHAGAT SINGH

STATE TECHNICAL CAMPUS

Moga Road (NH-95), Ferozepur-152004 (PB) INDIA

August 2013

CERTIFICATE

I, **Etika Mittal (1169038)**, hereby declare that the work being presented in this thesis on MULTIOBJECTIVE GAIN-IMPEDANCE OPTIMIZATION OF 6-WIRE YAGI-UDA ANTENNA 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 & Communication Engineering.

Etika Mittal (1169038)

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 do/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 M.Tech Viva-Voce Examination of Etika Mittal (1169038) is held at Department of ECE, SBS State Technical Campus, Ferozepur on ....

External Examiner
Name: ....

Dr. Satvir Singh
M.Tech. Coordinator, ECE

There are only two ways to live your life. One is as though nothing is a miracle. The other is as though everything is a miracle.

- Albert Einstein

Dedicated to

My Family & Guide

ACKNOWLEDGEMENTS

Apart from the efforts of myself, the success of Masters dissertation depends largely on the encouragement and guidelines of many others. I take this opportunity to express my gratitude to the people who have been instrumental in the successful completion of this project. I would like to express the deepest appreciation to my supervisor, **Dr. Satvir Singh**, Associate Professor, Department of Electronics & Communication Engineering , SBS State Technical Campus, Ferozepur (Punjab), India, who has the attitude and the substance of a genius: he continually and convincingly conveyed a spirit of adventure in regard to research and scholarship, and an excitement in regard to teaching. Without his guidance and persistent help this dissertation would not have been possible. I cant say thank you enough for his tremendous support and help. I feel motivated and encouraged every time I attend his meeting. Without his encouragement and guidance this project would not have materialized.

I am becoming increasingly present to the fact that research can indeed be an enjoyable and rewarding experience, despite the tedium and hardwork involved. This thesis is truly the culmination of his support, motivation, generous help and teachings. I can never forget the cheerful moments of my life when this charismatic personality accepted me as a research scholar. I must record my sincere gratitude to him for not only the great store-houses of knowledge he bestowed upon me but also for the chiseling and grooming I received in large measure in spheres of academic, professional and personal life. Without his constant chase and help, this work could not have taken this shape. I am pretty sure that his guidance would go a step beyond this thesis report and would be reflected in Doctorate Course and a couple of more publications of improved quality and of greater rigor and coverage, which I now look forward to.

Dr. Satvir Singh's mature research advice in the very initial stage never let me down in research throughout the longish period of research. I could learn the technique of organizing and writing quality research matter only because of his erudite teachings throughout the project. This in fact has left a permanent impression on my personality and written and verbal communication. I also express my great admiration and indebtedness for the manner in which he painstakingly carried out a thorough editing of our papers and the thesis, despite his overwhelming busy schedule and numerous responsibilities.

There are several other persons who made important contributions during this period. The guidance and support received from all the members who contributed and who are contributing to this project, was vital for the success of the project. I am grateful for their constant support and help.

First and foremost, I express my heartfelt gratitude to **Dr. Umesh K. Garg**, Associate Professor, Adesh Institue of Engineering & Technology, Faridkot (Punjab) for motivating and pushing me towards PG Course, otherwise, this qualification might have been rather late.

My sincere thanks to my teachers **Dr. Vishal Sharma**, Assistant Professor, Department of Electronics & Communication Engineering, SBS State Technical Campus, Ferozepur (Punjab) and **Dr. Tejeet Singh**, Associate Professor, Department of Mechanical Engineering, SBS State Technical Campus, Ferozepur (Punjab) who used to guide us with their inspiring lectures on research methodolgy and always boost our spirit towards research work.

My sincere thanks to **Dr. T. S. Sidhu**, Principal, SBS State Technical Campus, Ferozepur (Punjab) and to **Mr. Sanjeev Dewra**, Head, ECE Department, SBS State Technical Campus, Ferozepur (Punjab) and to **Mrs. Rajni**, Associate Professor, ECE Department, SBS State Technical Campus, Ferozepur (Punjab) for providing us the environment to learn.

I express my gratitude to **Mr. Sureh Kumar**, **Mr. Darshan Singh**, **Mr. Prafful**, Lab Staff at SBS State Technical Campus, Ferozepur for solving the Power problems and Wi-Fi problems occured during the course.

I wish to acknowledge the magnificent support I have received from my senior friend **Mr. Gagan Sachdeva** and my fellow friends **Ms. Shivangna Keer**, **Ms. Shelja Tayal** and my brother **Atul** in the form of useful discussions throughout this work.

My sincere thanks to my friend **Ms. Gurimandeep Kaur** for clearing my doubts in LaTeX and making my writing part easier.

I am very grateful to my younger brother **Diwakar** who used to rush to the market for my printouts even late night and gave up playing games at Laptop for my research work and my friends **Ms. Nini**, **Ms. Pawan Supreet Kaur**, **Ms. Gurjeet Kaur** and **Mr. Hunny Sharma** for always appreciating and encouraging me towards research.

There are several other persons who made important contributions during this period, **Ms. Amandeep Kaur**, **Ms. Amrita Tayagi**, **Ms. Deepika** who used to sit with me when I used to work alone till late.

Most profound regards to my mother **Smt. Meena Gupta** and my father **Sh. Anil Gupta**, who confined their needs all for my sake. Many a times, they took loan to meet my college expenses during my course of UG and PG studies; it is these sacrifices and constant blessings that kept me motivated and committed, 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: SBS STC Ferozepur
Date: August 31, 2013 Etika Mittal

LIST OF THESIS OUTCOMES

International Conferences Publications

1. S. Singh, E. Mittal and G. Sachdeva, "NSBBO for Gain-Impedance Optimization of Yagi-Uda Antenna Design", in *Proceedings of,* ***IEEE*** *World Congress on Information and Communication Technologies,* Trivandrum, India, 30 October - 2 November, 2012.

   [Online] http://ieeexplore.ieee.org.

2. S. Singh, Shivangna and E. Mittal, "Range Based Wireless Sensor Node Localization using PSO and BBO and its Variants", in *Proceedings of,* ***IEEE*** *International Conference On Communication Systems and Network Technologies,* Gwalior, India, 6-8 April, 2013.

   [Online] http://ieeexplore.ieee.org.

National Conferences Publications

1. S. Singh, E. Mittal and S. Tayal, "Evolutionary Performance Comparison Of BBO and PSO Variants for Yagi-Uda Antenna Gain Maximization", in *Proceedings of,* ***IEEE*** *National Conference on Contemporary Techniques & Technologies in Electronics Engineering,* Murthal, Sonepat, India, 13-14 March, 2013.

2. S. Singh, Shivangna and E. Mittal, "Performance of PSO with Different Ranges for Wireless Sensor Node Localization", in *Proceedings of,* ***IEEE*** *National Conference on Contemporary Techniques & Technologies in Electronics Engineering,* Murthal, Sonepat, India, 13-14 March, 2013.

International Journals Publications/Submissions

1. S. Singh, E. Mittal and G. Sachdeva, "Multiobjective Gain-Impedance Optimization of Yagi-Uda Antenna using NSBBO and NSPSO", in *International Journal of Computer Applications (IJCA)*.

   [Online] http://ijcaonline.org.

2. S. Singh, S. Tayal, E. Mittal and Shivangna, "Evolutionary Performance of Graded Emigration in BBO for Yagi-Uda Antenna Design Optimization", in *CiiT International Journal of Programmable Device Circuit and Systems*.

   [Online] http://www.ciitresearch.org.

3. S. Singh and E. Mittal, "Multiobjective Gain-Impedance Optimization of Yagi-Uda Antenna", in *International Journal of Network and Innovative Computing (JNIC)*. [Accepted]

Invited Talk

1. "Fitness Ranking for MOO", in *AICTE-TEQIP Sponsored 2 Week Faculty Development Programme on Nature Inspired Computational Intelligence*, SBS State Technical Campus, Ferozepur, Punjab, India, 6-17 May, 2013.

ABSTRACT

Multi-objective optimization is actually required to solve many real-world design or decision making problems, however, most of the time engineers do consider single objective and try to freeze less relevant parameters.

An antenna acts as an interface between free-space radiations and transmitter or receiver. Yagi-Uda antenna introduced in 1926 by H. Yagi and S.Uda acts as a directional antenna consisting of a driven element (a dipole or folded dipole) and additional parasitic elements (called reflector and directors). The reflector element is slightly longer than the driven dipole, whereas the directors can be more than one in numbers whose lengths decrease in the direction of radiation. Yagi-Uda antenna is one of the most popular antenna designs in VHF and UHF due to its constructional ease and high gain, typically greater than 10 dB.

Yagi-Uda antenna is, however, difficult to design as physical design parameters such as element lengths, spacings between adjacent elements, and diameter bear complex and non-linear relationships for gain, impedance and Side Lobe Level (SLL), etc. This antenna design problem, further, complicates as (i) the number of antenna elements are increased so as to achieve higher directional gain or (ii) number of objectives (gain, impedance, etc.,) are increased to meet real world requirements. Therefore, multi-objective optimization of the antenna has always been a catchy problem for researchers. Although, a lot of work is done in this domain, still scope of improvement is visible with modern heuristic of Swarm Intelligence.

This thesis is intended to propose Non-dominated Sorting Biogeography Based Optimization (NSBBO) and investigate for multi-objective optimization of six-element Yagi-Uda antenna

designs to achieve two objectives, viz. gain and impedance, simultaneously. However, designing a Yagi-Uda antenna involves determination of wire-lengths and their spacing which is highly complex and non-linear problem. If gain is intended to increase then imaginary part in impedance becomes significantly large.

The convergence performance of different variants of BBO algorithms reported, till date, are investigated in this thesis, along with Non-dominated Sorting (NS) approach for multi-objective optimization of Yagi-Uda antenna design parameters. NS Particle Swarm Optimization is also investigated for fair and comparative performance study.

BBO is one of the most recent population based stochastic optimization technique inspired from the science of biogeography, i.e., the study of distribution of biological species, over space and time. Similar to many other Evolutionary Algorithms (EAs), to evolve optimal solution to any problem, BBO involves two inherent activities, i.e., (i) *exploitation* of available solution features (species) is made to happen using process of migration among various potential solutions (habitats), (ii) *exploration* of new solution features occur due to mutation operator. As BBO has shown impressive performance over other EAs for single objective optimization, therefore, is proposed and considered here for multi-objective optimization.

PSO is another population based EA that is inspired from the flight patterns of bird flocks, which was mainly governed by three major concerns: (i) *collision avoidance*, (ii) *velocity matching* and (iii) *flock centering*. In nature, birds exhibit flocking/swarm behavior for maximizing protection from predator, gaining-food from a large effective search-space and finding mates for themselves. PSO uses a population of potential solutions called particles that are flown through the search-space with adaptable velocities. Each particle has memory and, therefore, is capable of remembering the ever visited best position in the search-space. The position corresponding to the past best fitness is known as *pbest* and the overall best amongst all particles in the population is called global best or *gbest*.

Since the introduction of BBO, in 2008, various BBO variants have been reported by many researchers intended towards improved convergence performance. Most significant three BBO variants, viz., (a) *Blended migration*, (b) *Immigration Refusal* and (c) *Enhanced Biogeography-Based Optimization* (EBBO), are considered here for comparative performances for multi-objective optimization. Similarly, variants of PSO involves (a) Gloabl Best *(gbest)* PSO, (b) Local Best *(lbest)* PSO. All these variants have been experimented for gain-impedance optimization of six-wire Yagi-Uda antenna design for multiple times, in this thesis, under identical conditions.

During simulations, the antenna designs are evolved 10 times for gain-impedance optimization with non-dominated sorting. Averages of all 10 Monte-Carlo evolutionary runs are presented for fair convergence investigation of stochastic natured BBO and PSO algorithms. C++ programming platform is used for coding of NSBBO and NSPSO algorithms, whereas, Method of Moments (MoM) based Numerical Electromagnetic Code (NEC2) freeware is used for evaluation of antenna designs for gain and impedance. During gain-impedance optimization of Yagi-Uda antenna, a maximum gain of 12.70 dBi and impedance of 50Ω is achieved using NSBBO that is better than that of reported in [Singh et al., 2010], i.e., 12.69 dBi.

This thesis is outlined as follow: Chapter 1 is devoted to introduction to M.Tech thesis that includes introduction to Research Topic, Motivation, Methodologies, Contributions, Research findings and Organization of thesis. State-of-the-art study of the historical research in optimizing Yagi-Uda antenna for single and multi-objectives using AI and non-AI techniques is represented in Chapter 2. Chapter 3, is devoted to multi-objective problems and methods used to find solution to such problems. Chapter 4 is dedicated to study of biogeography, literature background of BBO, algorithmic flow and BBO variants reported, till date. Chapter 5 is dedicated to study of PSO, literature background of PSO, algorithmic flow and PSO variants reported till date. In Chapter 6, various design parameters of Yagi-Uda antenna and radiation pattern are represented to formulate antenna design problem as optimization problem. In Chapter 7 firstly, NEC software is introduced to carry on simulation and analysis of electromagnetic behavior of various antenna designs. Secondly, implementation flow of NSBBO and NSPSO in C++ along with NEC2. Simulation results of convergence performance of NSBBO as well as NSPSO algorithm and its variants for maximization of gain and achieving an impedance of 50Ω and 75Ω, are represented in Chapter 8. Lastly, conclusion and future scope of this research are discussed in Chapter 9.

Place: Ferozepur　　　　Etika Mittal (1169038)

Date: August 31, 2013

ABBREVIATIONS

Abbreviations	Description
ACO	Ant Colony Optimization
AI	Artificial Intelligence
BBO	Biogeographical Based Optimization
CHI	Convex Hull of Individual Minima
CI	Computational Intelligence
CLPSO	Comprehensive Learning PSO
DM	Decision Maker
D	Dominated Solutions
DE	Differential Evolution
EBBO	Enhanced BBO
EA	Evolutionary Algorithm
EC	Evolutionary Computation
F/B ratio	Front to Back Ratio
GDE	Generalized Differential Evolution
GA	Genetic Algorithm
gbest	Global Best
GP	Goal Programming
GUI	Graphical User Interface

Abbreviations	**Description**
HSI	Habitat Suitability Index
hbest	Hybrid Best
IR BBO	Immigration Refusal BBO
lbest	Local Best
MBC	Meteor Burst Communication
MoM	Method of Moments
MOEAs	Multiobjective Evolutionary Algorithms
MOGA	Multiobjective Genetic Algorithm
MOO	Multiobjective Optimization
MOOP	Multi-objective Optimization Problem
NS	Nondominated Sorting
NSBBO	Nondominated Sorting BBO
NSGA	Nondominated Sorting Genetic Algorithm
NSPSO	Nondominated Sorting PSO
NBI	Normal Boundary Interaction
NPGA	Niched Pareto Genetic Algorithm
NEC	Numerical Electromagnetic Code
PF	Pareto Front
PO	Pareto Optimal
PS	Pareto Set
pbest	Previous Best
PSO	Particle Swarm Optimization
RF	Radio Frequency
rSLL	relative Side Lobe Level
SLL	Side Lobe Level
SA	Stimulated Annealing
SIV	Suitability Index Variable
SI	Swarm Intelligence
UOD	Universe of Discourse

NOTATIONS

Symbols	Description
p_g	Best particle in overall swarm
p_l	Best particle in local swarm
p_i	Best position of the i-th particle visited in the past
φ_1	Cognitive learning parameter
χ	Constriction factor
$\bar{y}$	Demand Level
δ	Decision Space
μ	Emigration rate
f_1	First fitness function to be optimized
λ	Immigration rate
α	Inertia weight Coefficient
L	Length of elements for Yagi-Uda Antenna
E	Maximum possible emigration rate
I	Maximum possible immigration rate
$mRate$	Mutation Rate
N	Number of Elements in Yagi-Uda antenna design
NP	Number of solutions in a swarm set
V_{max}	Particles movement with maximum velocity

Symbols	**Description**
V_{id}	Present velocity of i-th particle
r	Random Numbers
φ_3	Relative pull
f_2	Second fitness function to be optimized
φ_2	Social learning parameter
S	Spacing between elements for Yagi-Uda Antenna
m_{max}	User defined Parameter
w	Weight vector

LIST OF FIGURES

LIST OF TABLES

CONTENTS

CHAPTER 1

INTRODUCTION

This thesis presents investigational studies in multiobjective optimization using Biogeography Based Optimization (BBO) and Particle Swarm Optimization (PSO), and their use in optimal designing of Yagi-Uda antenna. This introductory chapter presents an overview of thesis. This includes introduction to research topic, motivation, methodologies, contributions, research findings and organization of thesis.

1.1 Introduction

Multiobjective optimization (also known as *multiobjective programming*, *vector optimization*, *multicriteria optimization*, *multiattribute optimization* or *Pareto optimization*) is an area of multiple criteria decision making, that is concerned with mathematical optimization problems involving more than one objective function to be optimized, simultaneously. Multiobjective optimization has been applied in many fields of science, including engineering, economics and logistics where optimal decisions need to be taken in the presence of trade-offs between two or more conflicting objectives. Minimizing weight while maximizing the strength of a particular component, and maximizing performance whilst minimizing fuel consumption and emission of pollutants of a vehicle are examples of multiobjective optimization problems involving two and three objectives, respectively. In practical problems, there can be more than three objectives.

An antenna forms interface between free space radiations and the transmitter or receiver. The choice of an antenna normally depends on many factors such as gain, bandwidth and directivity, etc. A high gain directional antenna is required where signals need to travel long distance, e.g., satellite-earth link. Our focus on this thesis is on low cost antennas, high gain and minimum impedance (targeted impedance) Yagi-Uda antenna. Since dipole and the folded dipole antennas cannot offer much needed gain and bandwidth, our attention is thus shifted to Yagi-Uda antenna and long-periodic dipole array antennas. Yagi-Uda antenna is difficult to design and optimize due to their numerous parasitic elements. 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.

BBO is a recent swarm based stochastic optimization technique inspired from the science of biogeography. In BBO, like other EAs, (a) the *exploitation* is made to happen using migration of solution features (species) to evolve optimal solution to any problem among various potential solutions (habitats), whereas, (b) the *exploration* of new solution features occurs due to mutation operator.

The Particle Swarm Optimization (PSO) technique is based on the concept that individuals refine their knowledge about the search space through social interactions. Social norms emerge as individuals tend to imitate their successful peers. Each individual juggles two antagonistic concepts: *individualism* - going back to the best solution found so far, and *conformism* - imitating the best solution found in the whole swarm or its neighborhood. PSO is an optimization technique in the area of SI that is endowed with several advantages, viz. (i) easy to describe, (ii) simple to implement, (iii) few parameters to adjust, (iv) uses a relatively small population, (v) needs a relatively small number of function evaluations to converge, and (vi) fast. These advantages have given it increasing popularity, over other Evolutionary Algorithms (EAs), in the field of numeric optimization. It can be applied, like other EAs, in the areas of system modeling, multi-objective optimization, classification, pattern recognition, load/job scheduling, signal processing, robotic applications, and decision making, etc.

This thesis dvels into various aspects of NS approach and BBO and PSO algorithms, and then presents their collective use to solve Yagi-Uda Antenna design problems where gain and impedance are to be optimized simultaneously.

1.2 Motivation

Artificial Intelligence (AI) is not too old paradigm that presents the scope for better design with non-traditional design. BBO in [Simon, 2008] is recently introduced population based optimization technique proposed by D. Simon. As it is introduced in 2008 and shows better results than other optimization techniques [Baskar et al., 2005; Jones and Joines, 1997; Rattan et al., 2008; Venkatarayalu and Ray, 2003] scope of better and multiobjective optimization of Yagi-Uda antenna are visible with BBO and PSO.

Yagi-Uda antenna introduced in 1926 by H. Yagi and S. Uda. Calculation of gain and impedance of Yagi-Uda antenna has always been a problem for researchers due to number of element lengths and spacings between them. There is no straight forward formula to calculate gain and parameters of Yagi-Uda antenna. A lot of work is already done in this domain still scope of improvement is visible with modern meta heuristics. BBO may be investigated for more than two-objective optimization and parallel computing as well.

1.3 Objectives

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

1. To study Multiobjective optimization approach to optimize two conflicting parameters simultaneously.
2. To study parameter characterization of Yagi-Uda antenna for maximum gain and minimum impedance that can be obtaineded for six-element Yagi-Uda antenna.
3. Investigation in BBO algorithm with an application of multiobjective optimization of Yagi-Uda antenna.
4. Performance comparison of migration variants of BBO for Yagi-Uda antenna.
5. Investigation in PSO algorithm with an application of multiobjective optimization of Yagi-Uda antenna.
6. Performance comparison for algorithmic variants of PSO for Yagi-Uda antenna.

1.4 Methodology

The methodology followed in this thesis is:

1. To find out gain, impedance and other parameter characterization, Method of Moments (MoM) based software, called Numerical Electromagnetics Code (NEC-2) will be studied.
2. A programming platform of QT Creator will be created and reviewed for developing multiobjective BBO and PSO algorithms.

3. The work carried out in [Singh et al., 2010] for designing Yagi-Uda antenna using BBO with constrained optimization will be done using non dominated sorting in C++ environment.

4. Then variants of BBO and PSO reported till date will be studied and experimented for multiobjective optimization solutions.

1.5 Contributions

The main contributions of this report are:

1. To study Nondominated sorting approach for solving multiobjective optimization problem.

2. To study Yagi-Uda antenna for design issues and their performance parameters.

3. To propose NSBBO and create NSPSO algorithms on QT Creator platform for C++ code to optimize various multiple parameters of Yagi-Uda antenna.

