UNIVERSITY OF CALIFORNIA

Los Angeles

Brain Storm Optimization for Electromagnetic Applications

A thesis submitted in partial satisfaction

of the requirements for the degree

Master of Science in Electrical Engineering

by

Alwaleed A F A F S Aldhafeeri

2018

ABSTRACT OF THE THESIS

Brainstorm Optimization for Electromagnetic Applications

by

Alwaleed A F A F S Aldhafeeri

Master of Science in Electrical Engineering
University of California, Los Angeles, 2018
Professor Yahya Rahmat-Samii, Chair

Brain Storm Optimization (BSO) is a swarm intelligence optimization algorithm inspired by the collective behavior of human beings in solving problems, i.e., brainstorming process. The algorithm was proposed by Y. Shi in 2011 inspired by such a behavior. Brainstorming is used when there is a difficult problem that cannot be solved by a single individual. Based on several tests, BSO has demonstrated great results to be a powerful optimization tool that can be used in electromagnetics and antenna engineering. To the best of our knowledge, this is the first time that BSO is applied to electromagnetic problem. BSO has been applied to common mathematical benchmark functions, the standard method of testing optimization algorithms. For electromagnetics and antenna applications, several representative examples such as a Yagi-Uda antenna, a Luneburg lens and others have been optimized using both BSO and PSO. Also, a design was optimized, prototyped and measured. Additionally, a novel binary version of BSO is proposed and applied with success to multiple antenna designs.

The thesis of Alwaleed A F A F S Aldhafeeri is approved.

Tatsuo Itoh

Lieven Vandenberghe

Yahya Rahmat-Samii, Committee Chair

University of California, Los Angeles

2018

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Acknowledgments

I would like to thank Prof. Tatsuo Itoh and Prof. Lieven Vandenberghe for their time and effort spent as part of my thesis committee. I am also forever grateful to my advisor Prof. Yahya Rahmat-Samii for his guidance, time, and effort throughout my Master’s study. I would also like to thank Jordan Budhu and Lingnan Song at UCLA Antenna Research, Analysis, and Measurement Laboratory for their help with the measurements in this thesis.

CHAPTER 1

1 Introduction

The goal of this first chapter is to provide a motivation behind the work in this thesis and the importance and the need of utilizing optimization tools. Then, a brief overview of global optimization will be given. Finally, we explain global optimization in the context of electromagnetic applications, and how one can use global optimization tools in conjunction with numerical electromagnetic solvers to obtain a good design.

1.1 Motivation

Engineers are challenged to push technology further every day. The challenge is to provide either a new design; or improve and modify already existed designs. In both cases, designers strive for the most optimum designs. With the growing power of computers, engineers are tempted to use tools that enables them to push the technology even further by utilizing the massive computation power. Besides the tools that enable engineers to model and simulated a very complex designs, the tools to optimize those designs will aid engineers to reach the best solutions to the most complex problems. The use of trial-and-error techniques can be very time consuming, especially, when the effect of different design parameters is unknown. Indeed, this is the case in electromagnetics and antenna engineering where it is usually hard to predict how sensitive the design to certain parameters and to what extent they affect the results. Additionally, solving an electromagnetic structure can take from few minutes to several hours depending on the size and complexity of the structure. Thus, the tools of global optimization provide a more systematic and better mechanism of finding the optimum designs.

1.2 Global Optimization

An optimization is simply finding values of variables to obtain the optimum (maximum or minimum) of a fitness function of those variables that may or may not be subject to certain restrictions. The global optimization problem can be formulated as [1]

$$\begin{array}{ll} \text{Optimize (Minimize or Maximize)} & f(x), \quad x \in \mathbf{R}^D \\ \text{Subject to} & g_i(x) \leq b_i\,, \quad i = 1,\ldots,m \end{array} \tag{1.1}$$

- $x = (x^1,\ldots,x^D)$ is a vector that contains the *optimization variables*.
- $f(x)$ is the *fitness* or *objective function*.
- $g_i(x)$ are *constraint functions*.

For simplicity, in this work the optimization scheme will take the form of minimization such that the x^* is called optimal or a solution of Eq. (1.1) if it has the smallest fitness value compared to other evaluated vectors. In most optimization applications, x is bounded between a lower bound $x_{\min}$ and upper bound $x_{\max}$. This results into a bounded D-dimensional *search* or *solution space*, and no solution outside of the search space will be tested. Moreover, inside the search space the solution may need to satisfy additional constraints $g_i(x)$ that would define what so called *a feasible space* [2]. There are different ways of handling constraints and solutions outside the search space. In this work, solutions outside the feasible space will be treated the same as solutions outside the search space; those solutions will not be evaluated. Instead, there fitness function will return a very large value, when the optimizer is a minimizer, to indicate how bad the solution is. The reason behind this approach is sometimes evaluating the fitness function can be time expensive; a huge time can be saved using this type of handling. This mechanism of handling solution outside the bounded search space is usually referred to as *invisible* boundary condition.

There are various classes and types of optimization; however, the optimization of interest in this work is *global optimization*, which is a subclass of *nonlinear optimization*. Another subclass of nonlinear optimization is *local optimization*. As the name indicates, local optimization is used to find a point that is locally optimal. This means that point does not necessarily have the lowest fitness value in the whole search space; i.e., this point is not guaranteed to be a global optimum. For example, x_1 in Figure 1-1 is a local minimum in its neighborhood. Also, there exist a point, x_2, that has lower fitness value. Another downside of local optimization method is that they require differentiability of the fitness function, and highly dependent on their initialization [2].

On the other hand, global optimization methods try to find a point that is the global optimum of the objective function such as x_2 in Figure 1-1. Moreover, other advantages of global optimization are: derivatives are not required in global optimization; variables of optimization can be discrete; and a prior knowledge of the fitness function is not required. With the explosive growth