4. To explore various algorithmic variants of BBO and PSO for improved convergence performance.

5. To develop QT Creator with NEC-2 for optimizing design of Yagi-Uda antenna using NSBBO and NSPSO.

1.6 Thesis Outline

After the brief introduction to M.Tech thesis given in Chapter 1, detailed study of the historical research in optimizing, both single objective and multi-objective, Yagi-Uda antenna using AI and non-AI techniques reported till date is represented in Chapter 2.

Chapter 3 is devoted to multiobjective problems, classical approaches used to solve these problems, ranking selection method used to find a pareto-optimal solution to these set of problems.

Chapter 4 is dedicated to study of biogeography, literature of BBO, BBO algorithms flow and its variants reported till date.

Chapter 5 is dedicated to study of particle swarm optimization, literature of PSO, PSO algorithms flow and its variants reported till date.

Chapter 6 devotes understanding of various design parameters of Yagi-Uda antenna. Here, radiation pattern obtained during simulation results are also represented for better understanding.

Chapter 7, Firstly, NEC software besed on Method of Moments (MoMs), for simulation and analysis of electromagnetic behavior of various antenna is discussed. Secondly, implementation flow of NSBBO and NSPSO in C-environment along with NEC-2 is discussed, and a brief introduction to NEC environment is also presented in this chapter.

Chapter 8 represents simulation results of convergence performance of NSBBO and NSPSO algorithms and its variants for optimization of gain and impedance of Yagi-Uda antenna. Best results in tabulated form are also represented in this chapter.

Lastly, conclusion and future scopes of this research are discussed in Chapter 9.

CHAPTER 2

LITERATURE SURVEY

The needed detailed literature survey, to get preliminary knowledge and search scope of investigation, to design Yagi-Uda antenna for multiobjective optimization of its various characterstics, i.e., gain, impedance, etc., simultaneously, is explained in this chapter. The principal concepts & developments that occurred till date, in these domains of research are presented in this chapter with a case built up for taking investigations that this thesis does.

2.1 Optimization

The goal of an optimization problem is to find a value for every variable that satisfies the constraints and maximizes/minimizes the objective function. The transformation of a maximization problem into a minimization one is straightforward. Heuristically, any optimization problem can be represented as a tuple of following three components:

1. A set of variables and respective domains.
2. A set of constraints which must be respected by candidate solutions.
3. A fitness function that measures the quality of each solution.

When solving an optimization problem, the goal is to find the global optimal solution in an acceptable amount of time. There are different methods capable of finding solutions to optimization problems which can be divided in several classes, as shown in Figure 2.1.

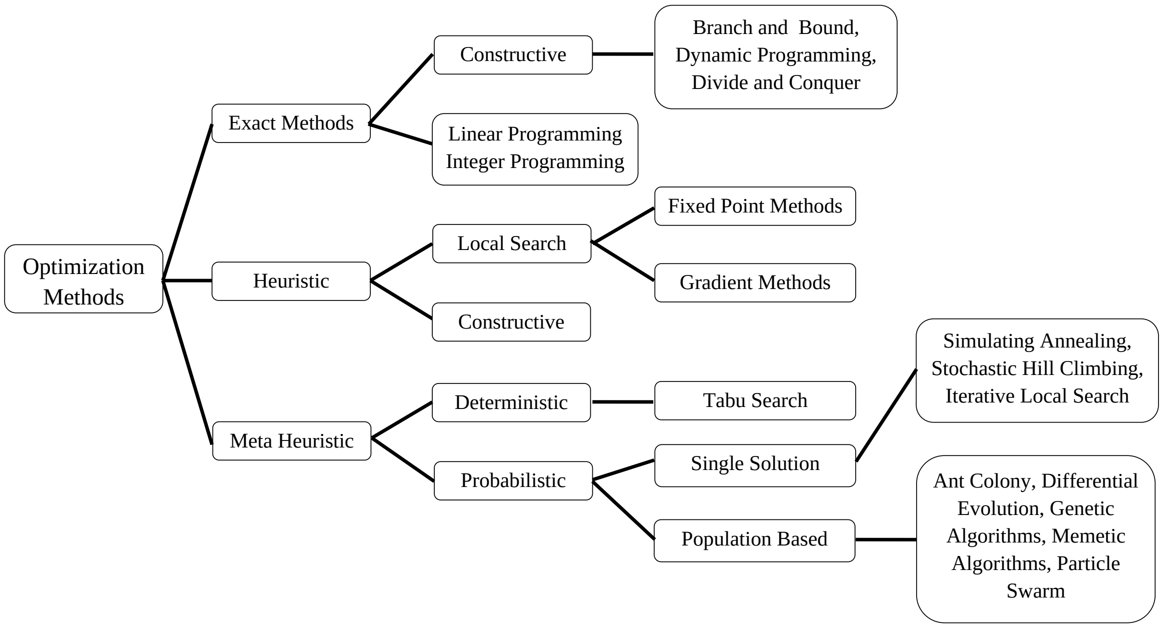


Figure 2.1: Optimization Methods

2.2 Multiobjective Optimization

As most optimization problems are multi objective to there nature, there are many methods available to tackle these kind of problems. Generally, the MOOP can be handled in four different ways depending on when the decision-maker articulates his or her preference on the different objectives, never, before, during or after the actual optimization procedure. These different possibilities are shown in Figure 2.2.

Although the classification gives a far from complete description of all available optimization techniques it constitutes a good frame work for discussing the most common methods suitable for engineering optimization.

2.2.1 No Preference Articulation

These types of methods do not use any preference information. Examples are the Min-Max formulation and global criterion method [Hwang et al., 1980; Osyczka, 1984; Steuer, 1986].

2.2.2 Prior articulation of preference information

The most common way of conducting multiobjective optimization is by prior articulation of the decision makers preferences. This means that before the actual optimization is conducted the different objectives are some how aggregated to one single figure of merit. This can be done in many ways, some of which are described here:

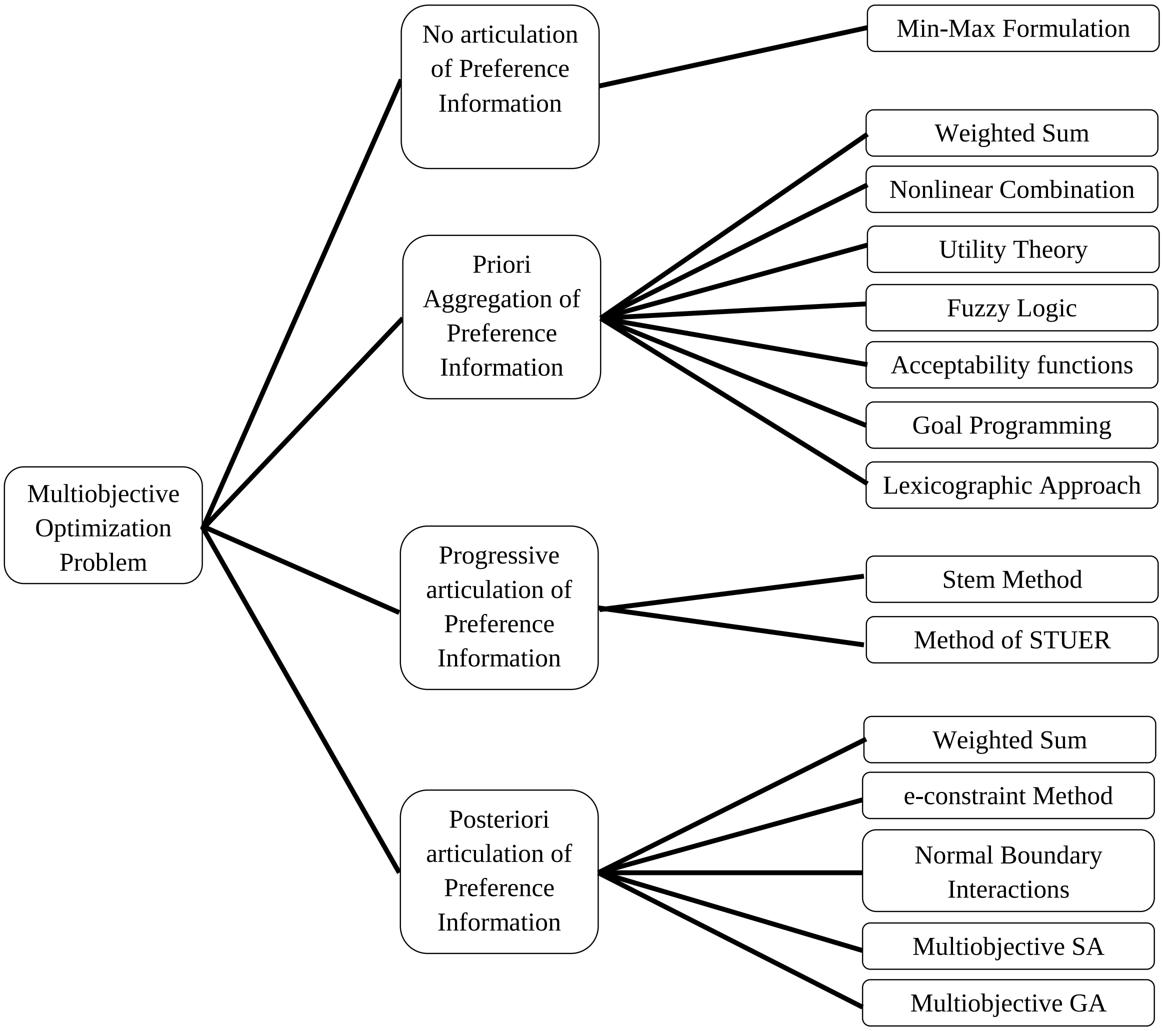


Figure 2.2: Classification of some methods of MOO

1. Weighted-sum approaches: The most easy and perhaps most widely used method is the weighted-sum approach discussed by Steuer in [Steuer, 1986].

2. Non-linear approaches: Anderson *et al.* employed this formulation as well as the House of Quality method in order to capture the preferences of a set of decision-makers in [Andersson et al., 1998].

3. Fuzzy logic approaches: The concept of fuzzy sets is based on a multi-valued logic where a statement could be simultaneously partly true and partly false. In fuzzy logic, a *membership* function μ expresses the degree of truthfulness of a statement, in the range from $\mu = 0$, indicating that the statement is false to $\mu = 1$ for truth. This is in opposite to binary logic where a statement can be only false or true. Examples of fuzzy approaches to multiobjective optimization could be found in [Chiampi et al., 1996; Chiampi M. and R., 1998; Zimmermann and Sebastian, 1995].

4. Utility theory: Utility theory forms the basics of decision making and dates back to Von Neumann and Morgenstern (1947), although the basic reference can be found in [Raiffa and Keeney, 1976]. Utility Theory is mathematically very rigorous. Once the right utility functions has been determined and the assumption holds true, it could be guaranteed that the solution found is the one with highest value to the decision-maker. However, deriving the individual utility functions and aggregating the overall utility function is hard for a simple problem, and for a complex multi-attribute problem, it might even be impossible. There are however examples on how utility theory is employed to solve design problems, [Thurston and Liu, 1991].

5. Acceptability functions: The method of acceptability functions has been developed by Wallace *et al.* in [Wallace et al., 1996] and Kim and Wallace in [Kim and Wallace, 1997]. This method is a goal-oriented design evaluation model that employs the same goals and targets that are commonly used in engineering design to evaluate the performance of each solution.

6. Goal Programming: Goal programming dates back to the early sixties [Charnas and Cooper, 1961; Charnes et al., 1955]. However, more recent references could be found in [Charnas and Cooper, 1961; Charnes et al., 1955; Steuer, 1986; Tamiz et al., 1998]. In goal programming (GP) the objectives are formulated as goal criteria that the decision maker wants each objective to possess. The criteria could be formulated in the following ways. We want the objective to be: (i) Greater than or equal to, (ii) Less than or equal to, (iii) Equal to, (iv) In the range of.

7. Lexicographic approaches: In lexicographic approaches, the decision-maker determines an order in which the objectives have to be optimized. Lexicographic methods are not so commonly used by themselves in engineering design, but jointly with other techniques, such as in goal programming or as a part of a selection mechanism in genetic algorithms.

2.2.3 Progressive articulation of preference information

This class of method is generally referred to as interactive method. They rely on progressive information about the decision-makers (DM) preferences simultaneously as they search through the solution space. Interactive methods are very common within the field of operation research.

1. STEM Method: The STEM-Method or STEP-method was first presented by Benayoun *et al.* in [Benayoun et al., 1971]. In STEM and related methods, preference information from the DM is used to reduce the solution space successively.

2. Steuer method: There are a set of methods that sample a progressively smaller subset of the nondominated set by employing progressively changing weights in weighted sum approaches. Steuer and Choo exemplifies these types of methods in [Steuer and Choo, 1983].

2.2.4 Posteriori articulation of preference information

There are a number of techniques which enables to first search the solution space for a set of Pareto optimal solutions and present them to the decision-maker. The big advantage with this type of method is that the solution is independent of the DMs preferences. The analysis has only to be performed once, as the Pareto set would not change as long as the problem description is unchanged. However, some of these methods suffer from a large computational burden. Another disadvantage might be that the DM has too many solutions to choose from. However, there are methods that support in screening the Pareto set in order to cluster optimal solutions, see [Morse, 1980] and [Rosenman and Gero, 1985].

1. Multiple Run Approaches: This section discusses the most common approaches to obtain a sample set of points of the Pareto-optimal front. By sampling a set of discrete points on the Pareto front, the decision maker could get a feeling for the form of the front and thereby the possible trade-off between objectives. Excluded are methods that require the objective function to be differentiable.

   (a) Weighted sum approaches

   (b) e-constraint approach: In the e-constraint method, one objective is selected for optimization and the others are reformulated as constraints.

   (c) Normal boundary interaction: Normal-boundary interaction (NBI) is presented in [Das and Dennis, 1998]. The algorithm enables an easy way of establishing an evenly spread set of points on the Pareto front, by starting the search from points evenly spread on the CHIM (Convex hull of the individual minima). However, if the Pareto front has a very complex shape, the method might identify non-Pareto optimal solutions as well as local Pareto optimal solutions.

   (d) Multiobjective simulated annealing: There has been some work done in multiobjective simulated annealing [Suppapitnarm et al., 1999] where the non-dominated solutions found so far in search are stored in a separate array. For each new point that are examined, the set of non-dominated points is updated to include the new point if it is non-dominated and to exclude the points that the new one dominates. However, this is quite a new field where lot of research still has to be done. The same principle might be applicable to other methods as well, as for instance Tabu-search.

2. Multiobjective genetic algorithms: Lately, there has been a large development of different types of multiobjective genetic algorithms, which is also reflected in the literature. The big advantage of genetic algorithms over other methods is that a GA manipulates a population of individuals. It is, therefore, tempting to develop a strategy in which the population captures the whole Pareto front in one single optimization run. For an overview on genetic algorithms in multiobjective optimization, see Fonseca and Fleming [Fonseca and Fleming, 1995b]. Literature surveys and comparative studies on multiobjective genetic algorithms are also given in [Coello Coello et al., 1996; Horn, 1997; Tamaki et al., 1996a; Zitzler and Thiele, 1999].

   Fonseca and Fleming have divided multiobjective genetic algorithms in non-Pareto and Pareto based approaches.

   (a) Non-Pareto based approaches: The first multi-objective genetic algorithm was VEGA (Vector Evaluating Genetic Algorithm) developed by Schaffer [Schaffer, 1985]. VEGA uses the selection mechanism of the GA to produce non-dominated individuals. Each individual objective is designated as the selection metric for a portion of the population. However, it is reported that the method tends to crowd results at extremes of the solution space, often yielding poor coverage of the Pareto frontier.

   Fourman presents a genetic algorithm using binary tournaments, randomly choosing one objective to decide each tournament in [Fourman, 1985]. Kurasawe further developed this scheme in [Kursawe, 1991] by allowing the objective selection to be random, fixed by the user, or to evolve with the optimization process. He also added crowding techniques, dominance, and diploidy to maintain diversity in the population.

   All of these Non-Pareto techniques tend to converge to a subset of the Pareto-optimal frontier, leaving a large part of the Pareto set unexplored. Preferably, one wants to maintain diversity so that the entire Pareto front is elicited. Additionally, maintaining diversity will tend to improve robustness in multi-objective problems by ensuring that there is a genetic variety for mating mechanisms to operate upon [Grüninger and Wallace, 1996; Harik, 1995].

   (b) *Pareto based approaches:* Goldberg introduced non-dominated sorting to rank a search population according to Pareto optimality in [Goldberg, 1989]. First, non-dominated individuals in the population are identified. They are given the rank 1 and are removed from the population. Then the non-dominated individuals in the reduced population are identified, given the rank 2, and then they are also removed from the population. This procedure of identifying non-dominated sets of individuals is repeated until the whole population has been ranked. Goldberg also discusses niching methods and speciation to promote diversity so that the entire Pareto front is covered.

The non-dominated sorting GA (NSGA) [Srinivas and Deb, 1994] of Srinivas and Deb implements Goldberg's thoughts about the application of niching methods. In NSGA, non-dominated individuals in the population are identified, given a high initial individual score and are then removed from the population. These individuals are considered to be of the same rank. The score is then reduced using sharing techniques between individuals with the same ranking. Thereafter, the non-dominated individuals in the remaining population are identified and scored lower than the lowest one of the previously ranked individuals. Sharing is then applied to this second set of non-dominated individuals and the procedure continues until the whole population is ranked.

Sharing is performed in the parameter space rather than in the attribute space. This means that the score of an individual is reduced according to individuals with similar parameters, regardless of how different or similar they might be based on objective attributes.

In the multi-objective GA(MOGA) presented by Foseca and Fleming each individual is ranked according to their degree of dominance [Fonseca and Fleming, 1995a, 1998a]. The more population members that dominate an individual, the higher ranking the individual is given. An individual's ranking equals the number of individual's that it is dominated by plus one. Individual's on the Pareto front have a rank of 1 as they are non-dominated. The rankings are then scaled to score individual's in the population. In MOGA both sharing and mating restrictions are employed in order to maintain population diversity. Fonseca and Fleming also include preference information and goal levels to reduce the Pareto solution set to those that simultaneously meet certain attribute values.

The niched Pareto GA (NPGA) [rey Horn et al., 1993] by Horn *et al.* is Pareto-based but does not use ranking methods. Rather, Pareto domination tournaments are used to select individuals for the next generation. For binary tournaments, a subset of the population is used as a basis to assess the dominance of the two contestants. If one of the contestants is dominated by a member in the subset but the other is not, the non-dominated one is selected to survive. If both or neither are dominated, selection is based on the niche count of similar individuals in the attribute space. An individual with a low niche count is preferred to an individual with a high count to help maintain population diversity.

Zitzler and Thiele developed a multi-objective genetic algorithm in [Zitzler and Thiele, 1999] called the strengthen Pareto evolutionary algorithm (SPEA). SPEA uses two populations, P and $P^{'}$. Throughout the process copies of all non-dominated individuals are stored in $P^{'}$. Each individual is given a fitness value, f_i, based on Pareto dominance. The fitness of the members of $P^{'}$ is calculated as a function of how many individuals in P they dominate

The individuals in P are assigned their fitness according to the sum of the fitness values for each individual in $P^{'}$ that dominate them plus one. Lower scores are better and ensure larger number of offsprings in the next generation. Selection is performed using binary tournaments from both populations until the mating pool is filled. In this algorithm, fitness assignment has a built-in sharing mechanism. The fitness formulation ensures that non-dominated individuals always get the best fitness values and that fitness reflects the crowdedness of the surroundings.

Other methods such as the one in [Tamaki et al., 1996b] presented by Tamaki builds on the methods described above and adds features such as elitism together with the usage of multiple populations [Fonseca and Fleming, 1998b].

2.3 Swarm Intelligence

The concept of SI was first used by [Hackwood and Beni, 1992; Hackwood and Wang, 1988]. In this context, simple agents occupied one or two dimensional grid environments and self organized through closest neighbor interactions.

In [Bonabeau et al., 1999] extended the SI definition as: "Using the expression *swarm intelligence* to describe only this work seems unnecessarily restrictive: that is why we extend its definition to include any attempt to design algorithms or distributed problem-solving devices inspired by the collective behavior of insect colonies and other animal societies."

SI could be defined as any attempt to design algorithms or distributed problem-solving devices whose behavior emerges from the social interaction between local neighbors. The word *swarm*, broadly speaking, describes a collection of interactive individuals. The classical example of a swarm is bees swarming around their hive; nevertheless the metaphor can easily be extended to any other system with a similar architecture. As ant colonies can be thought of as a swarm whose individuals are ants, so can a flock of birds. The concept of swarm can be extended to an even more general one: that of any type of collective behavior. Thus, a swarm might occur in high-dimensional cognitive spaces, where collision is no longer a concern and could simply mean agreement [Bonabeau et al., 1999].

SI is to simulate the social interactions between individuals, for evolution of global optima, similar to EAs. In SI, metaphors from successful behavior of animals or human societies are applied in problem solving. The goal is not to faithfully mimic the phenomena themselves but to use some of their aspects in practical applications.

2.3.1 Biogeography Based Optimization

Until the 1960s, the science of biogeography was mainly descriptive and historical [Darwin, 1995; Wallace, 2005]. 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 [MacArthur and Wilson, 1967]. Their interest was primarily focused on the distribution of species among neighboring *islands* (An island is any habitat that is geographically isolated from other habitats). 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.

2.3.2 Particle Swarm Optimization

PSO also belongs to the category of SI [Kennedy 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 [Eberhart and Kennedy, 1995]. PSO is an EC 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 [Eberhart et al., 1996]. 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 [Eberhart and Kennedy, 1995].

What differentiates the PSO paradigm from other instances of EC is memory and social interaction amongst the individuals. In the other paradigms, the important information an individual possesses, usually called *genotype*, is its current position, however, in PSO, really important asset is the *previous best experience*. Each individual stores the best position, found so far, that drives the evolution toward better solutions.

2.4 Yagi-Uda Antenna Design Optimization

A Yagi-Uda antenna is a widely used antenna design due to its high forward gain, low cost and ease of construction. It is a linear array of parallel elements, one of which is excited by voltage or current source and the others acts as directors in which currents are induced due to mutual coupling.

Yagi-Uda antenna was invented by H. Yagi and S. Uda in 1926 at Tohoku University in Japan [Uda and Mushiake, 1954], but first published in English in 1928 and it has been extensively used as an end-fire antenna [Yagi, 1928]. The goal of the design process is to develop an antenna that meets some desired performance characteristics. Yagi-Uda antenna is difficult to design and optimize due to their numerous parasitic elements. There are no simple formulas for designing Yagi-Uda antennas due to the complex relationship between physical parameters such as element length, spacing, and diameter. So many researchers have proposed different algorithm for the optimized design of Yagi-uda antenna those can be classified, broadly as:

1. Traditional mathematical based
2. Artificial Intelligence (AI) based

AI techniques are basically inspired from natural/biological systems, e.g., (1) social behavior of ants, birds and termites, etc, (2) Genetic improvements in species over generations, (3) Parallel processing of biological neurons (4) Decision making capabilities of human beings from linguistic information, etc. These all techniques are, further, classified as (1) Fuzzy Logic (2) Artificial Neural Networks (3) Evolutionary Algorithms (EAs).

2.5 Multiobjective Optimization for Yagi-Uda Antenna

Researchers have also presented multiobjective optimization of Yagi-Uda antenna using AI techniques. Romos *et al.* have used real-biased multiobjective GA to design wire antennas in [Ramos et al., 2003]. This procedure leads to better estimates of the Pareto set and is applied to the optimization of a Yagi-Uda antenna in a wide frequency range with several simultaneous performance specifications, providing antenna geometries with good performance.

In [Venkatarayalu and Ray, 2003, 2004], N. V. Venkatarayalu and T. Ray has used computational intelligence for single and multi-objective design of Yagi-Uda antenna. They introduce a population-based, stochastic, zero-order optimization algorithm and use it to solve single and multi-objective Yagi-Uda antenna design optimization problem. The algorithm is attractive as it is computationally efficient and does not require additional user inputs to model constraints or objectives.

Wang *et al.* has used hierarchical GA for optimization of Yagi array by optimizing the element spacing and lengths of Yagi-Uda antenna in [Wang et al., 2003]. This scheme has the ability to handle multiobjective functions as well as the discrete constraints in the numerical optimization process, where, more than one possible solution can be obtained. Furthermore, this feature also enables a design tradeoff between cost and performance without extra computational effort.

After the evolution in GA, Baskar *et al.* have used Comprehensive Learning Particle Swarm Optimization (CLPSO) in [Wang et al., 2003], to design Yagi-Uda antenna that is used to optimize the element spacing and lengths of Yagi-Uda antenna. SuperNEC, an object-oriented version of the numerical electromagnetic code (NEC-2) is used to evaluate the performance of various Yagi-Uda antenna designs. The three objectives considered are gain only, gain and input impedance only, and gain, input impedance and relative sidelobe level (rSLL). Each design problem is optimized using three variants of PSO algorithms, namely the modified PSO, fitness-distance ratio PSO (FDR-PSO), and comprehensive learning PSO (CLPSO). For the purpose of comparison, genetic algorithm and computational intelligence are taken into an account, and the results clearly show, that the CLPSO is a robust and useful optimization tool for designing Yagi antenna for the desired target specifications. In particular, this method can solve the multiobjective optimization problem using various Pareto-optimal solutions in an extremely efficient manner.

Y. Kuwahara has proposed multiobjective optimization design of Yagi-Uda antenna by using Pareto GA, in [Wang et al., 2003], by which various Pareto-optimal solutions for each objective function can be obtained that enables the selection of parameters in accordance with the design requirement. The effectiveness of the Pareto GA is compared with conventional GA and with the values of the design benchmark reference.

Varlamos *et al.* have proposed multiobjective genetic optimization to design Yagi-Uda arrays with additional parasitic elements [Varlamos et al., 2005]. The genetic algorithms are employed, and various objective functions such as gain, front-to-back ratio and input impedance are examined. Comparisons are made among the modified and conventional Yagi-Uda configurations and the modified Yagi-Uda array gave higher performance standards over an extended bandwidth around 2.4 GHz.

Lei *et al.* have used multiobjective optimization design of X-shape driven dipole Yagi-Uda antenna [Lei et al., 2007]. The effects of the angle that the x-shape driven dipole spread on the performance of the antenna are also studied. The simulated maximum antenna gain across the operating frequency band, obtained from the optimization process, is about 12.1 dBi. A wide-band Yagi-Uda antenna with an x-shape driven dipole for the Meteor Burst Communication (MBC) at the VHF is presented in said paper.

J. Y. Li and J. L. Guo have used DE algorithm for multiobjective optimization of Yagi-Uda antenna in [Li and Guo, 2009], in which method of moments MoM is used to evaluate antenna design and DE is employed to optimize the geometric parameters of Yagi-Uda antenna. The results clearly show that DE is a robust and useful optimization tool for the optimization of several conflicting objectives such as gain maximization, SLL reduction and input impedance matching. Multi-objective Evolutionary Algorithms (MOEAs) are suitable optimization techniques for solving such problems.

Goudos *et al.* have proposed Generalized Differential Evolution (GDE3), which is a multi-objective extension of DE to design Pareto optimal Yagi-Uda antenna in [Goudos et al., 2010]. Both GDE3 and Nondominated Sorting Genetic Algorithm-II (NSGA-II) are applied to Yagi-Uda antenna design under specified constraints. Three different Yagi-Uda antenna designs are considered and optimized and Pareto fronts are produced for both algorithms. The results indicate the advantages of this approach and the applicability of this design method.

2.6 Problem Formulation

All of the issues of nondominated sorting genetic algorithm (NSGA) proposed in [Srinivas and Deb, 1994] were addressed and an improved version of NSGA, which we call NSGA-II was proposed in [Deb et al., 2002] which performs much better as compared to the previous ones.

In [Singh et al., 2010], Singh *et al.* have optimized Yagi-Uda Antenna for three different design objectives which include gain, impedance and Side Lobe Level (SLL) by using constrained optimization approach and converting a multiobjective problem into a single objective. The results of BBO technique are better as compared to GA, [Jones and Joines, 1997] and CLPSO, [Baskar et al., 2005] both in terms of gain and impedance.

In [Singh and Sachdeva, 2012b], Singh and Sachdeva explored different migration variants of BBO for maximum gain optimization of Yagi-Uda antenna. In [Singh and Sachdeva, 2012a] mutation effects on BBO evolution in optimizing Yagi-Uda antenna design was carried out. The two, were optimized for single objective function, i.e., gain maximization.

In this thesis, the NS approach was used with BBO to propose NSBBO and to optimize two conflicting parameters Gain and Impedance simultaneously as multi-objective problem for Yagi-Uda Antenna design.

CHAPTER 3

MULTIOBJECTIVE OPTIMIZATION

In this chapter, the multiobjective problems and the derived optimization techniques are discussed. In two objective problem, both the objectives can be maximized, minimized or there can be a conflict for maximizing one objective and minimizing another objective. This chapter is dedicated to obtain the solutions for such conflicting objective problems.

3.1 Introduction

For a nontrivial multiobjective optimization problem, there does not exist a single solution that simultaneously optimizes each objective. In that case, the objective functions are said to be conflicting, and there exist (possible infinite number of) Pareto optimal solutions. A solution is called *nondominated*, *Pareto optimal*, *Pareto efficient* or *noninferior*, if none of the objective functions can be improved in value without impairment in some of the other objective values. Without additional preference information, all Pareto optimal solutions can be considered mathematically equally good (as vectors cannot be ordered completely). Researchers study multiobjective optimization problems from different viewpoints and, thus, there exist different solution philosophies and goals when setting and solving them. The goal may be finding a representative set of Pareto optimal solutions, and/or quantifying the trade-offs in satisfying the different objectives, and/or finding a single solution that satisfies the preferences of a human decision maker (DM).

3.2 Multiobjective Optimization Problems

Many real-world optimization problems involves multiple objectives. A *multiobjective optimization problem* (MOOP) can be mathematically formulated as given by (3.1):

$$\text{minimze F(x)} = (f_1(x), \ldots\ldots, f_m(x))^T; \text{s.t. x} \in \delta, \tag{3.1}$$

where δ is the decision space and $x \in \delta$ is a decision vector. *F(x)* consists of m objective functions $f_i : \delta \leftarrow R.\ i = 1, \ldots, m$, where R^m is the objective space.

The objectives in (3.1) often conflict with each other. Improvement of one objective may lead to deterioration of another. Thus a single solution, which can optimize all objectives simultaneously, does not exist. Instead, the best trade-off solutions called the *Pareto optimal solutions*, are important to a *decision maker* (DM). The Pareto optimality, concept, which was first proposed by Edgeworth and Pareto [Stadler, 1979], is formally defined as follows [Miettinen, 1999], [Deb, 2001].

Defination 1. A vector

$$u = (u_1, \ldots, u_m)^T$$

is said to dominate another vector

$$v = (v_1, \ldots, v_m)^T,$$

denoted as $u < v$, iff $\forall i \in (1, \ldots, m)$, $u_i \leq v_i$ and $u \neq v$.

Defination 2. A feasible solution $x^* \in \delta$ of problem (3.1) is called a *Pareto optimal solution*, iff $\nexists y \in \delta$ such that $F(y) < F(x^*)$. The set of all the Pareto optimal solutions is called the Pareto set (PS), denoted as

$$PS = \{x \in \delta | \nexists y \in \delta, F(y) \prec F(x)\}$$

The image formation of the PS in the objective space is called the *Pareto front* (PF)

$$PF = \{F(x) | x \in PS\}.$$

But in practise, since there could be a number of Pareto optimal solutions and the suitability of one solution depends on a number of factors, including designer's choice, problem environment, finding the entire set of Pareto optimal solutions may be desired. In the following section, the number of classical approaches to the solution of multiobjective optimization problems and their difficulties are described.

3.3 Classical Methods

A common difficulty with MOP is the conflicting nature of objectives. The opted solution is feasible that could be globally the best for all objectives [Hans, 1988]. In other words, individual optimal solution for all objectives are usually different. Thus mathematically most favorable Pareto-optimum solution is opted which offers least objective conflict. Such solutions can be viewed as points in the search space which are optimally placed from individual optimum of each objective. But such solutions may not satisfy a decision-maker because he/she may want a solution that satisfies some associated priorities of the objectives. To find such points, all classical methods scalarize the objective vector into one objective. Many classical algorithms for non-linear vector optimization techniques define a substitute problem, reducing the vector optimization to a scalar optimization problem. Using such a substitute, a compromise solution is found subject to specified constraints.

In the following subsections, three commonly used methods - *method of objective weighting*, *method of distance functions*, and *method of min-max formulation* - are discussed.

3.3.1 Method of Objective Weighting

This is probably the simplest of all classical techniques. Multiple objective functions are combined into one overall objective function, Z, as given by (3.2):

$$Z = \sum_{i=1}^{N} w_i f_i(x), \tag{3.2}$$

where $x \in X$, is the feasible region. The weights w_i are fractional numbers ($0 \leq w_i \leq 1$), and all weights are summed up to one, i.e., $\sum_{i=1}^{N} w_i = 1$. In this method, the optimal solution is controlled by the weight vector w. It is clear from (3.2) that the preference of an objective can be changed by modifying the corresponding weight. Mathematically, a solution obtained with equal weights to all objectives may offer least objective conflict, but as a real world situation demands a satisfying solution, priority must be induced in the formulation. In most cases, each objective is first optimized and all objective function values are computed at each individual optimum solution. Thereafter, depending on the importance of objectives a suitable weight vector is chosen and the single-objective problem given in (3.2) is used to find the desired solution. The only advantage of using this technique is that the emphasis of one objective over the other can be controlled and the obtained solution is usually a Pareto-optimum solution.

3.3.2 Method of Distance Functions

In this method, the scalarization is achieved by using a demand-level vector $\bar{y}$ which has to be specified by the decision maker. The single objective function derived from multiple objectives is as given by (3.3):

$$Z = \left[\sum_{i=1}^{N} |f_i(x) - \bar{y}_i|^r \right]^{1/r}, \tag{3.3}$$

where $1 \leq r < \infty$, and $x \in X$, is the feasible region. Usually an Euclidean metric $r = 2$ is chosen, with $\bar{y}$ as individual optima of objectives. It is important to note that the solution obtained by solving (3.3) depends on the chosen demand-level vector. Arbitrary selection of a demand level may be highly undesirable. This is because a wrong demand level will lead to a non Pareto-optimal solution. As the solution is not guaranteed, the decision maker must have a thorough knowledge of individual optima of each objective prior to the selection of demand level. In a way this method works as a goal programming technique imposing a goal vector/demand level, $\bar{y}$, for the given objectives. This method is similar to the method of objective weighting. The only difference is that in this method, the goal for each objective function is required to be known whereas in the previous method the relative importance of each objective is required.

3.3.3 Min-Max Formulation

This method is different in principle than the above two methods. It attempts to minimize the relative derivations of the single objective functions from individual optimum, i.e., it tries to minimize the objective conflicts. For a minimization problem, the corresponding min-max problem is formulated as given by (3.4):

$$\text{minimize } F(x) = \text{maximize } [Z_j(x)] \tag{3.4}$$

where $x \in X$, is the feasible region and $Z_j(x)$ is calculated for non-negative target optimal value $\bar{f}_j > 0$ as follows:

$$Z_j(x) = \frac{f_j - \bar{f}_j}{\bar{f}_j} \tag{3.5}$$

This method can yield, the best possible compromised solution when objectives with equal priority are required to be optimized. However, priority of each objective can be varied by introducing dimensionless weights in the formulation. This can also be modified as a goal programming technique by introducing a demand level vector in the formulation.

3.4 Drawbacks of Classical Method

In all above methods, multiple objectives are combined to form one objective by using some knowledge of the problem being solved. The optimization of the single objective may guarantee a Pareto optimal solution but results in single point solution. In real world situations decision makers often need different alternatives in decision making. Moreover, if some of the objectives are noisy or have discontinuous variable space, these methods may not work effectively. Some of these methods are also expensive as they require knowledge of individual optimum prior to vector optimization. The most profound drawback of these algorithms is their sensitivity towards weights or demand levels. The decision maker must have a thorough knowledge of the priority of each objective before forming the single objective from a set of objectives. The solutions obtained largely depend on the underlying weight vector or demand level. Thus, for different situations, different weight vectors need to be used and the same problem needs to be solved a number of times.

3.5 Non-Dominated Sorting

Solutions to a multiobjective optimization problem are mathematically expressed in terms of nondominated or superior points. The parameter x is a p dimensional vector having p design or decision variables. In a minimization problem, a vector x_1 is partially less than another vector x_2, ($x_1 < x_2$), when no value of x_2 is less than x_1 and at least one value of x_2 is strictly greater than x_1. If x_1 is partially less than x_2, we say that the solution x_1 *dominates* x_2 or the solution x_2 is inferior to x_1 [Tamura and Miura, 1979]. Any member of such vectors which is not dominated by any other member is said to be *nondominated* or *non-inferior*. Similarly, if the objective is to maximize a function we define a dominated point if the corresponding component is not greater than that of a nondominated point. The optimal solutions to a multiobjective optimization problem are known as nondominated solutions. These are also known as *Pareto-optimal* solutions. The idea behind the nondominated sorting procedure is that a ranking selection method is used to emphasize good points. This process continues till all solutions are classified into different non-dominated fronts, as shown in Figure 3.1. The pseudo code of nondominated sorting approach is depicted in Algorithm 1.

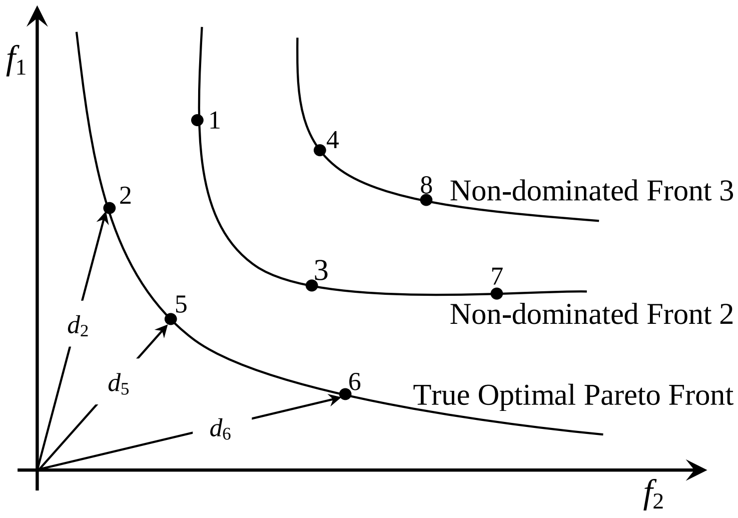


FIGURE 3.1: Non-dominated sorting and pareto-fronts

Algorithm 1 Pseudo Code for Nondominated Sorting

for $s = 1$ to NP

$\quad f_{1s} = |\mathrm{Re} - desired\ imp| + |\mathrm{Im}|$ and

$\quad f_{2s} = \dfrac{1}{gain}$

end for

$f = 1$ % Non-dominated front f

All solutions in the swarm set $\in F$

if (No. solutions in set $F \neq 0$)

$\quad f = f+1$

$\quad$ **for** i **=** 1 to NP

$\quad\quad$ **for** j **=** 1 to NP

$\quad\quad\quad$ **if** ($i \neq j$)

$\quad\quad\quad\quad$ **if** ($f_{1i} \le f_{1j}$ **and** $f_{2i} \le f_{2j}$)

$\quad\quad\quad\quad\quad$ j-th solution $\in D_f$

$\quad\quad\quad\quad$ **else**

$\quad\quad\quad\quad\quad$ j-th solution $\in P_f$

$\quad\quad\quad\quad$ **end if**

$\quad\quad\quad$ **end if**

$\quad\quad$ **end for**

$\quad$ **end for**

$\quad\quad F = D_f$

end if

3.6 Conclusion

In this chapter, multiobjective optimization problem and the methods to solve these problems are discussed to enable reader to have some background knowledge about multiobjective optimization. In this thesis, simultaneous optimization for maximal gain and impedance minimization to the targeted impedance for Yagi-Uda antenna is our target.

CHAPTER 4

BIOGEOGRAPHY BASED OPTIMIZATION

In this chapter, the science of biogeography and derived optimization technique (BBO) is discussed. BBO has two major operators, viz., migration and mutation. Researchers are constantly investigating and proposing variants to BBO for improved performance. This chapter is dedicated to all the variants of BBO and the algorithmic flow of these variants.

4.1 Introduction

Biogeography-based optimization (BBO) is one of the recently developed Evolutionary Algorithm for global optimization. Biogeography is the study of the geographical distribution of biological organisms over time and space. Its aim is to elucidate the reason of the changing distribution of all species in different environments over time. BBO has features in common with other biology-based optimization methods, such as Ant Colony Optimization, Genetic Algorithm, Particle Swarm Optimization, Stimulated Annealing that makes it applicable to many of the same types of problems that they are used for, namely, high dimension problems with multiple local optima. BBO also has some features that are unique among biology based optimization methods. Hence, BBO is one of the newest evolutionary algorithms, but it has already proven itself a worthy competitor to its better known siblings [Baskar et al., 2005; Jones and Joines, 1997; Rattan et al., 2008; Venkatarayalu and Ray, 2004].

4.2 Biogeography Based Optimization

Originally, the science of biogeography was traced to the work of nineteenth century naturalists such as Alfred Wallace and Charles Darwin as a descriptive study [Darwin, 1995; Wallace, 2005]. In the early 1960s, Robert MacArthur and Edward Wilson began working on mathematical models of biogeography [MacArthur and Wilson, 1967] and focussed their work on the distribution of species among neighboring islands and the mathematical models for the extinction and migration of species.

4.2.1 Terminology

1. **Habitat:** The habitat is any *Island* that is geographically isolated from other islands. Therefore, we use generic term habitat in place of island. In science of biogeography, a habitat is an ecological area that is inhabited or covered by particular plants or animal species. The candidate solutions for problem, in BBO, are encoded as string as given by (4.1) and termed as habitats.

$$H = [SIV_1, SIV_2, \cdots, SIV_M] \tag{4.1}$$

2. **Habitat Suitability Index (HSI):** HSI can be considered as the dependent variable of the habitat. It is a measure of the goodness or fitness of the solution which is represented as a habitat. HSI is a numerical index that represents the capacity of a given habitat to support a selected species. Some habitats are more suitable for habitation than others.

3. **Suitability Index Variable(SIVs):** SIVs can be considered as the independent variables of the habitats. The variables that characterize habitability are called Suitability Index Variable (SIVs). The factors correlating the features with HSI are rainfall, diversity of vegetation, diversity of topographic features, land area and temperature, etc.

4. **Migration:** Migration is the periodic seasonal movement of species from one geographic region to another, typically coinciding with available food supplies or breeding seasons. In BBO, migration of species from one island to another involves two important terms emigration and immigration. Where emigrate means to leave one's island to settle in another and immigrate means to settle in an island where one isn't a native. Emigrate stresses leaving while, immigrate stresses arriving.

4.2.2 Features

Following are some features of good habitats over poor habitats.