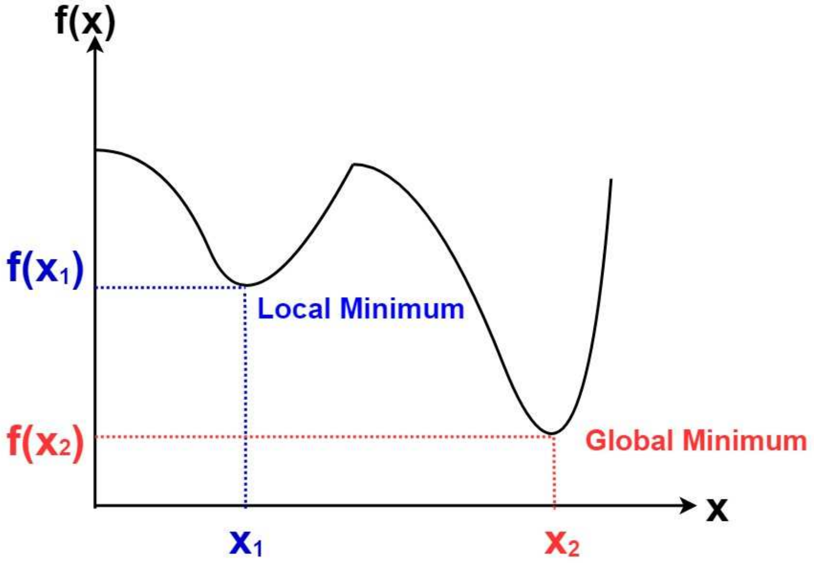


Figure 1-1: Global and local optima of a one-dimensional function

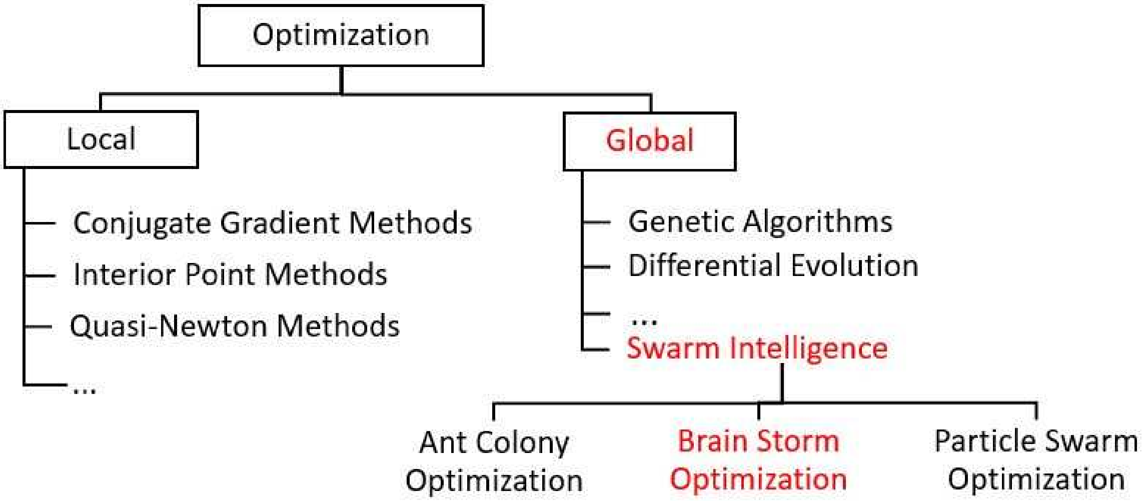


Figure 1-2: A sample of optimization algorithms

of the computation power, various optimization algorithms have been proposed, and a very short list of them can be seen in Figure 1-2.

1.3 Optimization in Electromagnetics

It has been stated in section (1.2) that the fitness function dictates whether a certain solution is better than other. In electromagnetics, the fitness function represents the desired characteristics of the system in the optimization. The system could be an antenna, a microwave circuit, an object for radar applications. In antenna systems, an optimization could be used to achieve certain pattern characteristics such as Directivity, Sidelobe Level, Beamwidth, etc. In microwave circuits, an optimization could be used to improve network performance i.e. Gain, Reflection coefficients, Isolation, etc.

The goal of optimization could be to achieve either a single criterion (objective) or multiple criteria (objectives). In the case of having only a single objective, the fitness function is rather simple which is the single objective of the optimization. For example, the objective is to optimize the impedance matching of an antenna. The impedance matching is represented by the reflection coefficient Γ which is computed in general as [3]

$$\Gamma = \frac{Z_L - Z_s}{Z_L + Z_s} \tag{1.2}$$

where Z_L is the impedance of the load, and Z_s is the impedance of the source. For antennas, the previous equation can be written as

$$\Gamma = \frac{Z_{in} - Z_0}{Z_{in} + Z_0} \tag{1.3}$$

where Z_{in} is the input impedance of the antenna, and Z_0 is the characteristic impedance of the transmission line that delivers the power to the antenna. It is common to use the reflection coefficient in a decibel scale such that

$$\Gamma_{dB} = 20\log_{10}|\Gamma| \tag{1.4}$$

Therefore, the single-objective fitness function can be simply written as

$$f = \Gamma_{dB} \tag{1.5}$$

On the other hand, when there is more than one goal in the optimization, the simple and most common case is casting the multiple objectives into a single fitness function by using a weighted sum of the different objectives. For example. When, optimizing an antenna to achieve a lower sidelobe level while maintaining a high forward directivity, three are various way to write the fitness function. A simple way is to write the fitness function when the optimization is a minimizer as

$$f = (-w_1 * D) + (w_2 * SLL) \tag{1.6}$$

where $w_{1,2}$ are weights of the objectives and it is chosen based on the application and the user preference. The minus sign in the first term is due the optimization is a minimizer, and the goal is to obtain a higher directivity D. Another way of writing the fitness function is as

$$f = w_1 * (D_{goal} - D) + w_2 * SLL \tag{1.7}$$

where D_{goal} is the directive that the user wishes to obtain. In the case of having multiple objectives, the optimization could be very sensitive to the choice of fitness function [1]. Since in most electromagnetic applications one cannot predict the behavior of the fitness function, the fitness function is usually developed through experimenting such that it represents the relative importance of each desired objective [4].

It can be seen from the previous discussion that the fitness function provides the interface between the physical problem and the optimization algorithm. The electromagnetic parameters needed to compute the fitness function such in Eq. (1.7) are usually obtained by solving Maxwell's equations for the problem of interest.

$$\nabla \times \vec{E} = -\frac{\partial \vec{B}}{\partial t} \tag{1.8}$$

$$\nabla \times \vec{H} = \vec{J} + \frac{\partial \vec{D}}{\partial t} \tag{1.9}$$

$$\nabla \cdot \vec{D} = \rho \tag{1.10}$$

$$\nabla \cdot \vec{B} = 0 \tag{1.11}$$

To solve Maxwell's equations, one needs to apply numerical technique for the problems of interest [5]. Examples of numerical techniques in electromagnets are the Finite Element Method [6, 7] , the Finite-Difference Time Domain [8, 9], and the Method of Moments [10, 11]. Therefore, combining the tool of optimization and the tool of numerical solvers can yields a huge benefit for microwave and antenna engineers. Figure 1-3 demonstrate an example of the optimization process in electromagnetics by optimizing a patch antenna to achieve a good impedance matching at the desired frequency f_c. Also, the flowchart shows how the two tools of optimization and electromagnetic solvers can be linked using an interface program.

The use of optimization tool is very common in electromagnetics. Various optimization algorithms have been used in the field that yielded very exciting results. Examples of global optimization algorithms that have been successfully applied in various electromagnetic applications are shown below:

- Genetic Algorithm [12, 13]
  - Yagi-Uda array [14].
  - Array Thinning [15].
  - Nonuniform Luneburg and Two-Shell Lens Antennas [16].
  - RCS reduction of canonical targets using synthesized RAM [17].
- Particle Swarm Optimization [4, 18, 19, 20, 21, 22]
  - Yagi-Uda Array [23].
  - Reconfigurable E-Shaped Patch Antenna [24, 25]
  - Optimization of a Spline-Shaped UWB Antenna [26].
- Invasive Weed Optimization [27]
  - Printed Yagi Antenna [28].
  - Broadband Cosecant Squared Pattern Reflector Antenna [29].
  - Large array synthesis [30].

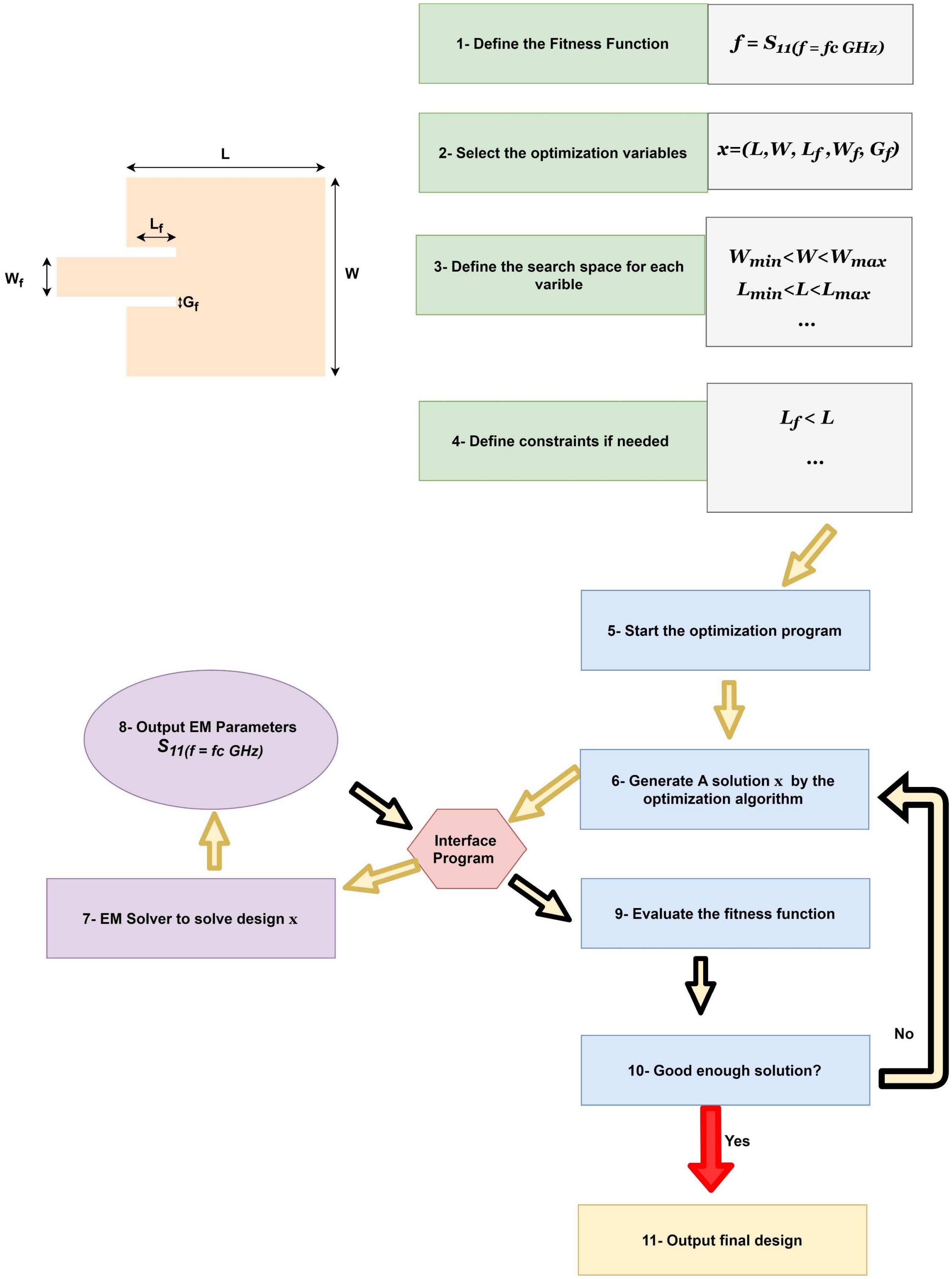

1- Define the Fitness Function
$f = S_{11(f = fc\ GHz)}$
2- Select the optimization variables
$x=(L,W, L_f ,W_f, G_f)$
3- Define the search space for each varible
$W_{min}<W<W_{max}$
$L_{min}<L<L_{max}$
...
4- Define constraints if needed
$L_f < L$
...
L
L_f
W_f
G_f
W
5- Start the optimization program
6- Generate A solution x by the optimization algorithm
Interface Program
7- EM Solver to solve design x
8- Output EM Parameters $S_{11(f = fc\ GHz)}$
9- Evaluate the fitness function
10- Good enough solution?
No
Yes
11- Output final design

Figure 1-3: Flowchart of optimizing a patch antenna

1.4 Outline of Work

A list of the main topics that will be presented in this thesis is shown in Figure 1-4. We started this chapter (Chapter 1) with a brief introduction of the concept of global optimization and how it can be used in the context of electromagnetic application. That was followed by a brief history of applying global optimization algorithms for electromagnetic applications. Moreover, a detailed overview of the process of using global optimization tools in conjunction with numerical solvers to obtain a satisfactory design was provided in this chapter. To the best of our knowledge, this is the first time that BSO is applied to electromagnetic problem. Hence, Chapter 2 will introduce the Brain Storm Optimization (BSO) algorithm. It will start with concept behind the algorithm. Then, it will provide detailed steps of the BSO algorithm. Moreover, definitions of the common terms to be used in the algorithm will be provided to help understand the algorithm more effectively.

Chapter 3 will compare BSO with the Particle Swarm Optimization algorithm. The difference between how the two algorithms work is explained as well as the difference in terminology between the two. Then, both algorithms will be applied to common mathematical benchmark functions, the standard method of testing optimization algorithms. In chapter 4, a six-element Yagi-Uda antenna will be optimized using both BSO and PSO to demonstrate the performance of BSO, and its potential for electromagnetic applications. Chapter 5 will demonstrate the same idea, but with a nonuniform Luneburg Lens antenna in different configurations.

Chapter 6 will introduce a novel binary version of BSO, and the process of transforming the concept of BSO into a binary language. The new parameters introduced along with the binary BSO (BBSO) will be explained as well as the selection of their optimum values. In chapter 7,

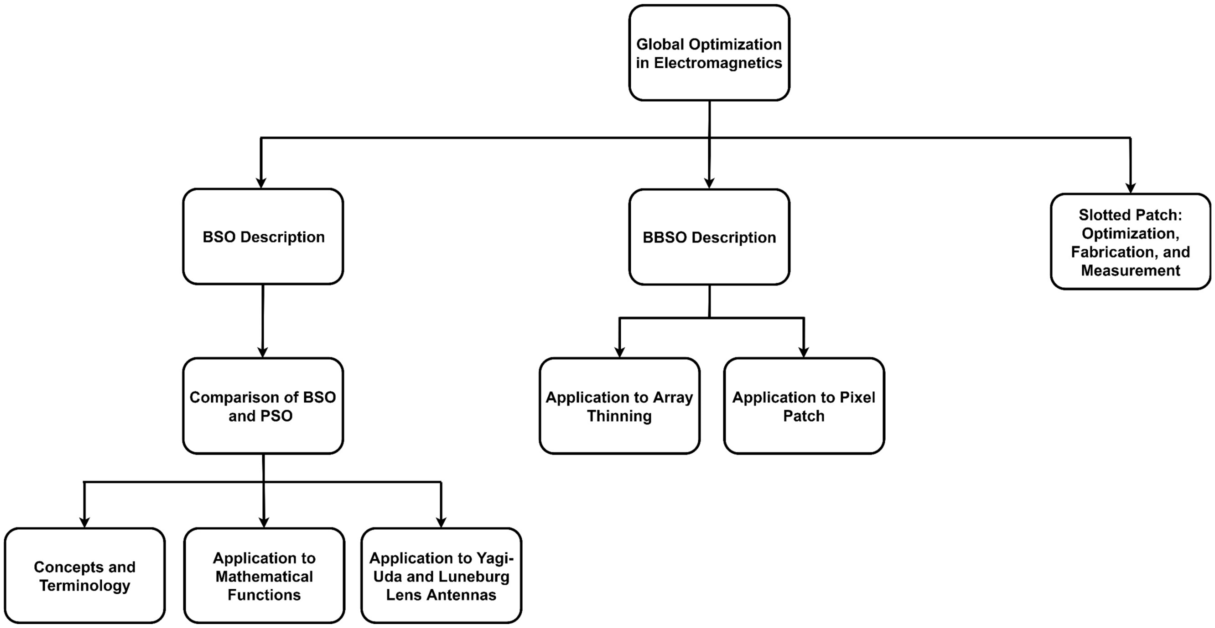

Global Optimization in Electromagnetics
BSO Description
BBSO Description
Slotted Patch: Optimization, Fabrication, and Measurement
Comparison of BSO and PSO
Application to Array Thinning
Application to Pixel Patch
Concepts and Terminology
Application to Mathematical Functions
Application to Yagi-Uda and Luneburg Lens Antennas

Figure 1-4: Outline of work in this thesis

BBSO will be applied to a two-dimensional array thinning problem where the goal is to minimize the sidelobe levels to demonstrate the potential of BBSO of optimizing binary-valued variables. In Chapter 8, BBSO is applied to a patch antenna problem, where the patch is divided into discrete number of elements. The goal is to achieve a dual-band patch antenna by removing those elements from the patch.