4.2.2.1 Features of High HSI Habitats

1. Habitats with high HSI tend to have large number of species, while those with low HSI have small number of species.
2. Habitats with high HSI have low immigration rate because they are already nearly saturated with species.
3. They have high emigration rate; large number of species emigrate to neighboring habitats.
4. Good solutions represents habitat with high HSI. Good solutions are more resistant to change than poor solutions.

4.2.2.2 Features of Low HSI Habitats

1. Habitats with low HSI have high immigration rate and low emigration rate because of their sparse population.
2. The emigration of new species to low HSI habitats may raise the HSI of the habitat.
3. An HSI is proportional to biological diversity.
4. Poor solutions represents habitat with low HSI. Poor solutions are more dynamic and accept a lot of new feature from good solutions.

4.2.3 Characterization

BBO characteristic curve gives us a general description of the process of immigration and emigration as shown in Figure 4.1 and illustrates a model of species abundance in a single habitat. The immigration rate, λ, and the emigration rate, μ, are the functions of the number of the species in that habitat. At the initial point, habitats with low HSI tend to have a 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 doesnot increase and remains low, species in that habitat go extinct that leads to additional immigration. For sake of simplicity, it is safe to assume a linear relationship between HSI (or population) and immigration and emigration rates and same maximum emigration and immigration rates, i.e., E = I as depicted graphically in Figure 4.1. On the other hand, the habitats with a high 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.

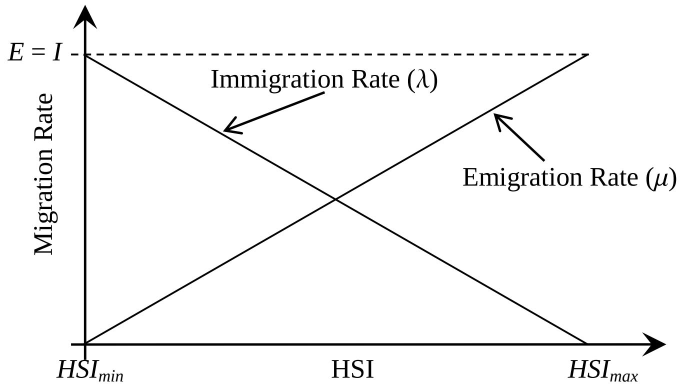


FIGURE 4.1: Migration Curves

For k-th habitat, i.e., HSI_k, the values of emigration rate and immigration rate are given by (4.2) and (4.3).

$$\mu_k = E \cdot \frac{HSI_k}{HSI_{max} - HSI_{min}} \tag{4.2}$$

$$\lambda_k = I \cdot \left(1 - \frac{HSI_k}{HSI_{max} - HSI_{min}}\right) \tag{4.3}$$

The candidate solutions are referred as habitats and associated HSI is analogous to fitness in other EAs. The immigration of new species from high HSI to low HSI habitats may raise the HSI of poor habitats as good solutions are more resistant to change than poor solutions whereas poor solutions are more dynamic and accept a lot of new features from good solutions.

Each habitat in a population of size NP, in BBO, is represented by M-dimensional vector as $H = [SIV_1, SIV_2, \cdots, SIV_M]$ where M is the number of SIVs (features) to be evolved for optimal HSI. HSI is the degree of acceptability that is determined by evaluating the cost/objective function, i.e., $HSI = f(H)$ H means habitat.

4.3 BBO Algorithms

Algorithmic flow of BBO involves two mechanisms, i.e., (i) migration and (ii) mutation, these are discussed in the following subsections:

4.3.1 Migration and its Variants

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 $i = 1, 2, \ldots, NP$) uses its immigration rate, λ_i given by (4.3), to probabilistically decide 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 (4.2). The process of migration is completed by copying values of SIVs from H_j to H_i at random chosen sites. The pseudo code of migration operator is depicted in Algorithm 2.

Algorithm 2 Pseudo Code for Migration

```
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
                Randomly select an SIV(s) from H_j
                Copy these SIV(s) in H_i
          end if
      end for
    end if
end for
```

Migration may lead to same types of habitats because of copying of SIVs or features from high HSI habitats to low HSI habitats. To reduce the number of same type of habitats and make BBO convergence faster, migration variants were introduced. Following subsections discusses the various migration variants:

1. Blended Migration

2. Immigration refusal

3. Enhanced Biogeography-Based Optimization

4.3.1.1 Blended BBO

Blended migration operator is a generalization of the standard BBO migration operator, inspired by blended crossover in GAs [McTavish and Restrepo, 2008]. In blended migration, a SIV value of immigrating habitat, $ImHbt$, is not simply replaced by a SIV value from emigrating habitat, $EmHbt$, as happened in standard BBO migration operator. Rather, a new solution feature is formed, i.e., SIV value is comprised of two components as $H_i(SIV) \leftarrow \alpha \cdot H_i(SIV) + (1-\alpha) \cdot H_j(SIV)$. Here α is a random number between 0 and 1. The pseudo code of blended migration is depicted in Algorithm 3.

Algorithm 3 Pseudo Code for Blended Migration

```
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
                H_i(SIV) ← α • H_i(SIV) + (1 − α) • H_j(SIV)
            end if
        end for
    end if
end for
```

4.3.1.2 Immigration Refusal BBO (IRBBO)

In BBO, if a habitat has high emigration rate, i.e, the probability of emigrating to other habitats is high and the probability of immigration from other habitats is low. However, the low probability does not mean that immigration will never happen. Once in a while, a highly fit solution may receive solution features from a low-fit solution that may degrade its fitness to a high-fit solution. In such cases, immigration is refused to prevent degradation of HSI values of habitats. This BBO variant with conditional migration is termed as Immigration Refusal [Du et al., 2009] whose performance with test bed of benchmark functions is found encouraging. The pseudo code of Immigration Refusal migration is depicted in Algorithm 4.

4.3.1.3 Enhanced BBO (EBBO)

Standard BBO migration operator tends to create duplicate solutions which decreases the diversity in the population. To prevent this diversity decrease in the population, duplicate habitats are replaced with randomly generated habitats. This leads to increase exploration of new SIV values. In EBBO, clear duplicate operator is integrated in basic BBO algorithm to improve its performance. The migration pseudo code of Enhanced BBO is depicted in Algorithm 5.

4.3.2 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.

Algorithm 4 Pseudo Code for Immigration Refusal BBO

```
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
```

Algorithm 5 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
```

The mutation rate, $mRate$, for k-th habitat is calculated as (4.4)

$$mRate_k = C \times \min(\mu_k, \lambda_k) \tag{4.4}$$

where μ_k and λ_k are emigration and immigration rates, respectively, given by (4.2) and (4.3) corresponding to HSI_k. To reduce fast generation of duplicate habitats, scaling constant, C, is chosen as 3 to keep exploitation rate much higher as compared to other EAs. The pseudo code of mutation operator is depicted in Algorithm 6.

Algorithm 6 Pseudo Code for Mutation