Finally, in Chapter 9, BSO is applied to a slotted patch antenna, then the optimized design is fabricated and measured. The goal is to achieve a dual-band patch antenna. This chapter shows how BSO can be used in the full design process that includes the use of global optimization tools, numerical electromagnetics solvers as simulation tools, design fabrication, and finally measuring the fabricated design. The final chapter, Chapter 10, will involve the final discussion of the results obtained throughout this thesis.

CHAPTER 2

2 Brainstorm Optimization

Brain Storm Optimization (BSO) is a swarm intelligence optimization algorithm inspired by the collective behavior of human beings in solving problems i.e. brainstorming process. The algorithm was proposed by Yuhui Shi in 2011 inspired by the behavior of the most intelligent creature [31]. Brainstorming is a common method used when a there is a difficult problem that cannot be solved by a single individual. The session starts with a group of people where each person contributes in generating ideas. Naturally, everyone has a different approach in solving the problem creating a set of diverse ideas. Therefore, the probability of solving the problem will increase as this interactive collaboration continues. In this process, ideas are generated either mostly based on the best existing ideas or as completely new ideas. In optimization, an idea represents a point in a search space. The BSO algorithm has three main operations; Grouping, Replacing, and Selecting. The concept of BSO is built on clustering or grouping similar ideas together and selecting the best idea in each cluster(group) as the cluster center. This will divide the search space into several sub-spaces; each cluster center represents a local optimum point in a local search space. In the Replacing operation, a cluster center is randomly chosen and replaced by a new random idea. In the Selecting operation, a cluster is selected according to the number of ideas it contains. The more ideas a cluster contains the higher the chance that cluster is selected. The selected idea in the selected cluster does not necessarily have to be the cluster center; sometimes, ideas that are not their cluster centers are selected. Ideas can also be generated based on two randomly chosen clusters instead of one cluster. Finally, new ideas are created by adding Gaussian noise to the selected ideas. The application of BSO in multiple fields has shown to be successful; examples of those applications includes:

- Optimal satellite formation reconfiguration [32].
- The design of DC Brushless Motor [33].
- Economic dispatch considering wind power [34].
- Wireless sensor networks [35].

2.1 The Concept Behind BSO

There are many tools and methods the are being used by various organizations to solve problems they face. One of the most common methods is brainstorming. Brainstorming is a widely used method to facilitate creative thinking among a group of people. Brainstorming was first developed and coined by Osborn in 1939 in his advertising company. He introduced his method of creative problem-solving in his book *Your Creative Power* [36]. Late in 1957, Osborn created a complete system and a detailed framework of brainstorming in his book *Applied Imagination* [37, 38].

When a company faces difficult problems that cannot be solved by a single individual, the company will call a group of people to brain storm. With each individual, from different background, contributing in generating ideas using different approaches, the probability of solving the problem will increase as this interactive collaboration continues. To illustrate this behavior with a simple example,

1. Imagine person A and person B are facing a problem. They both cooperate to solve this problem.
2. Person A comes up with idea X and person B comes up with idea Y.
3. They compare ideas and decide that person's A idea X is better.

4. Then, both persons will come up with ideas based on idea X; compare new ideas; decide which is better to come up with even more ideas. Sometimes, they combine their ideas for better solutions.

5. Repeat until a satisfactory solution is obtained.

The completer framework and process of brainstorming suggested in [31] is shown in Table 2-1.

In a brainstorming process, the brainstorming group must obey the Osborn's original four rules of idea generation in a brainstorming process. Osborn's original rules are shown in Table 2-2 [31]. The purpose of those rules is to generate ideas as diverse as possible. Rule 1 says there is no such thing as a bad idea. This rule encourages people produce ideas more often without having to worry about being criticized. Therefore, this rule can significantly help increase the diversity of the ideas. Rule 2 says all ideas need to be shared. This helps generate unusual ideas. Rule 2

Table 2-1: Steps in a Brainstorming Process

Step	*Action*
1	A group of people with as diverse background as possible sit together.
2	Generate ideas according to Osborn Rules.
3	Problem owners are chosen to pick up the better ideas to solve the problem.
4	Use the ideas picked up by the owners in step 3 as a main source of generating more ideas.
5	Randomly pick up ideas to generate more ideas similar to Step 3.
6	Let the owners choose several better ideas based on the newly generated ideas.
7	If no good enough solution is obtained, repeat steps 3-6.

Table 2-2: Osborn's original rules for brainstorming

Rule	*Statement*
1	Suspend Judgment
2	Anything Goes
3	Cross-fertilize
4	Go for Quantity

supplements Rule 1 in generating diverse idea and avoid being trapped in a box. Rules 3 says to generate ideas based on already generated ideas and combine those already generated ideas. This rule is achieved by having problem owners in Step 3 of Table 2-1. Finally, Rule 4 states the more ideas the better as quality will eventually come from quantity.

2.2 BSO The Algorithm

Since the first time Brain Storm Optimization algorithm was introduced in 2011 [31], there has been different versions of BSO proposed in the literature that each provides a different way to enhance BSO [39]. First, the original form of BSO will be discussed since it is new to the field of electromagnetic applications and antenna engineering. Then, in a later section, a suggested version that can be utilized for electromagnetic applications will be discussed. To facilitate the discussion and provide a clear picture, a list of common terminology that are used when discussing BSO is established. Those terms are listed and defined in Table 2-3. Some of those terms might be called differently in literature, and some might not be used in different versions of BSO. Figure 2-1 provides a pictorial illustration of the solution generation process that follows the steps of Table 2-1. The figure should provide a smooth transtion from the conept behind BSO to BSO the

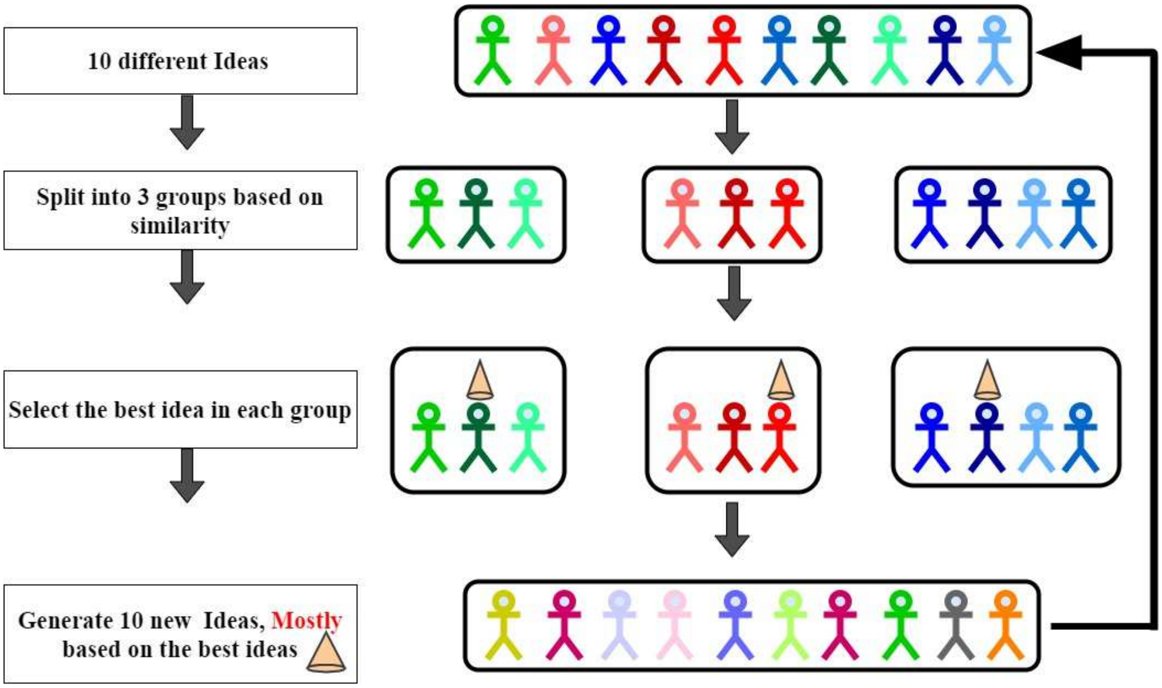

10 different Ideas
Split into 3 groups based on similarity
Select the best idea in each group
Generate 10 new Ideas, Mostly based on the best ideas

Figure 2-1: Basic idea generation concept in BSO

algorithm.

2.2.1 The Original Form of BSO

There are three major operations in BSO: Grouping, Replacing, and Selecting. After defining the problem to be optimized, and developing a fitness function, one is ready to apply BSO algorithm using the following steps.

A. Population Initialization:

This step is similar to most population-based algorithm. A population of N individuals/ideas is denoted as:

$$X = x_i = [x_i^1, x_i^2, ..., x_i^D] | 1 \le i \le N$$

First, define the boundary of the solution space in each dimension. Then, an individual in the population is represented by a vector such that the i^{th} individual is initialized randomly throughout the *D*-dimensional search space as:

$$x_i^d = x_{\min}^d + \beta(x_{\max}^d - x_{\min}^d), \qquad 1 \le d \le D \tag{2.1}$$

$(x_{\max}^d, x_{\min}^d)$: are the specified minimum and maximum value for each d^{th} dimension in a D-dimensional optimization. β is a random number uniformly distributed between 0 and 1.

B. Clustering Individuals:

Basically, we want to divide the *N* individuals into M groups or clusters based on their location in the search space such that individuals that are closer to each other belong to the same cluster as shown in Figure 2-2. These clusters represent the problem's local optima. There are various clustering algorithms that can be utilized in the brain storm optimization algorithm. In this work, the basic k-means clustering algorithm is utilized [40, 41].

C. Rank Individuals:

After evaluating individuals using a pre-defined fitness function, individuals are ranked in each cluster. The best individual in each cluster is selected to be the cluster center, such that for every cluster we have a cluster center.

D. Disrupting Cluster Centers:

This is a simple operation in BSO. To avoid deception, and increase population diversity, we occasionally replace a cluster center with a randomly generated idea. This operation is controlled by $P_{replace}$. A random number between 0 and 1 is generated. If the value is less than a probability $P_{replace}$, randomly generate an idea using Eq.(2.1) to replace the randomly selected cluster center.

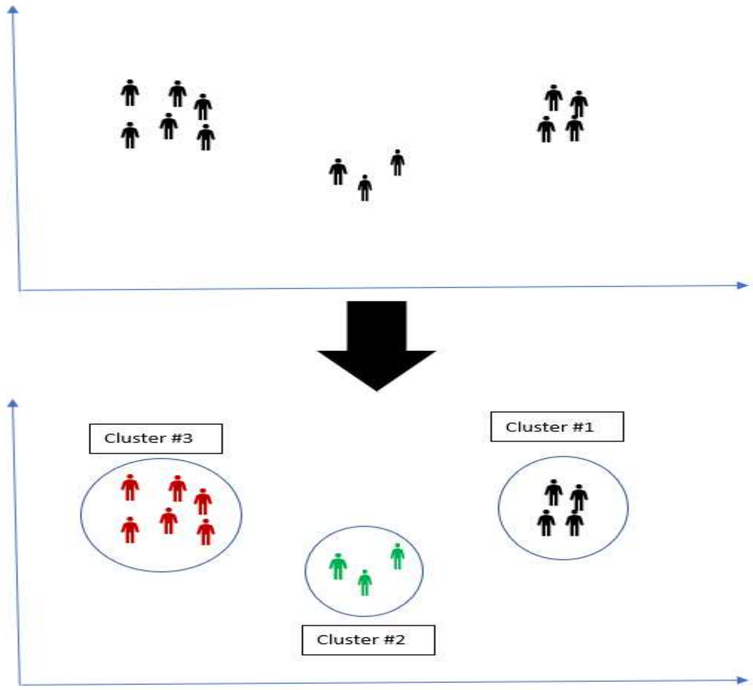


Figure 2-2: Simple grouping of individuals in 2-D space into 3 groups

This step helps individuals escape from local optima. $P_{replace}$ is recommended to be small to yield better results. The mean solutions obtained by BSO algorithm with different values $P_{replace}$ of are within the same order of magnitude in most of the common benchmark functions [42].

E. Updating Ideas:

The most important step in BSO, or any optimization algorithm is how new solutions are generated. In BSO, it is divided into two parts: the ideas are generated based on either one cluster or two clusters. This operation is controlled by $P_{generation}$. A random number between 0 and 1 is generated; if the generated number is less than $P_{generation}$, the new idea is created based on one cluster. Otherwise, an idea is generated based on two clusters:

$$\begin{cases} oneCluster & rand() < P_{generation} \\ twoCluster & otherwise \end{cases} \tag{2.2}$$

Generating an idea from one cluster improves the exploitation ability in the algorithm by having the solution converges within a confined solution region. On the other hand, generating an individual from two clusters improves the exploration ability and avoid deception. It has been shown that a relatively smaller value of $P_{generation}$ might improve the results as it increases the exploration ability [42].

I. One-Cluster idea generation:

A cluster is selected according to the number of ideas it contains. The more ideas a cluster contains the higher the chance that cluster is selected. Then, an idea is generated either on the cluster center of random individual in the same cluster. The choice of selecting the cluster center or random individual in the cluster is controlled by $P_{oneCluster}$. A random number between 0 and 1 is generated. If the number is smaller than a pre-determined probability $P_{oneCluster}$, choose the cluster center. Otherwise, randomly select an individual in the same cluster. In either case, the idea generation as follows:

$$x_{new}^{d} = x_{selected}^{d} + \xi * \mathrm{N}(\mu, \sigma) \tag{2.3}$$

Where d is the dimension index, $\mathrm{N}(\mu, \sigma)$ is a Gaussian random noise with mean μ and variance σ, and ξ is a weighting coefficient that is calculated as follows:

$$\xi(t) = \mathrm{logsig}\left(\frac{\frac{T}{2} - t}{k} \right) * rand(), \tag{2.4}$$

where logsig() is a logarithmic sigmoid transfer function. T is the maximum number of iteration, t is the number of current iteration, and k is for changing logsig() function's slope. The weighting coefficient will have larger values during the initial iterations compared when the algorithm reaches its final iterations. This will strengthen the exploration property during the beginning of the search. As the search goes on, it will be more in favor of exploitation property.