```
mRate = C x min(μk , λ k)
for n = 1 to NP do
    for j = 1 to number of SIV(s)
        Select Hj(SIV) with mRate
        if Hj(SIV) is selected then
            Replace Hj(SIV) with randomly generated SIV
        end if
    end for
end for
```

4.4 Conclusions

This chapter discusses the nature inspired biogeography based optimization with its various migration variants to propose Nondominated sorting BBO (NSBBO) for optimizing multiple parameters of Yagi-Uda antenna design.

CHAPTER 5

PARTICLE SWARM OPTIMIZATION

In this chapter, the sociologically inspired optimization technique is discussed. PSO has three major algorithmic variants, viz., gbest, lbest and hbest models. This chapter is dedicated to the strategic parameters, topologies and all the variants of PSO and the algorithmic flow of these variants.

5.1 Introduction

Eberhart and Kennedy developed PSO [Kennedy and Eberhart, 1995] based on the analogy of bird flock and fish school where each individual is allowed to learn from the experiences of its own and others.

PSO is a swarm based optimization tool which is useful, like other EAs, to evolve near optimum solution to a problem. The evolution is initialized with a set of randomly generated potential solutions and then is allowed to search for the optimum one, iteratively. It searches the optimum solution by observing the best performing particles. As compared to GAs, the PSO has much better intelligent background and could be performed more easily [Shi et al., 2007]. Due to its advantages, the PSO is not only suitable for scientific research, but also engineering applications. PSO has attracted broad attention in the fields of EC, optimization and many others [Angeline, 1998; Clerc and Kennedy, 2002; Trelea, 2003]. Although the PSO is developed for continuous optimization problems, however, investigational studies have been reported that are focussed on discrete problems as well [Kennedy and Eberhart, 1997].

5.2 Particle Swarm Optimization

5.2.1 Strategic Parameters

5.2.1.1 Population Size

Selection of population size is problem dependent, however, population sizes ranging from 20 to 50 are the most common as PSO needs smaller populations than other EAs to reach high quality solutions. Rate of evolution may become slower for much larger population size, as more computations will be required, whereas too small population of particles may vanish the primary requirement of randomness.

5.2.1.2 Maximum Velocity

To prevent explosion, in the original version, the particles velocities were clamped to a maximum velocity, V_{max}. If the velocity exceeded V_{max} in any coordinate, it was truncated to that value.

V_{max} was, therefore, an important parameter. If it was too high, particles could skip over good solutions. If V_{max} was too small, particles explored too slowly, and good solutions could not be found. Particles could become trapped in local optima because they were unable to move out of the attraction basin.

Early experience with the acceleration constants φ_1 and φ_2 concluded that it was possible to set them to 2.0 for almost all applications. V_{max} was the only parameter that needed to be adjusted. It was, however, noted that the optimal setting of this parameter was dependent on the problem [Mendes, 2004].

5.2.1.3 Bound Checking

Most of the research in PSO targets for the optimal value of the fitness function in a certain hypercube and, therefore, it is necessary to somehow enforce the exploration to remain inside that valid hyperspace. This is usually handled by resetting the particles within valid bounds whenever necessary.

However, in some situations, bound resetting does more harm than good. The best way to solve this predicament may be not to use bound checking at all. By modifying the fitness function as to assign the $+\infty$ value (assuming the optimization problems is a minimization one) to non-valid solutions may be a better approach. In this way, the particle will soon enter the valid space [Mendes, 2004].

5.2.1.4 Inertia Weight

In 1998, the concept of an inertia weight was introduced by [Shi and Eberhart, 1998a,b] to better control the scope of the search, motivated by the desire to reduce or even eliminate the importance of V_{max}. The velocity is described as follows, where an inertia weight coefficient, α, is included:

$$v_i = \alpha v_i + U[0, \varphi_1](p_i - x_i) + U[0, \varphi_2](p_g - x_i) \tag{5.1}$$

$$x_i = x_i + v_i \tag{5.2}$$

The use of the inertia weight improved performance in a number of applications. Originally, it was linearly varied between 0.9 and 0.4 during a run, providing a balance between *exploration* (larger steps in the beginning) and *exploitation* (smaller advancements, resulting in fine tuning in the later stage). Its use resulted in fewer iterations, on average, to attain a suitably good solution. Recent studies have used a random inertia weight about 0.5.

5.2.1.5 Constriction Coefficient

[Clerc, 1999] modified the system by introducing a constriction coefficient. Based on this system, he showed that a generalized particle swarm system can be created in which both explosion and convergence can be controlled. The simplified system given by [Clerc and Kennedy, 2002] is as follows:

$$v_i = \chi(v_i + U[0, \varphi_1](p_i - x_i) + U[0, \varphi_2](p_g - x_i)) \tag{5.3}$$

$$x_i = x_i + v_i \tag{5.4}$$

The following formula is used to compute the constriction coefficient:

$$\chi = \frac{2k}{|2 - \varphi - \sqrt{\varphi^2 - 4\varphi}|} \tag{5.5}$$

where, $k = [0, 1], \varphi = \varphi_1 + \varphi_2, \varphi > 4$; Most researchers using the constriction method use φ set to 4.1 (thus having $\varphi_1 = \varphi_2 = 2.05$ and $k = 1$ which determines that $\chi \approx 0.729$. This is algebraically equivalent to using the inertial model with $\alpha \approx 0.729$ and $\varphi_1 = \varphi_2 \approx 1.49445$.

The advantage of using constriction is that there is no need to use V_{max} nor to guess the values for any parameters governing convergence and preventing explosion.

5.2.2 Topologies

After observing the detrimental effect of using *gbest* PSO, [Kennedy, 1999] conducted the first study on other sociometries besides *gbest*, to observe their influence on the performance. In these experiments, following topologies were investigated shown in Figure 5.1:

1. **gbest**: where every particle is connected to every other.
2. **lbest**: where every particle is connected to two others.
3. **wheels**: where every particle was connected to a central one.

The main conclusions of this study were that *gbest* seemed to be faster but more vulnerable to local optima whilst *lbest* was much slower but more robust if the maximum number of iterations were increased. The *lbest* topology, as extreme vertices are interconnected, is also termed as ring topology. Wheels performed badly except on one of the functions, however, these results were deemed somewhat inconclusive.

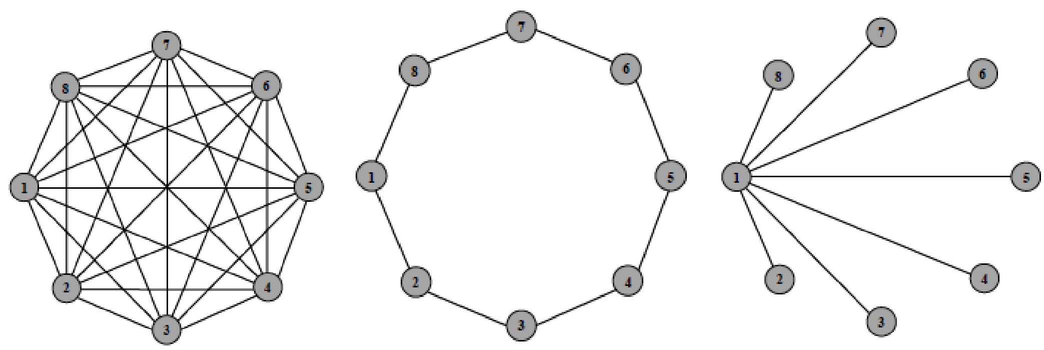


FIGURE 5.1: PSO Topologies *gbest*, *lbest*, and *wheels*

In [Mendes, 2004], investigated another topology, namely, *linear* there is only one possible path between two vertices, as shown in Figure 5.2. It is noticed that that particles with higher indices take longer to converge. The reason for this behavior is the fact that information takes time to travel along the graph from one individual to the next.

The next logical step was to generalize *lbest* by using a bigger neighborhood adjacency. Figure 5.3 shows how this is performed: instead of being connected to just one neighbor on each side, each individual is connected to several. Several researchers have used a neighborhood of size two on each side.

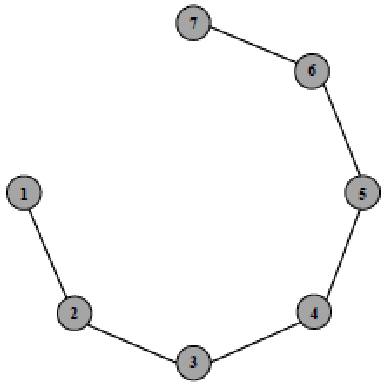


FIGURE 5.2: The linear topologies with only one path between two vertices

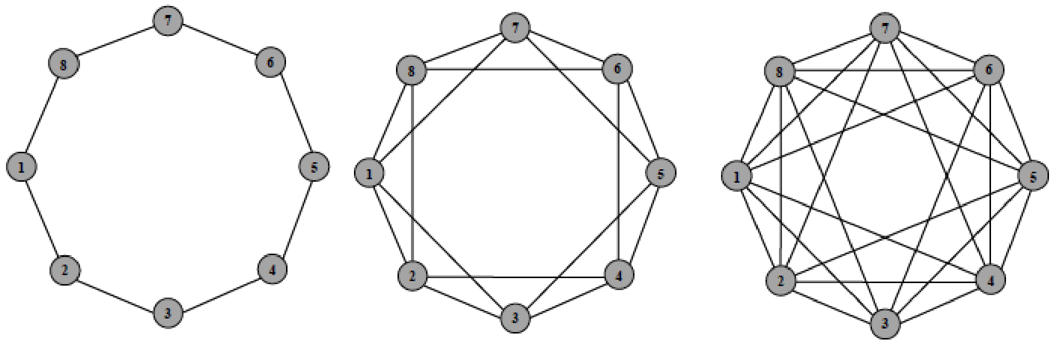


FIGURE 5.3: The ring topologies with neighborhood sizes 1, 2 and 3

The increase of robustness of *lbest* over *gbest* indicated that a more thorough study of the effect of sociometries in PSO performance was needed. An attempt to study the influence of neighborhoods on PSO performance is described in [Kennedy and Mendes, 2002]. They have reported considerable influence of different topologies on the behavior of the algorithm.

5.2.3 Characterization

There are several parameters that need to be defined in order to successfully use PSO to solve a given problem:

1. Solution Encoding: This is problem dependent, however, for the time being it is a mapping from the solution space to a valuation of a vector in d-space, where d is the problem dimensionality. This usually involves a minimum and a maximum value allowed in each dimension, thus defining a hyperspace. Some variations use a binary representation [Mendes, 2004].
2. Fitness Function: This function is also problem dependent and represents a measure of the quality of any given solution. The function should somehow create a total ordering

in the solution space. It is also of vital importance that small improvements in solution are visible to the algorithm.

3. Population Size: This parameter influences the behavior of the algorithm. A very small population does not create enough interaction for the emergent behavior pertaining to PSO to occur. However, the solution is not to simply increase the number of individuals. The relationship between population size (and thus the number of function evaluations per iteration) and performance follows a well known law of group interaction and is non-linear.

4. Acceleration Coefficients: The acceleration coefficients φ_1 and φ_2 are usually set to the same value. In fact, people usually talk about φ which sets the other two values $\varphi_1 = \varphi_2 = \frac{\varphi}{2}$. If φ is too small, the maximum step size used is quite small and so the algorithm will explore very slowly, degrading the performance. However, taking into account the adaptive step size of the algorithm, it is not such an important parameter. There is a consensus among the researchers to set $\varphi = 4.1$ [Mendes, 2004].

5. Constriction or Inertia Coefficient: Following the work of [Clerc and Kennedy, 2002], this value is determined by φ. It is thus not necessary to guess its value. If the value of φ is set to 4.1, then $\chi \approx 0.729$.

6. Maximum Velocity: With the advent of the constriction coefficient, most researchers do not bother using this parameter. However, some researchers report slightly better results using it nonetheless. Those who do, follow a simple rule of thumb: just set it to the dynamic range.

7. Neighborhood Topology: One of the aspects of the algorithm that is the least understood is how the interaction between the individuals actually works. When the algorithm was presented, the neighborhood best was the same for the entire population: the best performing solution found by the swarm so far. It has been shown, however, that such a strategy is prone to the pitfalls of local optima. The attempt to improve the algorithm resilience and the understanding of the interaction between the individuals triggered the study of population topologies.

5.3 PSO Algorithm Variants

PSO is a sociologically inspired optimization technique, since it was initially developed as a tool by Reynolds [Kennedy et al., 2001], [Reynolds, 1987] for simulating the flight patterns of bird flocks, which was mainly governed by three major concerns: *collision avoidance, velocity matching and flock centering.* On the other hand, the reasons presented for the flocking behavior observed in nature are: *protection* from predator and *gaining-food* from a large effective search-space. The latter reason assumes a great importance, when the food is unevenly distributed over a the search-space. It was realized by Kennedy and Eberhart that the bird flocking behavior can be adopted to be used as an optimizer and resulted in the first simple version of PSO [Kennedy and Eberhart, 1995], [Eberhart and Kennedy, 1995] that has been recognized as one of the computational intelligence techniques intimately related to EAs. Like EAs, it uses a population of potential solutions called particles that are flown through the search-space with adaptable velocities that determines their movements. Each particle also has a memory and hence it is capable of remembering the best position, in the search-space, ever visited by it. The position corresponding to the best fitness is known as *pbest* and the overall best out of all the particles in the population is called *gbest*.

Consider that the search space is d-dimensional and i-th particle in the swarm can be represented by $X_i = (x_{i1}, x_{i2}, \ldots, x_{id})$ and its velocity can be represented by another d-dimensional vector $V_i = (v_{i1}, v_{i2}, \ldots, v_{id})$. Let the best position ever visited in the past by i-th particle be denoted by $P_i = p_{i1}, p_{i2}, ..., p_{id})$. Many a times, the whole swarm is subdivided into smaller groups and each group/sub-swarm has its own local best particle, denoted as $P_l = (p_{l1}, p_{l2}, \ldots, p_{ld})$, and an over all best particle, denoted as $P_g = (p_{g1}, p_{g2}, \ldots, p_{gd})$, where g and l are particle indices.

5.3.1 Global-Best (*gbest*) PSO Model

In this PSO model, each particle is free to interact with its present *pbest* and *gbest* particles as described by (5.6) and (5.7)

$$v_{id}^{m+1} = \chi(\alpha v_{id}^{m} + \varphi_1 r_1 (p_{id}^{m} - x_{id}^{m}) + \varphi_2 r_2 (p_{gd}^{m} - x_{id}^{m})) \tag{5.6}$$

$$x_{id}^{m+1} = x_{id}^{m} + v_{id}^{m+1} \tag{5.7}$$

The PSO parameters viz., *inertia weight* α, *cognitive acceleration* φ_1, *social acceleration* φ_2, along with $v_{id} \in [-V_{max} + V_{max}]$ are known as the *strategy/operating* parameters of PSO algorithm that are specified by the user before the evolution starts. The parameter V_{max} is the maximum velocity along any dimension, which implies that, if the velocity along any dimension exceeds V_{max} , it shall be clamped to this value to avoid search explosion. The inertia weight, w, governs how much of the velocity should be retained from the previous time step. Generally the inertia weight is not kept fixed and is varied as the algorithm progresses so as to allow the PSO to explore a large area at the start of simulation run and to refine the search later by a smaller inertia weight. The parameters φ_1 and φ_2 determine the relative pull of *pbest* and *gbest*. Random numbers r_1 and r_2 help in, stochastically, varying these pulls, that also account for slight unpredictable natural of swarm behavior. This is depicted, graphically, in two-dimensional search-space in Figure 5.4.

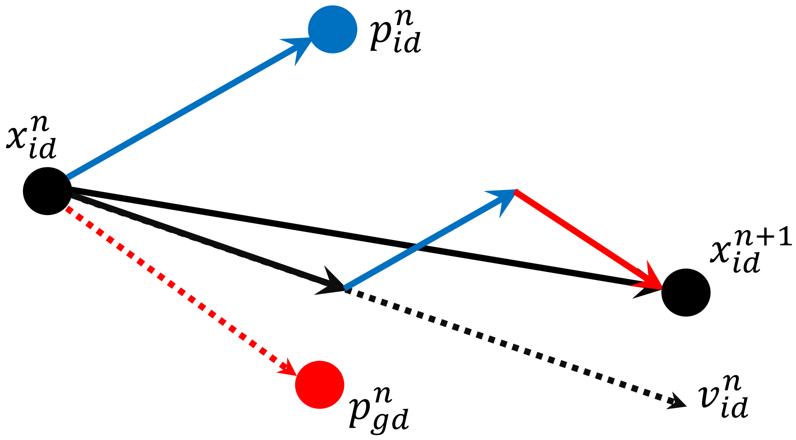


FIGURE 5.4: Particle movement in *gbest* model

5.3.2 Local-Best (*lbest*) PSO Model

In this model, the whole swarm is subdivided into sub-swarms. The model is termed as *lbest* PSO model if each particle in the swarm experiences attractions due its present *pbest* particle, P_i , and *lbest* particle, P_l. Mathematically, given by (5.8) and (5.9)

$$v_{id}^{m+1} = \chi(\alpha v_{id}^{m} + \varphi_1 r_1 (p_{id}^{m} - x_{id}^{m}) + \varphi_3 r_3 (p_{ld}^{m} - x_{id}^{m})) \tag{5.8}$$

$$x_{id}^{m+1} = x_{id}^{m} + v_{id}^{m+1} \tag{5.9}$$

The neighborhood acceleration parameter, φ_3, determines the relative pull of *lbest* particle, whereas random numbers, r_3, varies these pulls, stochastically, as shown in Figure 5.5.