II. Two-Cluster idea generation:

This is similar to the previous step. However, the two clusters are chosen randomly, whereas in the one-cluster idea generation, we chose the cluster based on a probability that it is proportional to the number of ideas it contains. Then, either select their centers or random individuals in the same clusters. Choosing between cluster centers and random ideas is controlled by $P_{twoCluster}$ in the same manner of the one-cluster idea generation strategy is controlled by $P_{oneCluster}$. Then, the two selected individuals are combined by a weighting sum:

$$x_{selected}^{d} = R * x_{selected1}^{d} + \left[1 - R\right] * x_{selected2}^{d} \tag{2.5}$$

where R is a random number between 0 and 1. Then, the new idea is created using Eq.(2.3). The BSO algorithm can also be extended to use more than two clusters in generating new ideas to make a more complex algorithm [31]. A flow chart of the discussed steps in the BSO algorithm is shown in Figure 2-3.

Table 2-3: BSO Dictionary

Term	*Definition*
Individual/Idea	Represents a point in the search space that is evaluated by the fitness function.
Population	Represents the total ideas used in the optimization.
x_i^d	Represents the location of i th individual in the d th dimension.
Fitness Function	Represent the quality the generated individual in the current population.
Cluster	Represents a group of individuals/ ideas that are positioned closely together.
Cluster Center	Represents the best idea/individual in a cluster.
$P_{replace}$	Pre-determined probability to determine whether to replace the cluster center with a randomly generated individual/idea.
$P_{generation}$	Pre-determined probability to determine whether to generate individuals/ideas based on one or two clusters.
$P_{oneCluster}$	Pre-determined probability that is used, when one cluster is used, to determine whether the cluster center or another individual/idea in the same cluster.
$P_{twoCluster}$	Pre-determined probability that is used, when two clusters are used, to determine whether the cluster center or another individual/idea in the same cluster.
ξ	A weghting coeffecinet to control the speed of convergence.

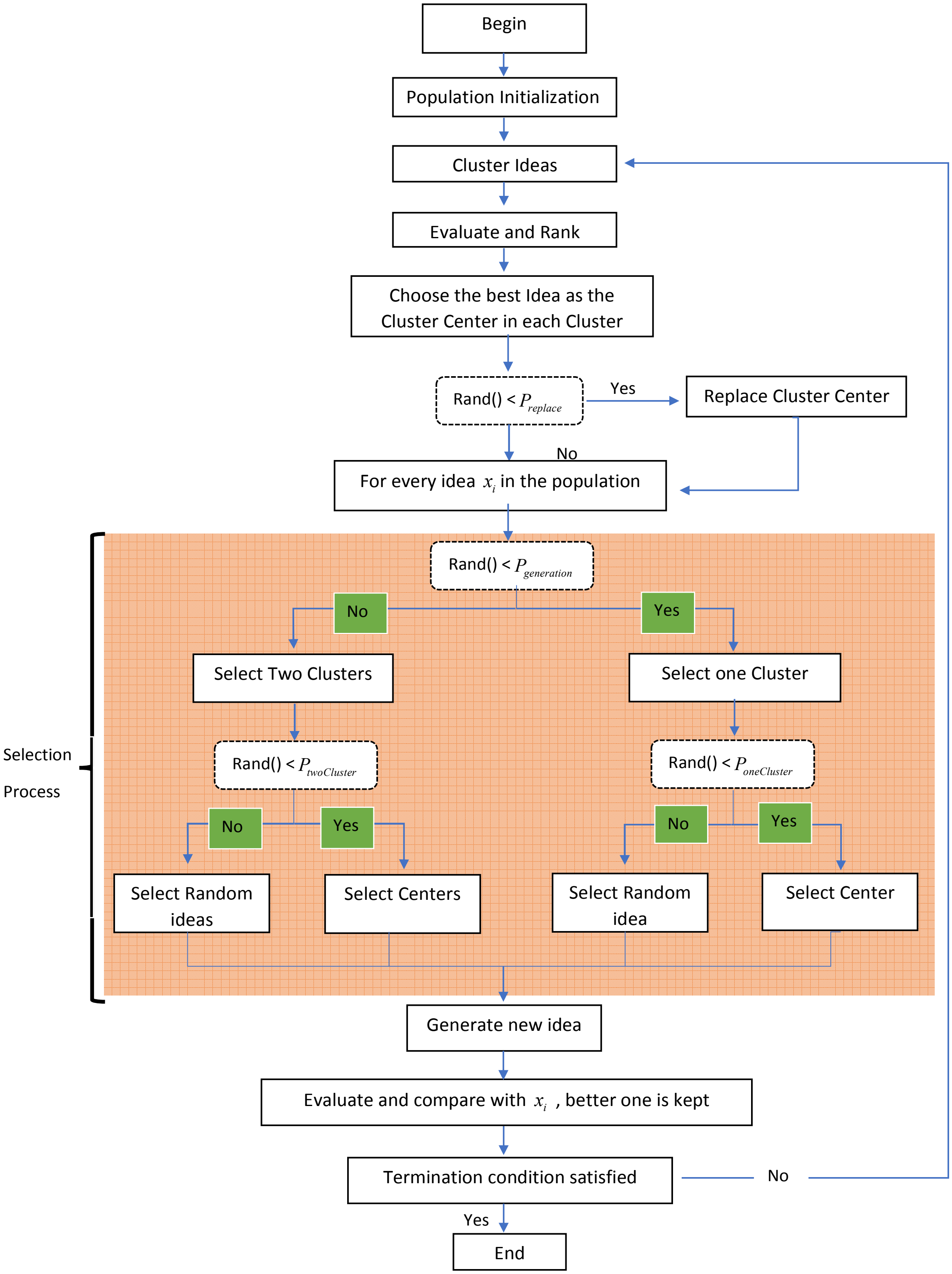

Begin
Population Initialization
Cluster Ideas
Evaluate and Rank
Choose the best Idea as the Cluster Center in each Cluster
Rand() < $P_{replace}$
Yes
Replace Cluster Center
No
For every idea x_i in the population
Rand() < $P_{generation}$
No
Yes
Select Two Clusters
Select one Cluster
Rand() < $P_{twoCluster}$
Rand() < $P_{oneCluster}$
No
Yes
No
Yes
Select Random ideas
Select Centers
Select Random idea
Select Center
Selection Process
Generate new idea
Evaluate and compare with x_i , better one is kept
Termination condition satisfied
No
Yes
End

Figure 2-3: Flowchart of BSO Algorithm

2.2.2 Modifying BSO For Electromagnetic Problems

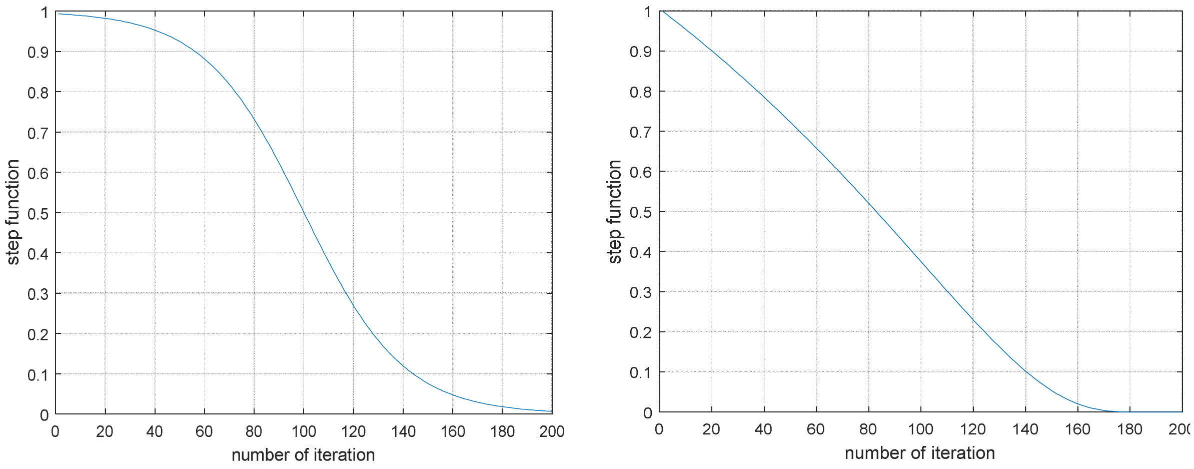
There have been many works suggesting changes to BSO for better performance. Some of the attempts to improve BSO are:

- Brain storm optimization algorithm for multi-objective optimization problems [43].
- A modified brain storm optimization [38]. The authors introduce a new grouping and solution generation strategies.
- Brain storm optimization algorithm with modified step-size and individual generation [44].
- Maintaining population diversity in brain storm optimization algorithm [45]. The authors introduce two kinds of partial re-initialization strategies.
- Brainstorm optimization with Chaotic Operation [46].
- Brainstorm optimization algorithm with re-initialized ideas and adaptive step size [47].
- An improved brain storm optimization with differential evolution strategy [48].
- Random grouping brain storm optimization algorithm with a new dynamically changing step size [49].

Modifying the step function Eq.(2.4) has been proven to improve the performance of BSO [44، 48، 47]. The suggested step function is

$$\xi(t) = rand() * \exp\left(1 - \frac{T}{(T-t)+1}\right) \tag{2.6}$$

where $rand()$ is a random number between 0 and 1. T is the maximum number of iteration, t is the number of current iteration. The modification is aimed at improving the balance between exploration and exploitation at different searching generations. The new step function provides a smooth transition between the exploration and exploitation stages. Figure 2-4 shows the difference



(a) Eq.(2.4), original BSO with k=20. (b) Eq.(2.6), the new suggested step function

Figure 2-4: Step function comparison without multiplying by a random number.

between the two step functions without multiplying by a random number to illustrate the behavior of the two step functions.

Another modification to BSO is also to the step function; there is a big flaw in the solution generation mechanism in the original BSO that is shown in Eq. (2.3). The disadvantage of the step function in Eq. (2.6) returns a value within (0,1) and random(0,1) is also a random value within (0,1), their product ξ is still within (0,1). Then, the step function ξ is multiplied by a Gaussian random value with mean of 0 and variance of 1 in Eq. (2.6). This random noise is with very high probability within the range (-3,3), which may be not efficient enough for global search when the search space is very large as pointed out in [47, 38]. This is illustrated in Figure 2-5. Also, this applies to when the search is too small; solutions in the first iterations will be outside the search space. Hence, this approach is very inefficient. To solve this problem, there needs to be a way to take the search space into consideration during the solution generation strategy. In [47], it is

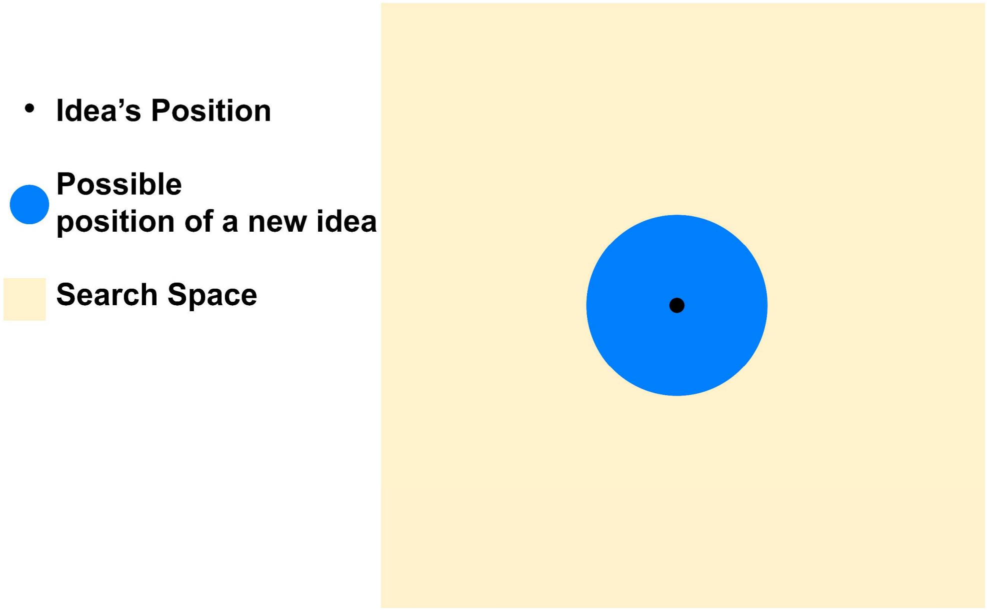


Figure 2-5: Original BSO solution generation in a large search space

proposed to multiply the step function ξ by a new parameter, α that is proportional to the search space size. This will transform Eq. (2.6) to

$$\xi(t) = rand() \cdot \exp\left(1 - \frac{T}{(T-t)+1}\right) \cdot \alpha \tag{2.7}$$

In the is work, the α will be set to

$$\alpha = \frac{1}{4}(x_{\max} - x_{\min}) \tag{2.8}$$

where $x_{\max}$ and $x_{\min}$ denotes the upper bound and the lower bound in the search space, respectively. This will ensure that all the solution space is covered during the initial iterations where exploration is encouraged. A smaller value of α can lead to a faster convergence rate. However, this also means the distance among ideas will be very close to each other, which might

Table 2-4: BSO Parameters Used in Thesis

Parameter	*Value*
$P_{replace}$	0.2
$P_{generation}$	0.8
$P_{oneCluster}$	0.4
$P_{twoCluster}$	0.5
Clusters	5
μ	0
σ	1

lead the algorithm to get trapped in a local optimum due to lack of diversity. Unless otherwise stated, values of intrinsic BSO parameters that will be used throughout the thesis is listed in Table 2-4.

CHAPTER 3

3 Comparison Between BSO and PSO

Due to the stochastic nature of global optimization algorithms, it is difficult to provide an analytical proof that one optimization algorithm's convergence is superior to another optimization algorithm's convergence. Part of the difficulty resides in the lack of knowledge on the behavior of the fitness function in most applications. Although the behavior of BSO in a continuous space has been analyzed in [50], the analyzed behavior is based on the number of generation goes to infinity when the interest of this work is the convergence in as few iterations as possible. Nonetheless, population-based optimization algorithms have gained popularity in solving a lot of optimization problems. Different types population-based algorithms have emerged in the recent years such as nature-inspired optimization algorithms. Swarm intelligence algorithms is one form of nature-inspired optimization algorithms in which each individual in the population represent a point in the search space. One of the most famous optimization algorithms in the EM community is particle swarm optimization (PSO) algorithm [51]. One of the key features of PSO is its implementation simplicity compared to any other algorithm. PSO has been introduced thoroughly to the EM community by Robinson and Rahmat-Samii in 2004 [4]. Furthermore, PSO has been applied to many applications in EM problems in the literature [22, 24, 52, 21]. Hence, PSO will be used as gauge to measure the performance of BSO and comparing different aspects such as convergence speed.