5.3.3 Hybrid-Best (*hbest*) PSO Model

This is a newly proposed PSO variant where each particle belongs to a sub-swarm and feels collective attraction towards its present *pbest* particle, P_i , the *lbest* particle, P_l , and the *gbest* particle, P_g, as expressed mathematically,

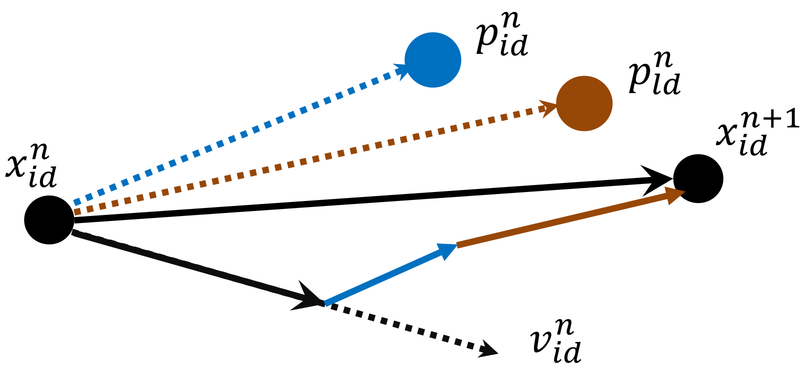


FIGURE 5.5: Particle movement in *lbest* PSO model

$$v_{id}^{m+1} = \chi(\alpha v_{id}^m + \varphi_1 r_1(p_{id}^m - x_{id}^m) + \varphi_2 r_2(p_{gd}^m - x_{id}^m) + \varphi_3 r_3(p_{ld}^m - x_{id}^m)) \tag{5.10}$$

$$x_{id}^{m+1} = x_{id}^m + v_{id}^{m+1} \tag{5.11}$$

These collective pulls that govern the movement of each particle in the swarm are depicted, in two dimensional search space, in Figure 5.6.

The *gbest* PSO model is endowed with the quality of faster evolution due to involvement of less number of computations but prone to be trapped into local optima. The second model, *lbest* PSO is slower as it involves more calculation, however, capable of escaping local optima and, therefore, avoids premature convergence. In this model, these both capabilities, i.e., fast and matured convergence, are embedded together.

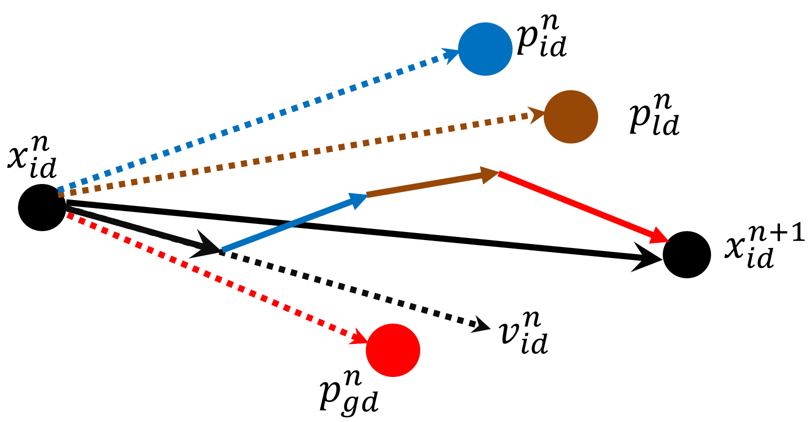


FIGURE 5.6: Particle movement in *hbest* PSO model

5.4 Conclusions

This chapter discusses the PSO algorithm based on the analogy of bird flock and fish school with its various algorithmic variants. In the forthcoming chapters, *gbest* algorithmic variant of PSO have been explored for comparative performance analysis with NSBBO to optimize multiple parameters of Yagi-Uda Antenna.

CHAPTER 6

YAGI-UDA ANTENNA

An antenna is an integral component of a radio communications system that is used to send or receive a radio signal. A radio frequency (RF) signal that has been generated in a radio transmitter travels through a transmission line (coaxial cable) to an antenna. An antenna connected to a transmitter is the device that releases RF energy (in the form of an electromagnetic field) to be sent to a distant receiver. The receiving antenna picks up the RF energy. As the electromagnetic field strikes the receiving antenna, a voltage is induced into the antenna, which serves as a conductor. The induced RF voltages are then used to recover the transmitted RF information.

6.1 Introduction

The Yagi antenna was invented in Japan, with results first published in 1926. The work was originally done by Shintaro Uda, but published in Japanese. The work was presented for the first time in English by Yagi (who was either Uda's professor or colleague), who went to America and gave the first English talks on the antenna, which led to its widespread use. Hence, even though the antenna is often called a Yagi antenna, Uda probably invented it.

The Yagi-Uda antenna or Yagi Antenna is one of the most brilliant antenna designs. It is simple to construct and has a high gain, typically greater than 10 dB. The Yagi-Uda antennas typically operate in the HF to UHF bands (about 3 MHz to 3 GHz), although their bandwidth is typically small, on the order of a few percent of the center frequency.

6.2 Geometry of Yagi-Uda Antenna

Yagi-Uda Antenna designing consists of three types of parasitic elements, viz, *driven element*, *reflector element* and *director elements*. Their features and characteristics are described in the following subsections and depicted by the Figure 6.1. An N element Yagi-Uda Antenna consists of $2N-1$ variables and is given as:

$$y = [L_0, L_1, \cdots, L_{N-1}, S_0, S_1, \cdots, S_{N-2}] \tag{6.1}$$

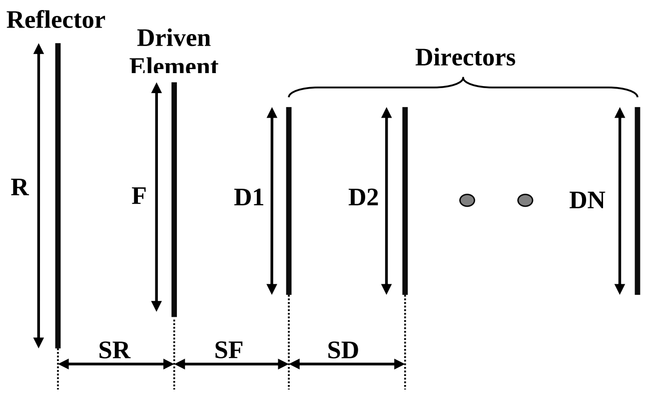


FIGURE 6.1: Geometry of Yagi-Uda Antenna

6.2.1 Driven Element

The driven element of a Yagi-Uda antenna is the feed point where the transmission line is attached to the antenna. 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. It might contain traps so that it resonates on more than one band. The driven element can be either as an electrically separate dipole or together with the boom.

6.2.2 Reflector Element

The reflector is the element that is placed at the rear of the driven element. It's resonant frequency is lower and its length is approximately 5% longer than the driven element. The length of reflector depends on the spacing, the element diameter as well as gain, bandwidth, front to Back ratio (F/B ratio), and SLL pattern requirements of the antenna design.

6.2.3 Director Elements

The directors elements can be one or more in number with different lengths. These are smaller than feeder and reflector elements. There resonance is slightly higher in frequency than the driven element and its length will be about 5% shorter, progressively than the driven element. The length of directors depends upon the director spacing, the number of directors used in the antenna, the desired pattern, pattern bandwidth and element diameter. The number of directors that can be used are determined by the physical size or length.

6.3 Radiation Pattern of Yagi-Uda Antenna

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 the reception pattern as well, since it also describes the receiving properties of the antenna. The three dimensional radiation pattern is depicted, in Figure 6.2, however, usually the measured radiation patterns are a two dimensional slice of the three-dimensional pattern, in the horizontal and/or vertical planes as shown in Figure 6.3

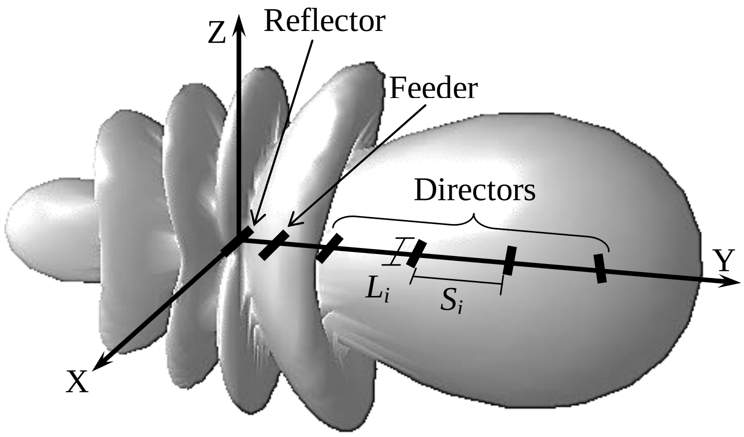
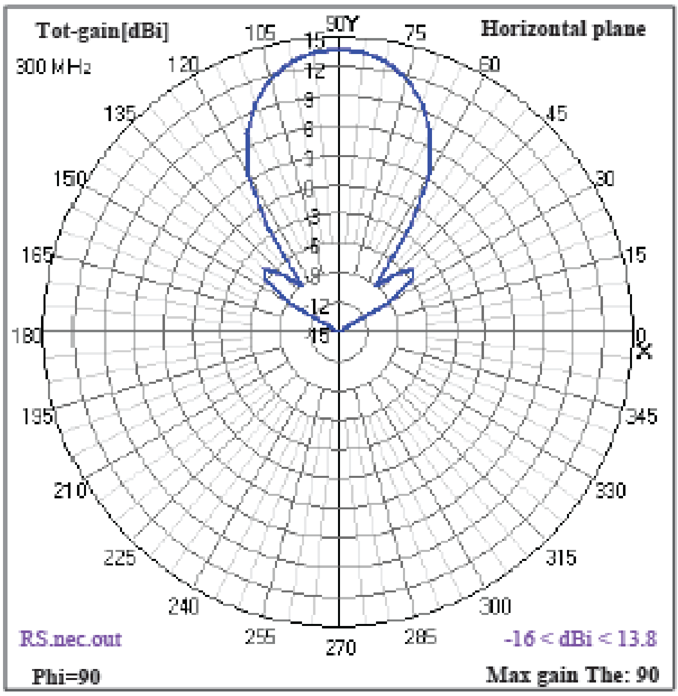


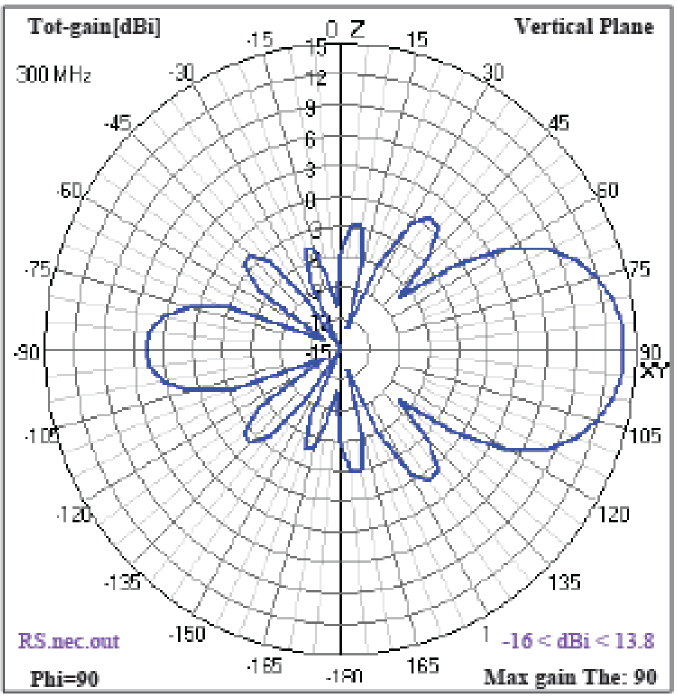
FIGURE 6.2: 3D Radiation Pattern of Six Elements Yagi-Uda Antenna

These pattern measurements are presented in either a rectangular or a polar format. A polar format of the gain verses orientation (radiation pattern) is useful when characterizing antennas. Some important features 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 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 decibles (dB) relative to an isotropic source or a standard dipole (in direction of maximum gain) represent the



(a) Horizontal Plane



(b) Vertical Plane

Figure 6.3: 2D Radiation Pattern of Six Elements Yagi-Uda Antenna

improvement in signal level to reference antenna. Typically, the higher the gain, more the efficient antenna performance, and longer the range of the antenna will operate. Radiation pattern of six-elements Yagi-Uda antenna, depicted in Figure 6.4, which is used to calculate gain of Yagi-Uda antenna.

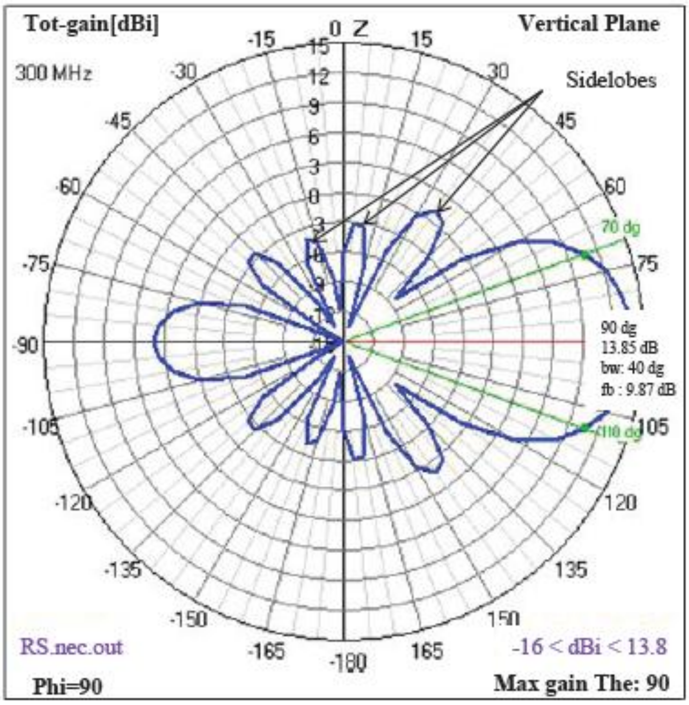


FIGURE 6.4: Radiation Pattern for Gain of Yagi-Uda Antenna

2. **Front to Back ratio:** The F/B ratio is used in describing directional radiation patterns for antennas. If an antenna has a unique maximum 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) also expressed in dB as depicted in Figure 6.4.

3. **Beamwidth:** Beamwidth is the angle between directions where the power is the half the value at the direction of maximum gain which is -3dB. It gives a measure a directivity of antenna as depicted in Figure 6.4.

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 Figure 6.4, 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. **Relative 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. Radiation resistance is used to match the impedance of antenna to impedance of transmission cable otherwise signal loss and high voltages in the cable may occur.

6.4 Conclusions

In this chapter, various design parameters of Yagi-Uda antenna are discussed to enable reader to have some background knowledge about Yagi-Uda antenna. In this thesis, simultaneous gain and impedance optimization is targeted in the coming using NSBBO algorithms and its variants and NSPSO and its algorithmic variants reported till date.

CHAPTER 7

IMPLEMENTATION

In this chapter, firstly, NEC software, developed for evaluation of antenna designs, is presented. Secondly, implementation of algorithmic flow of nondominted sorting approach for multiobjective optimization of two conflicting objectives is presented. Thirdly, algorithmic flow of NSBBO and NSPSO in C++ environment along with NEC2, are discussed, in detail.

7.1 Introduction

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. 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, and performance characteristics such as gain and input impedance. With an N element Yagi-Uda antenna, there are $2N-1$ parameters, i.e., N wire lengths and $N-1$ spacings. To evolve optimal antenna design NSBBO and its variants are proposed and investigated for faster convergence and better results. The results are explored for NSPSO too. QT Creator is used as to create NSBBO and NSPSO algorithms in C++, whereas NEC2 is used to evaluate all antenna designs for gain, impedance, etc.

7.2 Implementation Requirements

The design evolution of Yagi-Uda antenna using NSBBO and NSPSO requires QT Creator for C++ programming and NEC2 to evaluate antenna design based on method of moments. Their brief introduction is presented in following subsections:

7.2.1 QT Creator for C Code

Qt Creator is a cross-platform C++ integrated development environment which is part of the Qt SDK. It includes a visual debugger and an integrated GUI layout and forms designer. The editor's features includes syntax highlighting and auto-completion, but not tabs. Qt Creator uses the C++ compiler from the GNU Compiler Collection on Linux and FreeBSD. On Windows it can use MinGW or MSVC with the default install and can also use cdb when compiled from source.

7.2.2 Numerical Electromagnetic Code2 (NEC2)

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.

7.2.3 How to use NEC2

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.

2. **Scale Structure Dimensions (GS):** It is used to scale all dimensions of a structure by a constant.

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

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

5. **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.

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

7. **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.

8. **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 are writen into a text file in particular defined format and create text file with .nec extension.

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

7.3 Implementation Algorithms

Design of Yagi-Uda antenna for optimization is done in three algorithms, first is Fitness Algorithm to design an antenna without any optimization technique, second is the nondominated sorting approach used for ranking selection method, and third is the NSBBO or NSPSO algorithm that is used to apply BBO or PSO technique with nondominated sorting approach for simultaneous, gain-impedance optimization design of Yagi-Uda antenna.

7.3.1 Fitness Algorithm

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 is passed to nec2.exe, otherwise create correct input file as shown in fitness algorithm flow chart 7.1 .
4. In next step, generate output text file with .out extension.
5. Read all characteristics of an antenna that is required for optimization.

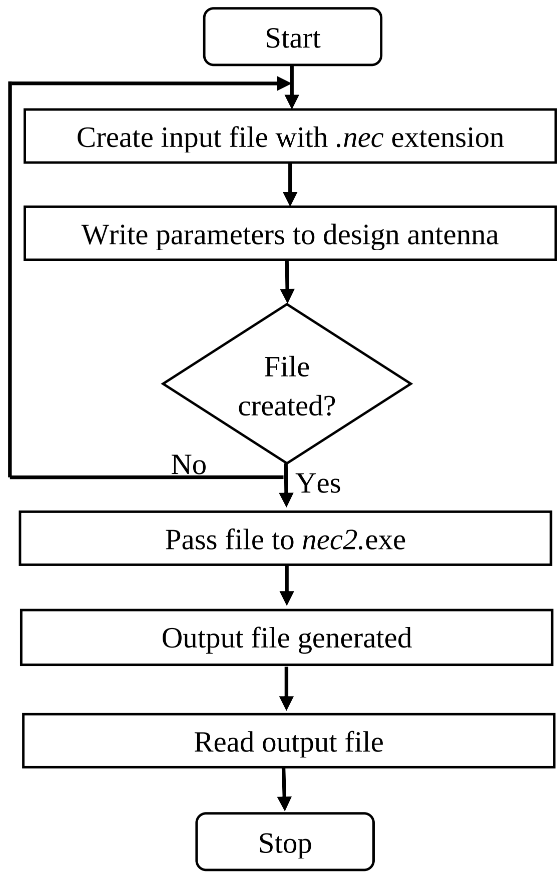


FIGURE 7.1: Flow Chart of Fitness Evaluation Algorithm

7.3.2 Nondominated Sorting Algorithm

The idea behind the nondominated sorting procedure is that ranking selection method is used to emphasize good points, Problem, presented in this thesis work, of optimizing an antenna design has two objectives viz, (i) desired resistive antenna impedance and (ii) maximum antenna gain. Desired antenna impedance, i.e., $(Re + jIm)\Omega$ is formulated as a fitness function, f_1, given as (7.1).

$$f_1 = |Re - \text{desired impedance}| + |Im| \tag{7.1}$$

Whereas, second objective of gain maximization is also converted into minimization fitness function, f_2, given as (7.2)

$$f_2 = \frac{1}{Gain} \tag{7.2}$$

Suppose every solution, in a swarm of NP solutions, yields f_{1_k} and f_{2_k} as fitness values (where $k = 1, 2, \ldots, NP$), using (7.1) and (7.2), that belongs to either non-dominated solution set, P, or dominated solution set, D. An i-th solution in set P dominates the j-th solution in set D if it satisfies the condition of dominance, i.e., $f_{1_i} \leq f_{1_j}$ and $f_{2_i} \leq f_{2_j}$, where both objectives are to be minimized. This condition of dominance is checked for every solution in the universal set of NP solutions to assign it either P set or D set. Solution members of set P form the first non-dominated front, i.e., the pareto optimal front, and then remaining solutions, those belong to set D, are made to face same condition of dominance among themselves to determine next non-dominated front. This process continues till all solutions are classified into different non-dominated fronts, as shown in Figure 3.1. Preference order of solutions is to be based on designer's choice, however, here in this research work euclidian distance is determined from origin for every member solution in a non-dominated front and are picked up in ascending order. The pseudo code of nondominated sorting approach is already depicted in Algorithm 1

7.3.3 NSBBO Algorithm

Algorithmic flow for NSBBO is described below step wise:

1. In first step, identify SIVs and their universe of discourse (UODs).
2. In next step, create a habitat (string) as shown in flow chart of NSBBO algorithm in Figure 7.2.
3. Then generate a random population.
4. After creating a random population, check maximum iteration is done or not. If yes, select a best ranked habitat and stop the NSBBO algorithm. If no, then evaluate fitness.

5. If fitness is achieved then select a best ranked habitat and terminate the NSBBO algorithm. If no, then apply migration process of NSBBO.

6. If fitness is achieved then select a best ranked habitat and terminate the NSBBO algorithm. If no, then apply mutation process of NSBBO.

7. If fitness is achieved then select a best habitat and stop the NSBBO algorithm. If no, then repeat the process from maximum iteration as shown in Figure 7.2

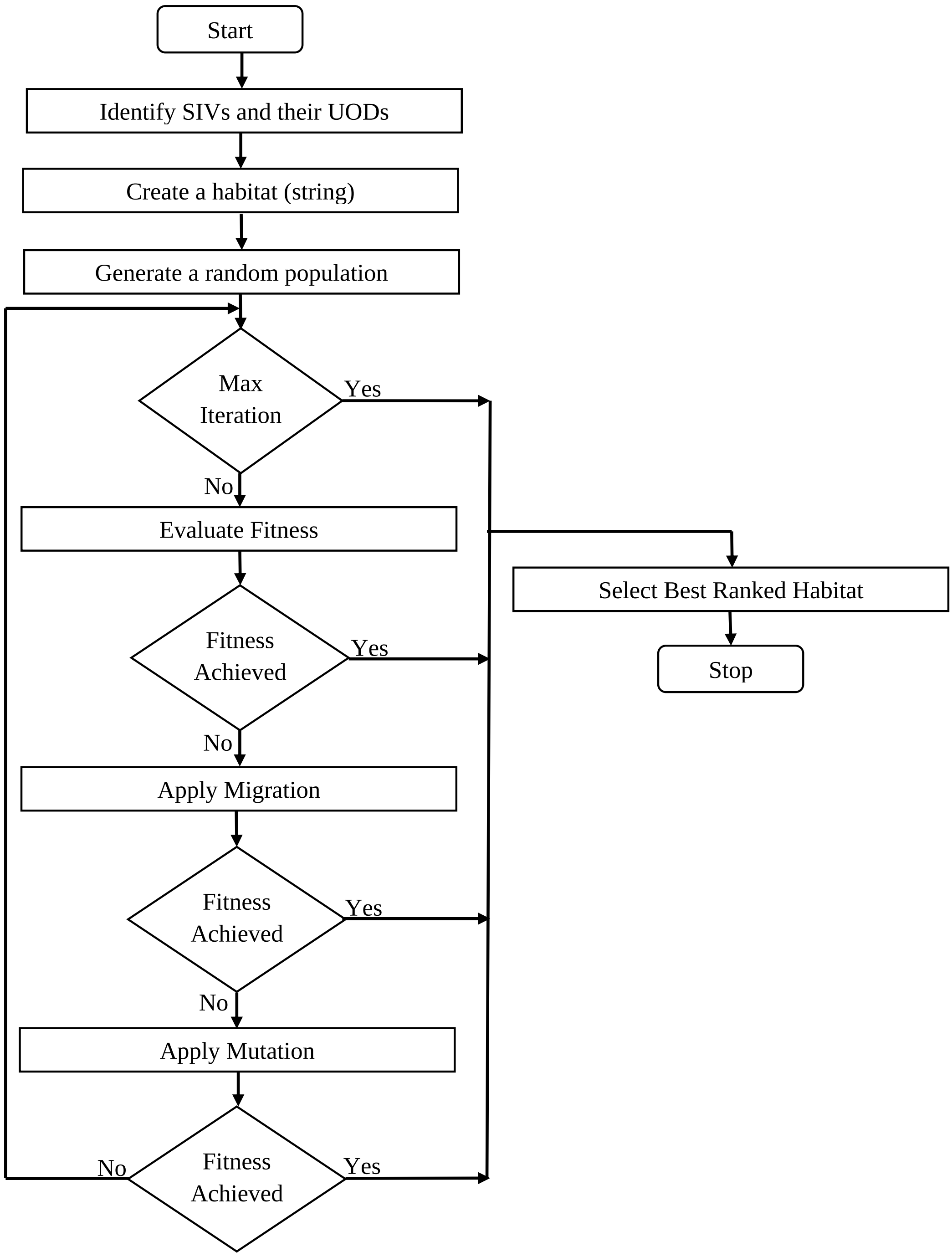


FIGURE 7.2: NSBBO Algorithmic Flow

7.3.4 NSPSO Algorithm

Algorithmic flow for NSPSO is described below step wise:

1. In first step, identify SIVs and their universe of discourse (UODs).
2. In next step, create a habitat (string) as shown in flow chart of NSBBO algorithm in Figure 7.2.
3. Then generate a random population and random velocity.
4. After creating a random population and velocity, evaluate fitness.
5. If fitness is achieved then select a best ranked habitat globally and locally.
6. Then calculate the new modified velocity and new positions for the individuals.
7. Check the new positions obtained, they should not exceed the limits.
8. Make sure if all individual are selcted or not. If no then move back to the initials as shown in Figure 7.2 and if Yes, then proceed further.
9. If convergence is achieved then carry out the iterations with an increment in value.

7.4 Conclusions

In this chapter, various implementation steps of NSBBO algorithms are discussed. Simulation results to analyze convergence performance of NSBBO and its different migration operators and PSO and its variants algorithms are represented in next chapter for the application of optimizing Yagi-Uda antenna design for gain-impedance optimization.

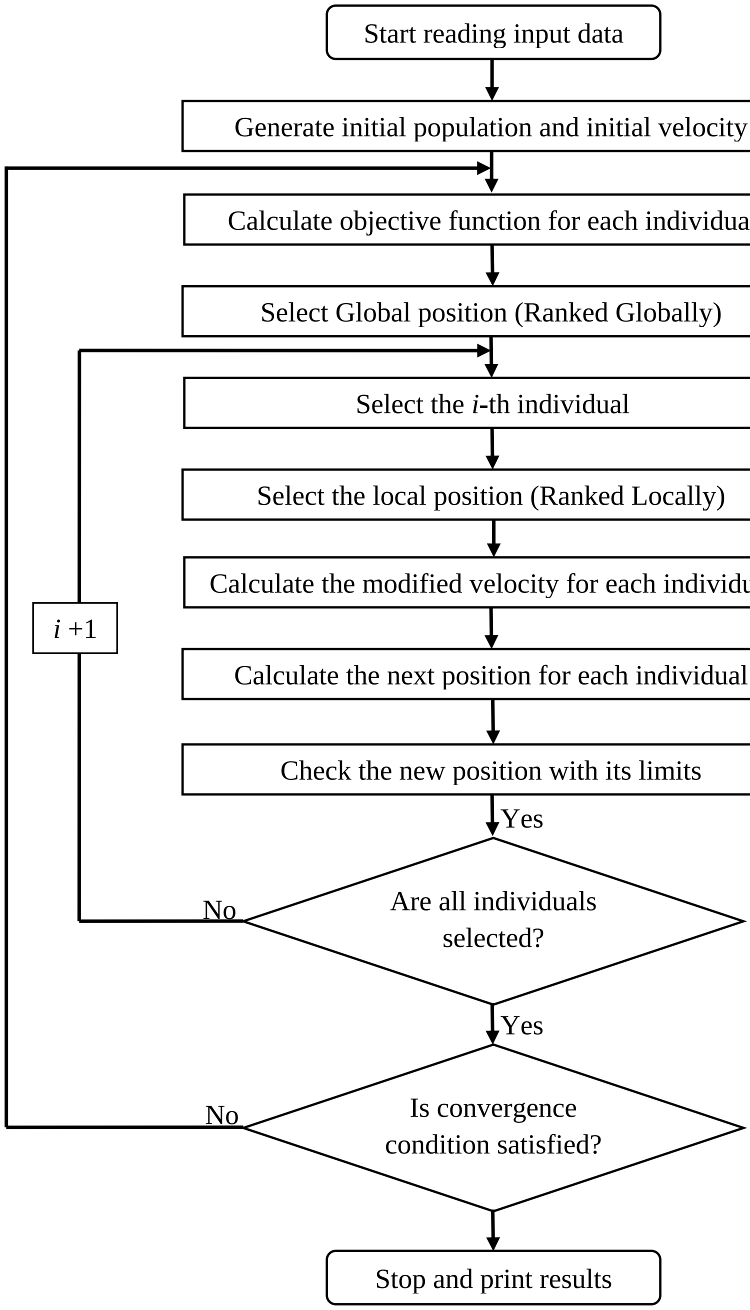


FIGURE 7.3: Diagrammatic Flow of NSPSO Algorithm

CHAPTER 8

RESULTS AND DISSCUSSIONS

As already discussed, design of Yagi-Uda antenna is not an easy task due to large number of geometrical parameters, i.e., wire-lengths and spacings in between them, and their complex relationship for gain, impedance and SLL, etc. C++ programming platform is used for coding of NSBBO and NSPSO algorithms and NEC2, antenna modeling software, is used to determine antenna characteristics like gain and impedance, etc. This chapter presents average of 10 monte-carlo evolutionary runs to conclude, finally, the comparative convergence performance of stochastic NSBBO and NSPSO algorithms.

8.1 Introduction

BBO and PSO are the stochastic search algorithms, therefore, require multiple runs to present fair analysis. Here, NSBBO is proposed and investigated for multiobjective optimization. Different migration variants of BBO discussed in (4.3.1) are explored for better results and made to run for 300 iterations in each case. Comparison for NSBBO and NSPSO results is performed. Every habitats involves 11 SIVs for designing a six-wire Yagi-Uda antenna.

8.2 Simulation Scenario

To present fair analysis, a six-wire Yagi-Uda antenna design is optimized for 10 times using 300 iterations under similar evolutionary conditions. The universe of discourses to search optimal values of wire-lengths and wire-spacings are fixed as $0.40\lambda - 0.50\lambda$ and $0.10\lambda - 0.45\lambda$, respectively. However, cross-sectional radius and segment size for all wires are kept constant, i.e., 0.003397λ and 0.1λ, respectively, where λ is the wavelength corresponding to frequency of operation of 300MHz. The C++ programming platform is used for algorithm coding, whereas, method of moments based software, Numerical Electromagnetic Code (NEC2) [Burke and Poggio, 1981], is used to evaluate antenna designs. Both objectives, gain and impedance, are optimized simultaneously using two fitness functions, given by (7.1) and (7.2).

8.3 Simulation Results

8.3.1 Convergence flow of NSBBO migration variants

8.3.1.1 Convergence flow for 75Ω antenna impedance and maximal gain

Six-wire Yagi-Uda antenna designs are evolved using NSBBO and NSPSO for 75Ω resistive antenna impedance and zero reactive antenna impedance, whose fitness function is given as (7.1) and (7.2).

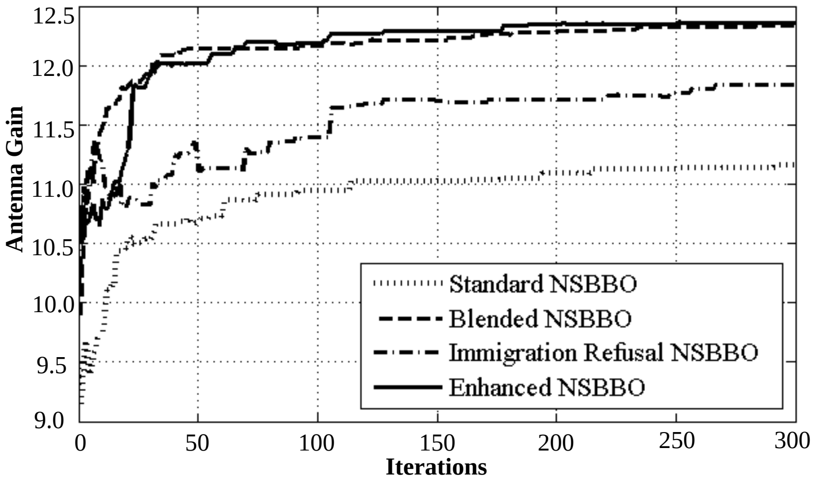
Average of 10 Monte-Carlo simulation runs for 30 habitats for each algorithm are plotted in Figure 8.1 to show convergence flow while achieving (a) maximum antenna gain, (b) 75Ω resistive antenna impedance and (c) zero reactive antenna impedance.

From the plots, it can be observed that EBBO performs better amongst all the migration variants, gives the maximum gain, however, the convergence performance for blended variant is fast as compared to others during initial iterations.

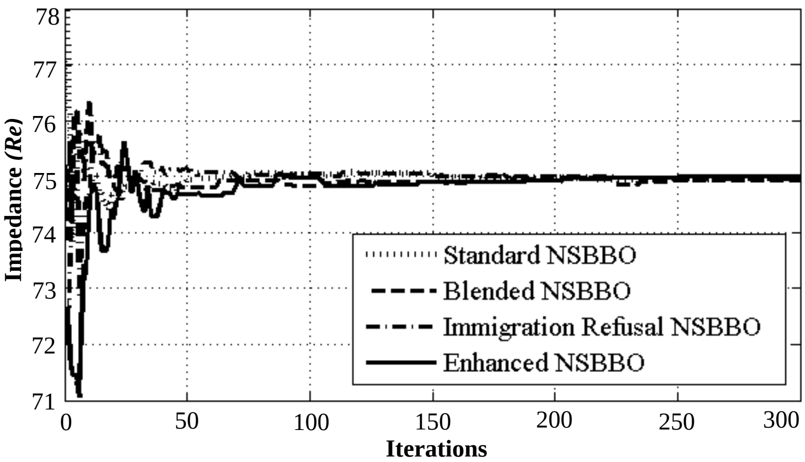
Typically, the best antenna designs evolved and the average results of 10 monte-carlo runs, depicted in Figure. 8.1, are tabulated in Table 8.1, respectively.

TABLE 8.1: The best antenna designs obtained during optimization and average results after 300 iterations for 75Ω impedance

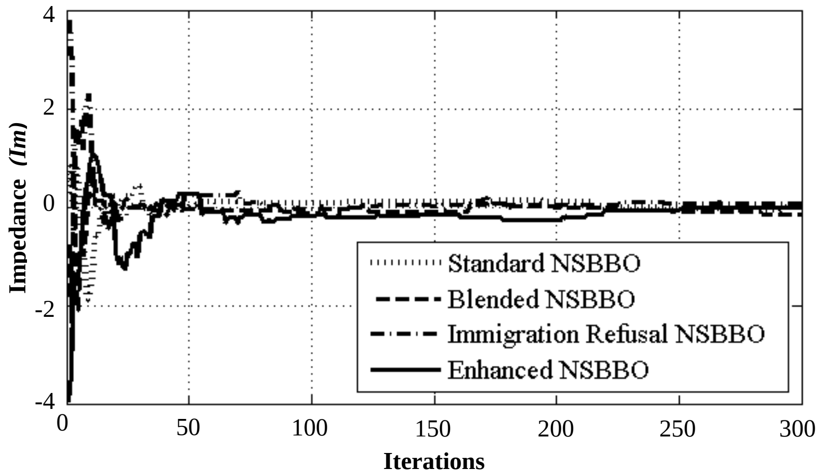
Element	Standard BBO		Blended BBO		IR BBO		EBBO	
	Length	Spacing	Length	Spacing	Length	Spacing	Length	Spacing
$1(\lambda)$	0.4732	-	0.4738	-	0.4652	-	0.4732	-
$2(\lambda)$	0.4780	0.1979	0.4622	0.2185	0.4546	0.2308	0.4693	0.2123
$3(\lambda)$	0.4397	0.1631	0.4417	0.3929	0.4333	0.1239	0.4457	0.1741
$4(\lambda)$	0.4316	0.2735	0.4289	0.6546	0.4249	0.2295	0.4329	0.2484
$5(\lambda)$	0.4193	0.3902	0.4225	1.0289	0.4258	0.3162	0.4221	0.3644
$6(\lambda)$	0.4307	0.3360	0.4283	1.3835	0.4114	0.4423	0.4272	0.3758
Best Gain	**12.58 dBi**		**12.58 dBi**		**12.33 dBi**		**12.63 dBi**	
Best Imp.	74.9414+j0.036 Ω		75.2441-j0.084 Ω		74.9729+j0.077 Ω		75.117+j0.7644 Ω	
Best Abs Imp.	**74.9414 Ω**		**75.2441 Ω**		**74.9729 Ω**		**75.1209 Ω**	
Average Gain	**11.16 dBi**		**12.40 dBi**		**11.843 dBi**		**12.364 dBi**	
Average Imp.	74.946-j0.024 Ω		75.050-j0.073 Ω		74.9633+j0.0763 Ω		74.9971-j0.0155 Ω	
Average Abs Imp.	**74.9460 Ω**		**75.0500 Ω**		**74.9633 Ω**		**74.9971 Ω**	



(a) Antenna Gain Convergence



(b) Resistive Antenna Impedance



(c) Reactive Antenna Impedance

FIGURE 8.1: Average of Convergence flow of NSBBO variant algorithms for 75Ω resistive antenna impedance and maximal gain

8.3.1.2 Convergence flow for 50Ω antenna impedance and maximal gain

Different migration variants of BBO, viz., Standard BBO, Blended BBO, IR BBO and EBBO are experimented for gain maximization and evolving antenna impedance 50Ω, simultaneously.

Average of 10 Monte-Carlo simulation runs with 50 habitats for each variant algorithm are plotted in Figure 8.2 to analyse convergence flow while achieving both objectives.

The convergence performance of standard BBO, EBBO and Blended BBO are comparable. However, IRBBO resulted in poorest performance under same evolutionary conditions and 300 iterations.

Typically, the best antenna designs evolved and the average results of 10 monte-carlo runs depicted in Figure 8.2, are tabulated in Table 8.2, respectively.

TABLE 8.2: The best antenna designs obtained during optimization and average results after 300 iterations for 50Ω impedance

Element	Standard BBO		Blended BBO		IR BBO		EBBO	
	Length	**Spacing**	**Length**	**Spacing**	**Length**	**Spacing**	**Length**	**Spacing**
1(λ)	0.4777	-	0.4764	-	0.4754	-	0.4746	-
2(λ)	0.4700	0.1901	0.4674	0.2168	0.4652	0.2105	0.4653	0.2111
3(λ)	0.4436	0.1826	0.4428	0.1801	0.4419	0.1816	0.4407	0.1994
4(λ)	0.4292	0.2912	0.4272	0.3032	0.4286	0.3068	0.4293	0.2999
5(λ)	0.4239	0.3553	0.4235	0.3401	0.4250	0.3307	0.4233	0.3349
6(λ)	0.4287	0.3475	0.4272	0.3609	0.4280	0.3528	0.4251	0.3719
Best Gain	**12.70 dBi**		**12.68 dBi**		**12.70 dBi**		**12.66 dBi**	
Best Imp.	50.1265-j0.0124 Ω		50.1755-j0.0833 Ω		49.9502+j0.0612 Ω		49.9784-j0.0599 Ω	
Best Abs Imp.	**50.1265 Ω**		**50.1755 Ω**		**49.9502 Ω**		**49.9784 Ω**	
Average Gain	**12.616 dBi**		**12.624 dBi**		**12.282 dBi**		**12.593 dBi**	
Average Imp.	49.9835+j0.0902 Ω		49.9532+j0.00266 Ω		50.00034+j0.03253 Ω		49.9755+j0.08409 Ω	
Average Abs Imp.	**49.9836 Ω**		**49.9532 Ω**		**50.00035 Ω**		**49.9756 Ω**	

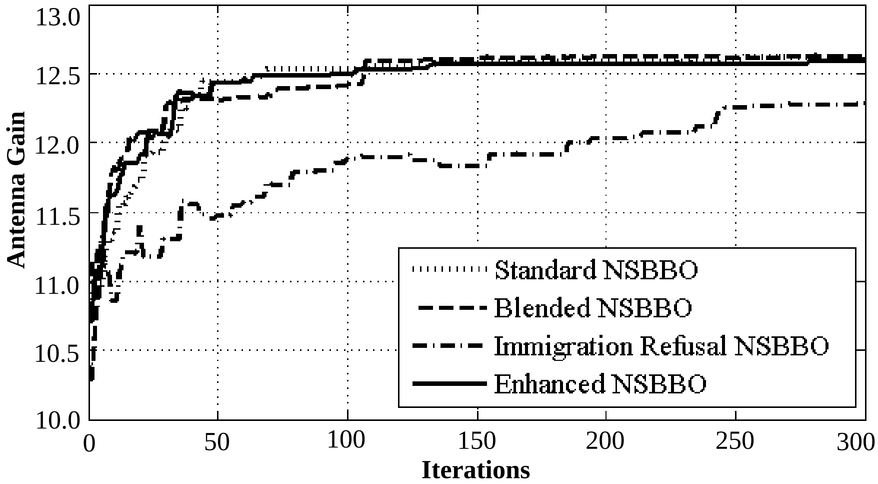
8.3.2 NSBBO Vs NSPSO Convergence flow

8.3.2.1 Convergence flow for 75Ω antenna impedance and maximal gain

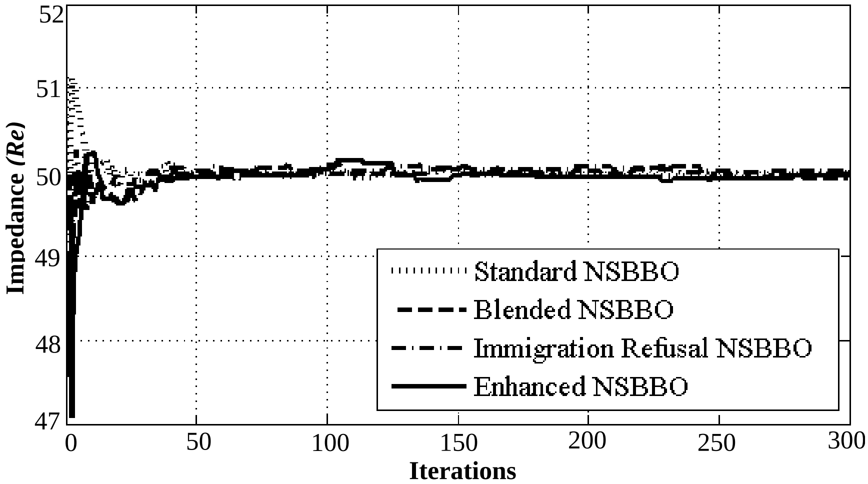
Six-wire Yagi-Uda antenna designs are evolved using NSBBO and NSPSO for 75Ω resistive antenna impedance and zero reactive antenna impedance, whose fitness function is given as (7.1) and (7.2).

Average of 10 Monte-Carlo simulation runs for 30 habitats for each algorithm are plotted in Figure 8.3 to show convergence flow while achieving (a) maximum antenna gain, (b) 75Ω resistive antenna impedance and (c) zero reactive antenna impedance.

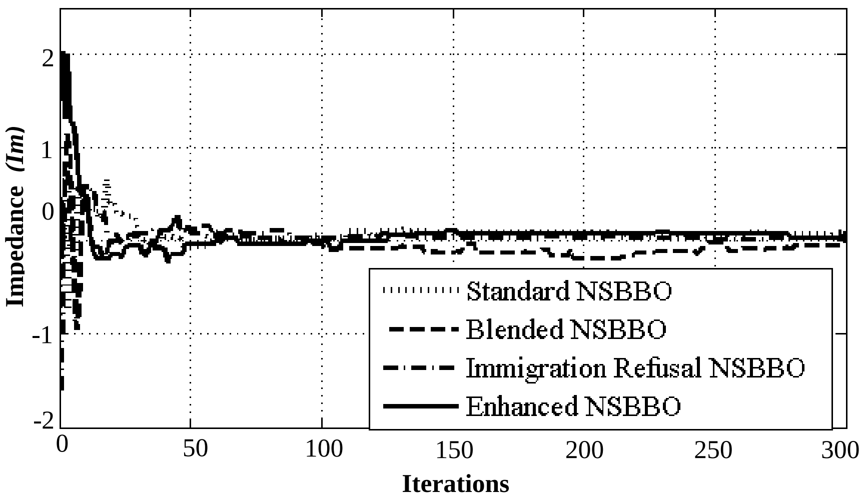
From the plots, it can be observed that best compromised solution, sometimes lead to poor solutions in terms of gain or impedance. However, with increasing iteration number best



(a) Antenna Gain Convergence



(b) Resistive Antenna Impedance



(c) Reactive Antenna Impedance

FIGURE 8.2: Convergence flow for different NSBBO migration variants at 50Ω resistive antenna impedance

compromised solution improves in aggregate that may, improve further, if maximum iteration number is kept higher.

Reasons for poor performance of PSO may include use of global best PSO model, where each particle learns from every other particle in the swarm and globally best particle, therefore, is prone to get trapped in local optima.

Typically, the best antenna designs obtained during process of optimization and the average results of 10 monte-carlo runs, depicted in Figure 8.3, are tabulated in Table 8.3.

TABLE 8.3: The best antenna designs evolved for 75Ω resistive antenna impedance and maximal gain.

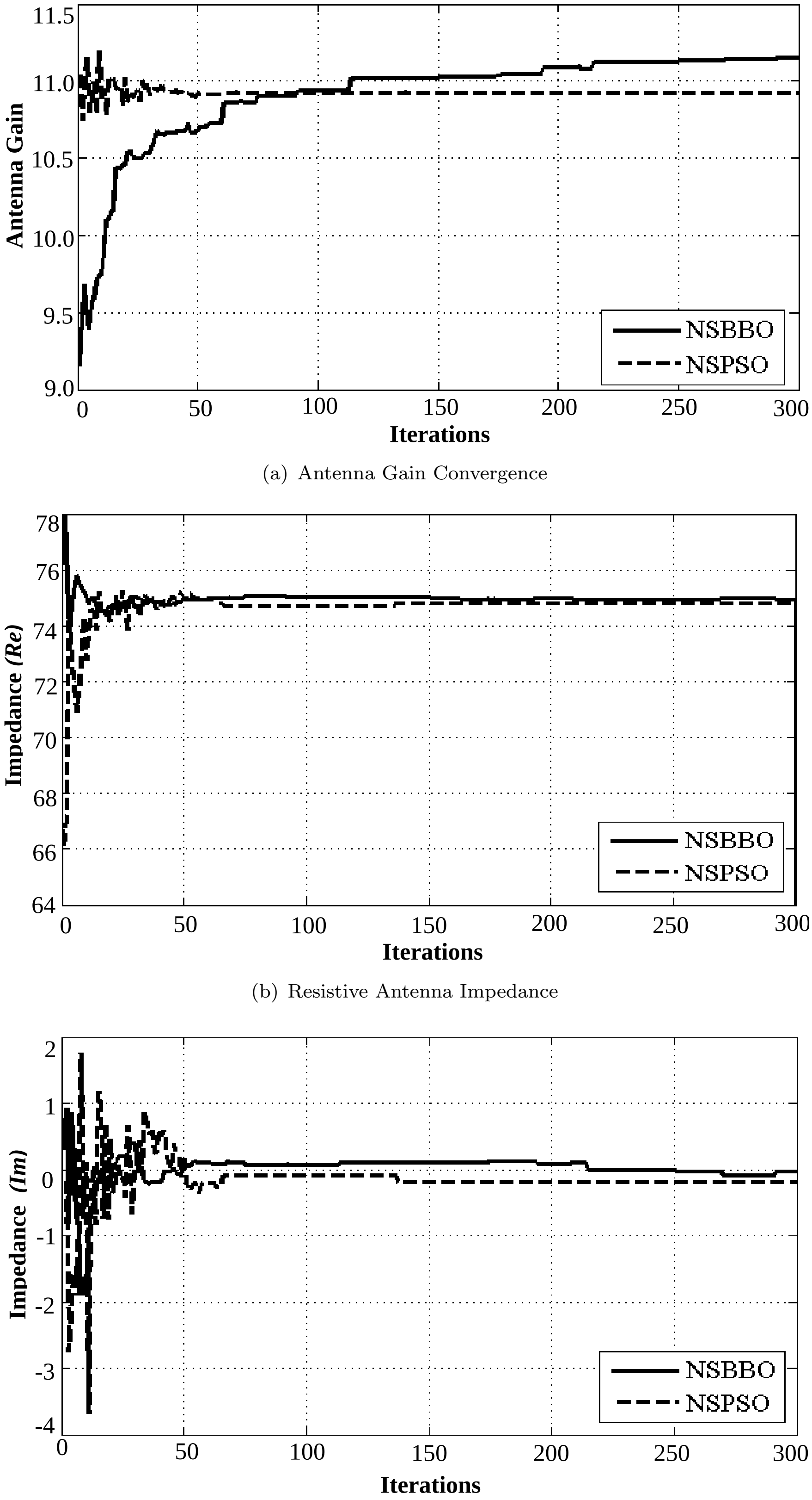
Element	**Standard BBO**		**PSO**	
	Length	**Spacing**	**Length**	**Spacing**
1(λ)	0.4732	-	0.4732	-
2(λ)	0.4780	0.1979	0.4787	0.1953
3(λ)	0.4397	0.1631	0.4396	0.2092
4(λ)	0.4316	0.2735	0.4343	0.2411
5(λ)	0.4193	0.3902	0.4167	0.4353
6(λ)	0.4307	0.3360	0.4334	0.3298
Best Gain	**12.58 dBi**		**12.28 dBi**	
Best Imp.	74.9414 + j 0.0364 Ω		72.903 + j 1.490 Ω	
Best Abs Imp.	**74.9414 Ω**		**72.918 Ω**	
Average Gain	**11.16 dBi**		**10.925 dBi**	
Average Imp.	74.9458 - j 0.0238 Ω		74.8032 + j 0.178 Ω	
Average Abs Imp.	**74.9458 Ω**		**74.8034 Ω**	

8.3.2.2 Convergence flow for 50Ω antenna impedance and maximal gain

Average of 10 Monte-Carlo simulation runs for 50 habitats using NSBBO and NSPSO are plotted in Figure 8.4 to show convergence flow while achieving (a) maximum antenna gain, (b) 50Ω resistive antenna impedance and (c) zero reactive antenna impedance.

From the plots, it can be observed that almost every time NSPSO gets trapped in local optimal for gain objective function and at the same time reactive impedance is capacitive and longer than that of Figure 8.4.

Typically, the best antenna designs for maximal gain and 50Ω antenna gain obtained during process of optimization and the average results of 10 monte-carlo runs, shown in Figure 8.4, are tabulated in Table 8.4.

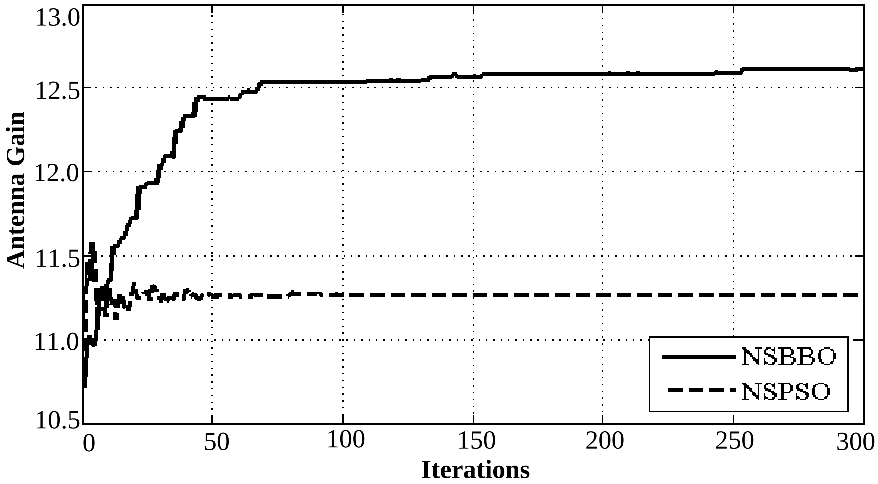


(a) Antenna Gain Convergence

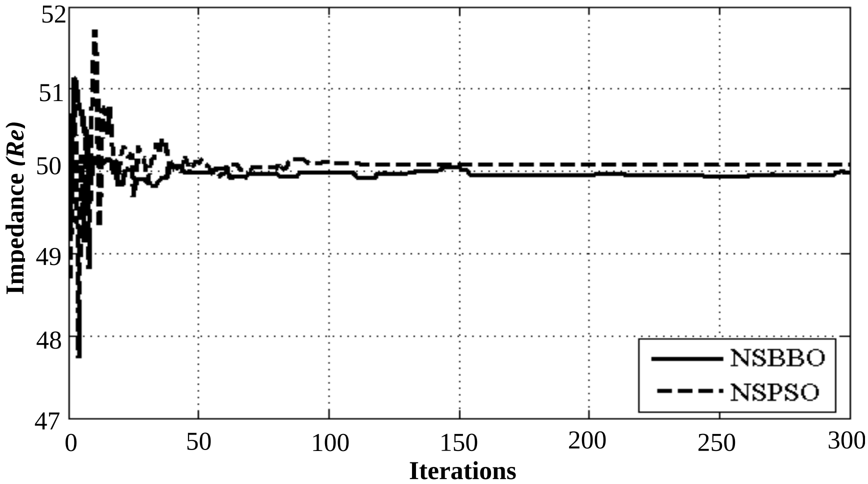
(b) Resistive Antenna Impedance

(c) Reactive Antenna Impedance

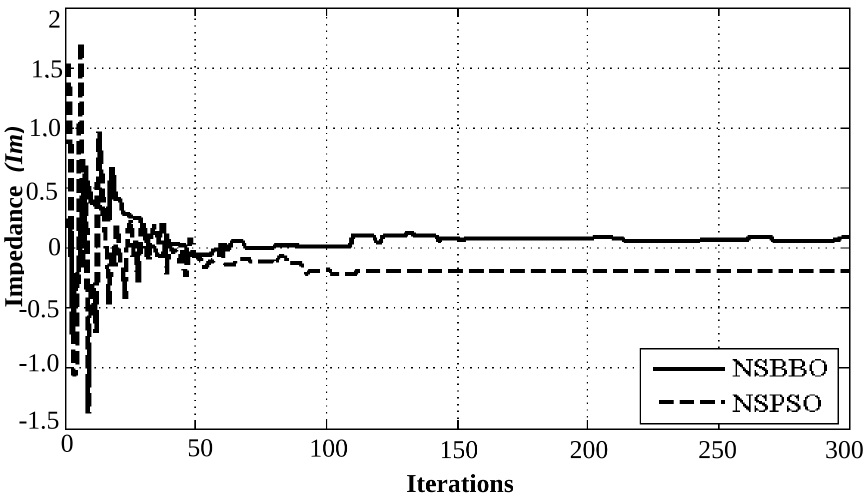
FIGURE 8.3: NSBBO and NSPSO Convergence flow for 75Ω resistive antenna impedance



(a) Antenna Gain Convergence



(b) Resistive Antenna Impedance



(c) Reactive Antenna Impedance

FIGURE 8.4: Average NSBBO and NSPSO Convergence flow for 50Ω antenna impedance and maximal gain

TABLE 8.4: The best antenna designs evolved for 50Ω antenna impedance and maximum gain.

Element	**Standard BBO**		**PSO**	
	Length	**Spacing**	**Length**	**Spacing**
1(λ)	0.4777	-	0.4744	-
2(λ)	0.4700	0.1901	0.4609	0.2025
3(λ)	0.4436	0.1826	0.4350	0.2107
4(λ)	0.4292	0.2912	0.4290	0.3042
5(λ)	0.4239	0.3553	0.4236	0.3418
6(λ)	0.4287	0.3475	0.4224	0.3529
Best Gain	**12.70 dBi**		**12.57 dBi**	
Best Imp.	50.1265 + j 0.0124 Ω		50.562 - j 0.507 Ω	
Best Abs Imp.	**50.1265 Ω**		**50.5645 Ω**	
Average Gain	**12.616 dBi**		**11.263 dBi**	
Average Imp.	49.9835 + j 0.0902 Ω		50.099 - j 0.131 Ω	
Average Abs Imp.	**49.9836 Ω**		**50.099 Ω**	

8.3.3 Convergence flow for BBO Vs gbest and lbest Model of PSO

Average of 10 Monte-Carlo simulation runs are plotted in Figure 8.5 to show convergence flow while achieving maximum antenna gain.

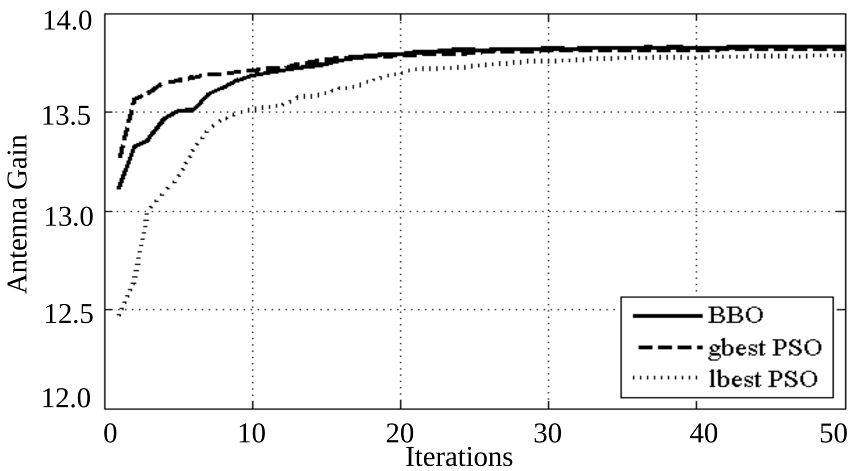


FIGURE 8.5: Convergence flow for BBO vs gbest and lbest PSO models for maximal gain

From the plots, it can be observed that best compromised solution, is shown by BBO. However, the maximum gain is achieved by *gbest* PSO model. Typically, the best antenna designs obtained during process of optimization and the average results of 10 Monte-Carlo runs, shown in Figure 8.5 are tabulated in Table. 8.5.

TABLE 8.5: The best antenna designs evolved for maximum gain.

Element	**BBO**		**gbest PSO**		**lbest PSO**	
	Length	**Spacing**	**Length**	**Spacing**	**Length**	**Spacing**
1(λ)	0.4838	-	0.4862	-	0.4855	-
2(λ)	0.4728	0.1745	0.4766	0.1525	0.4945	0.1623
3(λ)	0.4388	0.2561	0.4401	0.2553	0.4417	0.2423
4(λ)	0.4244	0.3986	0.4271	0.3725	0.4243	0.3931
5(λ)	0.4198	0.4060	0.4184	0.4298	0.4220	0.4271
6(λ)	0.4289	0.3786	0.4282	0.3829	0.4368	0.3381
Best Gain	**13.84 dBi**		**13.85 dBi**		**13.83 dBi**	
Average Gain	**13.83 dBi**		**13.8282 dBi**		**13.79 dBi**	

8.4 Conclusions

In this chapter, simultaneous optimization of gain-impedance of BBO migration operators are experimented and discussed for designing of six-wire Yagi-Uda antenna. The performance comparisons of NSBBO and NSPSO is also performed. In the ending of section average results and the best results are tabulated for overall comparison. The best gain obtain, here in this work is 12.70 dBi which is better than reported in [Singh et al.,2010]. Similar, results can be obtained with exploration of mutation operators. More discussions are presented in the last chapter.

CHAPTER 9

CONCLUSIONS AND FUTURE SCOPE

Research is an iterative process very similar to BBO and PSO 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.

9.1 Introduction

The highlights of this thesis are:

1. NSBBO is proposed and investigated to maximize gain and minimize impedance of a six-wire Yagi-Uda antenna, as multiobjective objective problem, by evolving element lengths and spacings between adjacent elements.
2. NSBBO and its migration variants are experimented for improved convergence performance.
3. NSPSO is investigated to maximize gain and minimize impedance of a six-wire Yagi-Uda antenna, as multiobjective objective problem, by evolving element lengths and spacings between adjacent elements.

4. *gbest* and *lbest* PSO models are experimented for improved convergence performance.

Section 9.2 presents the concluding remarks about what has been investigated, developed, and contribution throughout the work. In Section 9.3, various offshoots of the work are discussed as future research agenda.

9.2 Conclusions

In this thesis, NSBBO is proposed and its variants are investigated for better convergence performance and gain and impedance of Yagi-Uda antenna are optimized simultaneously. Conclusion of this investigation study, as a whole, are discussed. The best results are depicted in Figure 9.1.

Maximum gain of Yagi-Uda antenna achieved during multiobjective optimization using NSBBO and BBO variants is 12.70 dB at targeted 50Ω antenna impedance, as depicted in Table 9.1 observed to be better than that reported in [Singh et al., 2010], i.e., 12.69 dB.

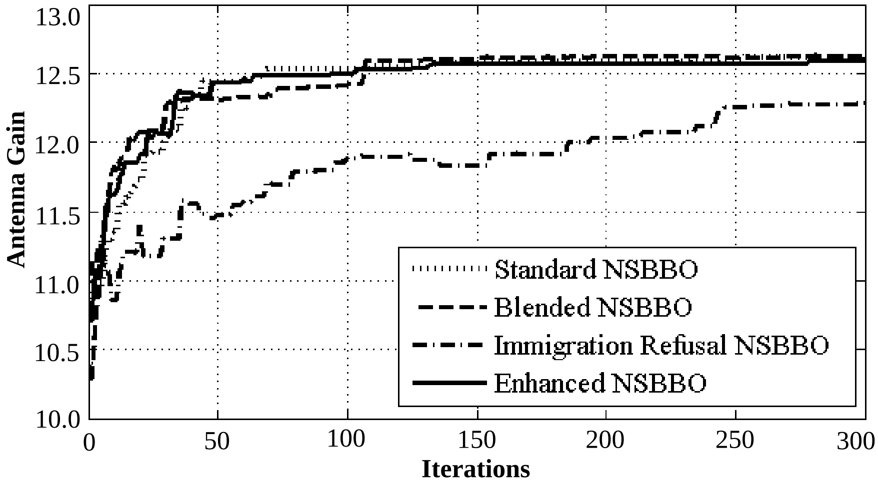
TABLE 9.1: The best antenna designs obtained during optimization and average results after 300 iterations for 50Ω impedance

Element	**Standard BBO**		**Blended BBO**		**IR BBO**		**EBBO**	
	Length	**Spacing**	**Length**	**Spacing**	**Length**	**Spacing**	**Length**	**Spacing**
1(λ)	0.4777	-	0.4764	-	0.4754	-	0.4746	-
2(λ)	0.4700	0.1901	0.4674	0.2168	0.4652	0.2105	0.4653	0.2111
3(λ)	0.4436	0.1826	0.4428	0.1801	0.4419	0.1816	0.4407	0.1994
4(λ)	0.4292	0.2912	0.4272	0.3032	0.4286	0.3068	0.4293	0.2999
5(λ)	0.4239	0.3553	0.4235	0.3401	0.4250	0.3307	0.4233	0.3349
6(λ)	0.4287	0.3475	0.4272	0.3609	0.4280	0.3528	0.4251	0.3719
Best Gain	**12.70 dBi**		**12.68 dBi**		**12.70 dBi**		**12.66 dBi**	
Best Imp.	50.1265-j0.0124 Ω		50.1755-j0.0833 Ω		49.9502+j0.0612 Ω		49.9784-j0.0599 Ω	
Best Abs Imp.	**50.1265 Ω**		**50.1755 Ω**		**49.9502 Ω**		**49.9784 Ω**	
Average Gain	**12.616 dBi**		**12.624 dBi**		**12.282 dBi**		**12.593 dBi**	
Average Imp.	49.9836+j0.0902 Ω		49.9533+j0.0027 Ω		50.0003+j0.0325 Ω		49.9756+j0.0841 Ω	
Average Abs Imp.	**49.9836 Ω**		**49.9532 Ω**		**50.00035 Ω**		**49.9756 Ω**	

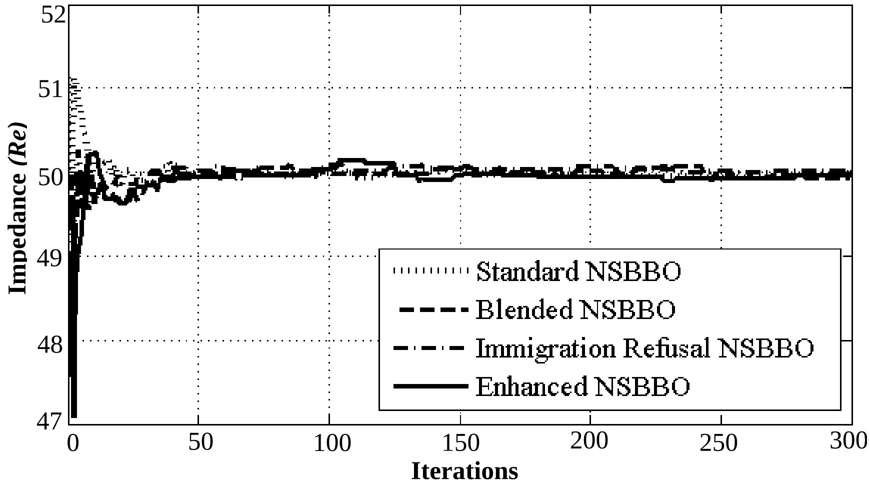
9.3 Future Research Agenda

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

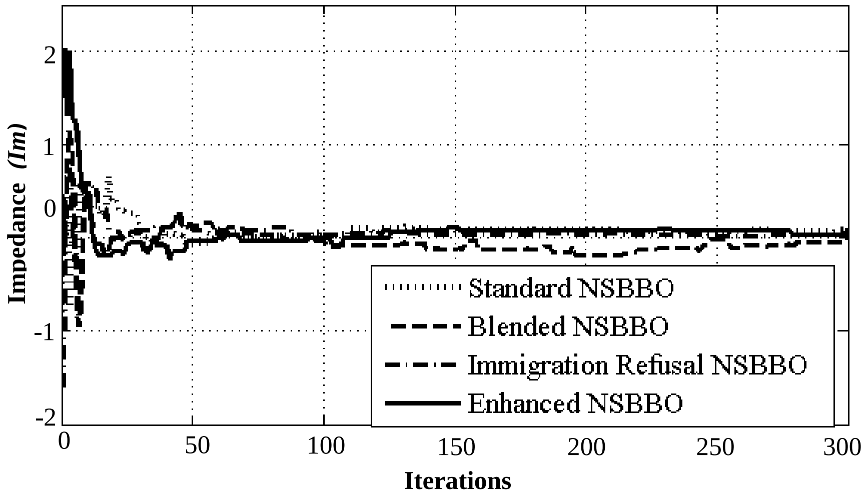
1. In this thesis, Yagi-Uda antenna design problem are considered for optimization, however, different types of antennas, i.e., helical antenna, spida antenna, planar antenna, microstrip antenna, etc., can be designed using NSBBO and NSPSO.



(a) Antenna Gain Convergence



(b) Resistive Antenna Impedance



(c) Reactive Antenna Impedance

FIGURE 9.1: Convergence flow for different NSBBO migration variants at 50Ω resistive antenna impedance

2. Here, design of Yagi-Uda is optimized using NSBBO. Yagi-Uda antenna can be optimized using nondominated sorting approach for AI Techniques, e.g., Artificial Bee colony (ABC), Ant Colony Optimizaton (ACO) and other recently proposed optimization techniques. Antenna can also be optimized using hybridization of EAs.

3. We have used serial programming to develop NSBBO/NSPSO algorithms and for antenna designing that slows down the evolution due to serial execution. For faster evolution, parallel programming paradigm such as Compute Unified Device Architecture (CUDA) and OpenCL etc. can be explored. CUDA and OpenCL is a scalable parallel programming model and a software environment for parallel computing

4. In this thesis, we target multiple objective as gain and impedance of Yagi-Uda antenna. Here, different parameters like SLL and bandwidth, etc can be targeted.

REFERENCES

Andersson, J., Pohl, J., and Krus, P. (1998). Design of objective functions for optimization of multi-domain systems. In *ASME Annual Winter meeting, FPST Division, Anaheim, California, USA*.

Angeline, P. (1998). Evolutionary optimization versus particle swarm optimization: Philosophy and performance differences. In *Evolutionary Programming VII*, pages 601–610. Springer.

Baskar, S., Alphones, A., Suganthan, P., and Liang, J. (2005). Design of yagi-uda antennas using comprehensive learning particle swarm optimisation. In *Microwaves, Antennas and Propagation, IEE Proceedings*, volume 152, pages 340–346. IET.

Benayoun, R., De Montgolfier, J., Tergny, J., and Laritchev, O. (1971). Linear programming with multiple objective functions: Step method (stem). *Mathematical programming*, 1(1):366–375.

Bonabeau, E., Dorigo, M., and Theraulaz, G. (1999). *Swarm intelligence: from natural to artificial systems*. Number 1. OUP USA.

Burke, G. and Poggio, A. (1981). Numerical electromagnetics code (nec) method of moments. part iii: Users guide. *Lawrence Livermore National Laboratory*, (1).

Charnas, A. and Cooper, W. (1961). Management models and industrial applications of linear programming, vol. i.

Charnes, A., Cooper, W. W., and Ferguson, R. O. (1955). Optimal estimation of executive compensation by linear programming. *Management science*, 1(2):138–151.

Chiampi, M., Ragusa, C., and Repetto, M. (1996). Fuzzy approach for multiobjective optimization in magnetics. *Magnetics, IEEE Transactions on*, 32(3):1234–1237.

Chiampi M., G F., C. M. C. R. and R., M. (1998). Multi-objective optimization with stochastic algorithms and fuzzy definition of objective function. *International Journal of Applied Electromagnetics in Materials*, 9:381–389.

Clerc, M. (1999). The swarm and the queen: towards a deterministic and adaptive particle swarm optimization. In *Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on*, volume 3. IEEE.

Clerc, M. and Kennedy, J. (2002). The particle swarm-explosion, stability, and convergence in a multidimensional complex space. *Evolutionary Computation, IEEE Transactions on*, 6(1):58–73.

Coello Coello, C. A. et al. (1996). An empirical study of evolutionary techniques for multi-objective optimization in engineering design.

Darwin, C. (1995). Origin of species. 1995. *Gramercy, New York, Gramercy, USA*.

Das, I. and Dennis, J. E. (1998). Normal-boundary intersection: A new method for generating the pareto surface in nonlinear multicriteria optimization problems. *SIAM Journal on Optimization*, 8(3):631–657.

Deb, K. (2001). Multi-objective optimization. *Multi-objective optimization using evolutionary algorithms*, pages 13–46.

Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: Nsga-ii. *Evolutionary Computation, IEEE Transactions on*, 6(2):182–197.

Du, D., Simon, D., and Ergezer, M. (2009). Biogeography-based optimization combined with evolutionary strategy and immigration refusal. In *Systems, Man and Cybernetics, 2009. SMC 2009. IEEE International Conference on*, pages 997–1002. IEEE.

Eberhart, R. and Kennedy, J. (1995). A new optimizer using particle swarm theory. In *Micro Machine and Human Science, 1995. MHS'95., Proceedings of the Sixth International Symposium on*, pages 39–43. IEEE.

Eberhart, R., Simpson, P., and Dobbins, R. (1996). *Computational intelligence PC tools*. Academic Press Professional, Inc.

Fonseca, C. M. and Fleming, P. J. (1995a). Multiobjective genetic algorithms made easy: selection sharing and mating restriction.

Fonseca, C. M. and Fleming, P. J. (1995b). An overview of evolutionary algorithms in multiobjective optimization. *Evolutionary computation*, 3(1):1–16.

Fonseca, C. M. and Fleming, P. J. (1998a). Multiobjective optimization and multiple constraint handling with evolutionary algorithms. i. a unified formulation. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, 28(1):26–37.

Fonseca, C. M. and Fleming, P. J. (1998b). Multiobjective optimization and multiple constraint handling with evolutionary algorithms. ii. application example. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, 28(1):38–47.

Fourman, M. P. (1985). Compaction of symbolic layout using genetic algorithms. In *Proceedings of the 1st International Conference on Genetic Algorithms*, pages 141–153. L. Erlbaum Associates Inc.

Goldberg, D. E. (1989). Genetic algorithms in search, optimization, and machine learning.

Goudos, S. K., Siakavara, K., Vafiadis, E., and Sahalos, J. N. (2010). Pareto optimal yagi-uda antenna design using multi-objective differential evolution. *Progress In Electromagnetics Research*, 105:231–251.

Grüninger, T. and Wallace, D. (1996). Multimodal optimization using genetic algorithms. *Master's thesis, Stuttgart University.*

Hackwood, S. and Beni, G. (1992). Self-organization of sensors for swarm intelligence. In *Robotics and Automation, 1992. Proceedings., 1992 IEEE International Conference on*, pages 819–829. IEEE.

Hackwood, S. and Wang, J. (1988). The engineering of cellular robotic systems. In *Intelligent Control, 1988. Proceedings., IEEE International Symposium on*, pages 70–75. IEEE.

Hans, A. (1988). Multicriteria optimization for highly accurate systems. *Multicriteria Optimization in Engineering and Sciences*, 19:309–352.

Harik, G. R. (1995). Finding multimodal solutions using restricted tournament selection. In *Proceedings of the Sixth International Conference on Genetic Algorithms*, pages 24–31. San Francisco, CA.

Horn, J. (1997). Multicriterion decision making. *Handbook of evolutionary computation*, 1:F1.

Hwang, C.-L., Paidy, S., Yoon, K., and Masud, A. (1980). Mathematical programming with multiple objectives: A tutorial. *Computers & Operations Research*, 7(1):5–31.

Jones, E. A. and Joines, W. T. (1997). Design of yagi-uda antennas using genetic algorithms. *Antennas and Propagation, IEEE Transactions on*, 45(9):1386–1392.

Kennedy, J. (1999). Small worlds and mega-minds: effects of neighborhood topology on particle swarm performance. In *Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on*, volume 3. IEEE.

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

Kennedy, J. and Eberhart, R. C. (1997). A discrete binary version of the particle swarm algorithm. In *Systems, Man, and Cybernetics, 1997. Computational Cybernetics and Simulation., 1997 IEEE International Conference on*, volume 5, pages 4104–4108. IEEE.

Kennedy, J., Eberhart, R. C., and Shi, Y. (2001). Swarm intelligence. 2001. *Kaufmann, San Francisco.*

Kennedy, J. and Mendes, R. (2002). Population structure and particle swarm performance. In *Evolutionary Computation, 2002. CEC'02. Proceedings of the 2002 Congress on*, volume 2, pages 1671–1676. IEEE.

Kim, J. B. and Wallace, D. (1997). A goal-oriented design evaluation model. In *ASME Design Theory and Methodology conference, Sacramento, CA, USA.*

Kursawe, F. (1991). A variant of evolution strategies for vector optimization. In *Parallel Problem Solving from Nature*, pages 193–197. Springer.

Lei, J., Fu, G., Yang, L., and Fu, D. (2007). Multi-objective optimization design of the yagi-uda antenna with an x-shape driven dipole. *Journal of Electromagnetic Waves and Applications*, 21(7):963–972.

Li, J.-Y. and Guo, J. (2009). Optimization technique using differential evolution for yagi-uda antennas. *Journal of Electromagnetic Waves and Applications*, 23(4):449–461.

MacArthur, R. and Wilson, E. (1967). The theory of biogeography. *Princeton University Press, New Jersey*, pages 19–67.

McTavish, T. and Restrepo, D. (2008). Evolving solutions: The genetic algorithm and evolution strategies for finding optimal parameters. *Applications of Computational Intelligence in Biology*, pages 55–78.

Mendes, R. (2004). *Population topologies and their influence in particle swarm performance.* PhD thesis, Universidade do Minho.

Miettinen, K. (1999). Nonlinear multiobjective optimization, volume 12 of international series in operations research and management science. Kluwer Academic Publishers, Dordrecht.

Morse, J. N. (1980). Reducing the size of the nondominated set: Pruning by clustering. *Computers & Operations Research*, 7(1):55–66.

Osyczka, A. (1984). Multicriterion optimization in engineering with fortran programs. *JOHN WILEY & SONS, INC., 605 THIRD AVE., NEW YORK, NY 10158, USA, 1984, 200.*

Raiffa, H. and Keeney, R. (1976). Decisions with multiple objectives: Preferences and value tradeoffs. *Decisions with multiple objectives: Preferences and Value Tradeoffs.*

Ramos, R. M., Saldanha, R. R., Takahashi, R. H., and Moreira, F. J. (2003). The real-biased multiobjective genetic algorithm and its application to the design of wire antennas. *Magnetics, IEEE Transactions on*, 39(3):1329–1332.

Rattan, M., Patterh, M., and Sohi, B. (2008). Optimization of yagi-uda antenna using simulated annealing. *Journal of Electromagnetic Waves and Applications*, 22(2-3):291–299.

rey Horn, J., Nafpliotis, N., and Goldberg, D. E. (1993). Multiobjective optimization using the niched pareto genetic algorithm. *IlliGAL report*, (93005):61801–2296.

Reynolds, C. W. (1987). Flocks, herds and schools: A distributed behavioral model. In *ACM SIGGRAPH Computer Graphics*, volume 21, pages 25–34. ACM.

Rosenman, M. and Gero, J. (1985). Reducing the pareto optimal set in multicriteria optimization (with applications to pareto optimal dynamic programming). *Engineering Optimization*, 8(3):189–206.

Schaffer, J. D. (1985). Multiple objective optimization with vector evaluated genetic algorithms. In *Proceedings of the 1st international Conference on Genetic Algorithms*, pages 93–100. L. Erlbaum Associates Inc.

Shi, X., Liang, Y., Lee, H., Lu, C., and Wang, Q. (2007). Particle swarm optimization-based algorithms for tsp and generalized tsp. *Information Processing Letters*, 103(5):169–176.

Shi, Y. and Eberhart, R. (1998a). A modified particle swarm optimizer. In *Evolutionary Computation Proceedings, 1998. IEEE World Congress on Computational Intelligence., The 1998 IEEE International Conference on*, pages 69–73. IEEE.

Shi, Y. and Eberhart, R. (1998b). Parameter selection in particle swarm optimization. In *Evolutionary Programming VII*, pages 591–600. Springer.

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

Singh, S. and Sachdeva, G. (2012a). Mutation effects on bbo evolution in optimizing yagi-uda antenna design. In *Emerging Applications of Information Technology (EAIT), 2012 Third International Conference on*, pages 47–51. IEEE.

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).

Singh, U., Kumar, H., and Kamal, T. S. (2010). Design of yagi-uda antenna using biogeography based optimization. *Antennas and Propagation, IEEE Transactions on*, 58(10):3375–3379.

Srinivas, N. and Deb, K. (1994). Muiltiobjective optimization using nondominated sorting in genetic algorithms. *Evolutionary computation*, 2(3):221–248.

Stadler, W. (1979). A survey of multicriteria optimization or the vector maximum problem, part i: 1776–1960. *Journal of Optimization Theory and Applications, Springer*, 29(1):1–52.

Steuer, R. (1986). Multiple criteria optimization: theory, computation, and application. 1986. *Willey, New York*.

Steuer, R. E. and Choo, E.-U. (1983). An interactive weighted tchebycheff procedure for multiple objective programming. *Mathematical programming*, 26(3):326–344.

Suppapitnarm, A., Seffen, K., Parks, G., Clarkson, P., and Liu, J. (1999). Design by multiobjective optimization using simulated annealing. In *International Conference on Engineering Design (ICED 99) Munich, Germany*, volume 33, pages 59–85. Taylor & Francis.

Tamaki, H., Kita, H., and Kobayashi, S. (1996a). Multi-objective optimization by genetic algorithms: A review. pages 517–522.

Tamaki, H., Kita, H., and Kobayashi, S. (1996b). Multi-objective optimization by genetic algorithms: A review. In *Evolutionary Computation, 1996., Proceedings of IEEE International Conference on*, pages 517–522. IEEE.

Tamiz, M., Jones, D., and Romero, C. (1998). Goal programming for decision making: An overview of the current state-of-the-art. *European Journal of operational research*, 111(3):569–581.

Tamura, K. and Miura, S. (1979). Necessary and sufficient conditions for local and global nondominated solutions in decision problems with multi-objectives. *Journal of Optimization Theory and Applications*, 28(4):501–523.

Thurston, D. and Liu, T. (1991). Design evaluation of multiple attributes under uncertainty. *International journal of systems automation and applications,*, 1:143–159.

Trelea, I. C. (2003). The particle swarm optimization algorithm: convergence analysis and parameter selection. *Information processing letters*, 85(6):317–325.

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

Varlamos, P. K., Papakanellos, P., Panagiotou, S., and Capsalis, C. (2005). Multi-objective genetic optimization of yagi-uda arrays with additional parasitic elements. *Antennas and Propagation Magazine, IEEE*, 47(4):92–97.

Venkatarayalu, N. V. and Ray, T. (2003). Single and multi-objective design of yagi-uda antennas using computational intelligence. In *Evolutionary Computation, 2003. CEC'03. The 2003 Congress on*, volume 2, pages 1237–1242. IEEE.

Venkatarayalu, N. V. and Ray, T. (2004). Optimum design of yagi-uda antennas using computational intelligence. *Antennas and Propagation, IEEE Transactions on*, 52(7):1811–1818.

Wallace, A. (2005). The geographical distribution of animals (two volumes). 2005. *Adamant Media Corporation, Boston, MA*, pages 232–237.

Wallace, D. R., Jakiela, M. J., and Flowers, W. C. (1996). Design search under probabilistic specifications using genetic algorithms. *Computer-Aided Design*, 28(5):405–421.

Wang, H., Man, K., Chan, C., and Luk, K. (2003). Optimization of yagi array by hierarchical genetic algorithms. In *Radio and Wireless Conference, 2003. RAWCON'03. Proceedings*, pages 91–94. IEEE.

Yagi, H. (1928). Beam transmission of ultra short waves. *Proceedings of the institute of radio engineers*, 16(6):715–740.

Zimmermann, H.-J. and Sebastian, H.-J. (1995). Intelligent system design support by fuzzy-multi-criteria decision making and/or evolutionary algorithms. In *Fuzzy Systems, 1995. International Joint Conference of the Fourth IEEE International Conference on Fuzzy Systems and The Second International Fuzzy Engineering Symposium., Proceedings of 1995 IEEE International Conference on*, volume 1, pages 367–374. IEEE.

Zitzler, E. and Thiele, L. (1999). Multiobjective evolutionary algorithms: A comparative case study and the strength pareto approach. *Evolutionary Computation, IEEE Transactions on*, 3(4):257–271.

INDEX