In this chapter, firstly, an analogy between PSO and BSO will be discussed to further explain the behavior of BSO, and facilitate the transition from PSO to BSO, and vice versa. This will illustrate how each algorithm uses its inherent characteristics to optimize a problem. Next, to compare the ability of algorithms, one section will be dedicated to the use sets of mathematical benchmark functions to provide a general insight of both algorithms. In later chapters, both algorithms will be applied to electromagnetic applications, the real purpose of this work.

3.1 Particle Swarm Optimization

The use of particle swarm optimization (PSO) is very common in electromagnetic applications; however, a short introduction of the PSO algorithm will be given before making a comparison between BSO and PSO. PSO mimics the social behavior of swarm of bees searching for flowers. Each bee relies on its own past experience as well as the experience of other bees in the swarm. The bee tries to combine its knowledge with the knowledge of the swarm to find the location that is abundant with flowers. Thus, two forces attract a bee in the swarm; the force of the best location found personally, and the force of the best location found by the entire swarm. The major advantage of PSO the algorithm is the simplicity of its implementation. However, some key terms used to describe PSO is explained in Table 3-1.

The beauty of PSO is the whole algorithm can be explained by two equations. The two equations represent the solution generation strategy in PSO i.e. how a solution is generated for each particle in the optimization. The two equations can be written as

$$v^{t+1} = w * v^t + c_1 R * (P^t_{best} - x^t) + c_2 R * (G^t_{best} - x^t) \tag{3.1}$$

$$x^{t+1} = x^t + v^{t+1} \tag{3.2}$$

Where v^{t+1} is the velocity of the particle at iteration $t+1$, x^{t+1} is the particle's position or coordinate in the search space, P_{best} is the best solution found by the particle, G_{best} is the best solution found by the entire swarm, c_1 and c_2 are weighting factors to determine the amount

Table 3-1: PSO Dictionary

Term	*Definition*
Particle/Agent	One single individual in the population (swarm). It represents a point in the search space that is evaluated by the fitness function.
Swarm	The entire population of particles
Pbest	The personal best of a particle. The location of the best fitness returned for a certain particle
Gbest	The global best of the swarm. The location of the best fitness returned for the entire swarm

influence by the personal experience of the particle and the experience of the entire swarm, R is random number between 0 and 1, and w is the inertial weight that controls the amount of adjustment that a particle makes when looking for new solutions. After random initialization of the swarm in the beginning, each particle is updated using Eq. (3.1) and (3.2) for the desired amount of iterations or good enough solution is found.

3.2 Algorithm Comparison

The purpose of this section is not to show which is better, but to draw a map that helps translate from PSO to BSO. They both share the origin of population-based algorithm, and their populations are initialized in the same manner. However, they are both built on different foundation, or they use different species. The essence of the BSO algorithm is grouping or clustering the population. The clustering operation in BSO makes it more complex than PSO in which one or two equations are needed to control most of the algorithm. This difference makes

BSO more expensive computationally. There have been efforts made to remedy this problem by introducing a new grouping operator [38]. The positive side is that most electromagnetic problems are computationally expensive, which makes the effect of the clustering operation completely negligible. Hence, any inherent computational time of the algorithm can be neglected when applied for electromagnetic applications.

When investigating any optimization algorithm, there are two important features that define the algorithm. The two features are exploration and exploitation. In other words, how an algorithm diverges to avoid deception and escapes local minimum traps, and how it converges towards an optimal solution as well as how an algorithm maintains a healthy balance between the two features. First, when a particle in PSO moves more towards its personal best, then the algorithm is leaning to explore other solutions. In the case of BSO, exploring is achieved by different operations: adding noise when generating ideas, disrupting the cluster center; and two-cluster idea generation [45]. In the other hand, when a particle in PSO moves more towards the global best, then the algorithm is leaning to exploit what it could be the global minimum (or maximum). In the case of BSO, converging towards the best solution is achieved via clustering similar ideas together, which is strengthened by one-cluster idea generation.

Maintaining a good balance between the ability is important to obtain good results. PSO has been investigated by many researchers to obtain an optimal balance. In PSO, the convergence is controlled through adjusting the inertial weight w. In the case of BSO, the parameter to control the two abilities were discussed in detail in section 2.2 . However, there yet need to be further investigation in how much control these parameters have under different conditions. A summary of comparison between BSO and PSO is shown in Table 3-2.

Table 3-2: Differences between BSO and PSO

Aspect	*BSO*	*PSO*
Population Initialization	Random	Random
Updating Individuals	Gaussian Noise	Particle velocity
Controlling Convergence Speed	Weghting Coeffecinet ξ	Inertial Weight
Achieving Exploitation	- Clustering - One-Cluster Generation	Moving towards the global best
Achieving Exploration	- Disrupting the center - Two-Cluster Generation	Moving towards the personal best

This is a small comparison between the basic BSO and the basic PSO. PSO is much older than BSO, 16 years older, and has received the attention of many engineering communities. Thus, PSO has had a big window to improve. BSO remains young with the opportunity of further improvements.

3.3 Mathematical Benchmark Function Comparison

Traditionally, when introducing a new optimization algorithm, it is compared with different algorithms using different common benchmark functions. It is a tradition in the literature to validate and demonstrate optimization algorithms. Mathematical benchmark functions are a good way to test and compare different optimization algorithms because they have a fast computation time, which is not the case in electromagnetic applications. This allows us to preform thousands of tests on those function in a matter of few seconds. Hence, researchers use benchmark test functions as a technique to test and compare global optimization algorithms. With the increasing number of global optimization algorithms throughout the years, the number of those test functions to validate those algorithms are also increasing [53].

The standard procedure for testing global optimization algorithms is to run those algorithms many times on each benchmark function, and average those runs to see the performance of the algorithm. Most of those benchmark functions are to be minimized. The method of testing many runs might lead to undesired results when an algorithm is not guaranteed to converge to a global optimum. This happens when the values between the premature runs and the optimized runs are different by several orders of magnitude. A one premature run out of total of 50 runs can cause the average of those runs to be equal to it, which does not give an efficient insight of the algorithm performance. This behavior can be illustrated with the Schwefel function that is defined in D-dimensional space as

$$f_{sch}(x) = 418.9828872724339 \cdot D - \sum_{i=1}^{D}[x_i \sin(\sqrt{|x_i|})] \tag{3.3}$$

Figure 3-2 shows the Schwefel function in two-dimensional space. The function is usually evaluated in the search space $x_i \in [-500, 500]$, for all $i = 1, \mathrm{K}, D$. The function has a global minimum at $x_i = 420.96874636, \forall i \in 1, \mathrm{K}, D.$

Using BSO with parameter of 4 individuals and 2 clusters, the two-dimensional Schwefel function is optimized 50 times for maximum number of iterations of 2000. Figure 3-1a shows all the 50 runs of the Schwefel function optimization. Out of the 50 runs, 14 runs have premature convergence, which indicates that BSO fell into a local minimum trap. Figure 3-1b demonstrates the mentioned problem of the premature runs dominating the converged runs despite the fact they

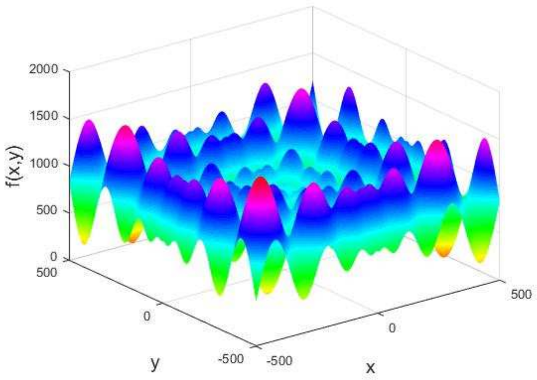
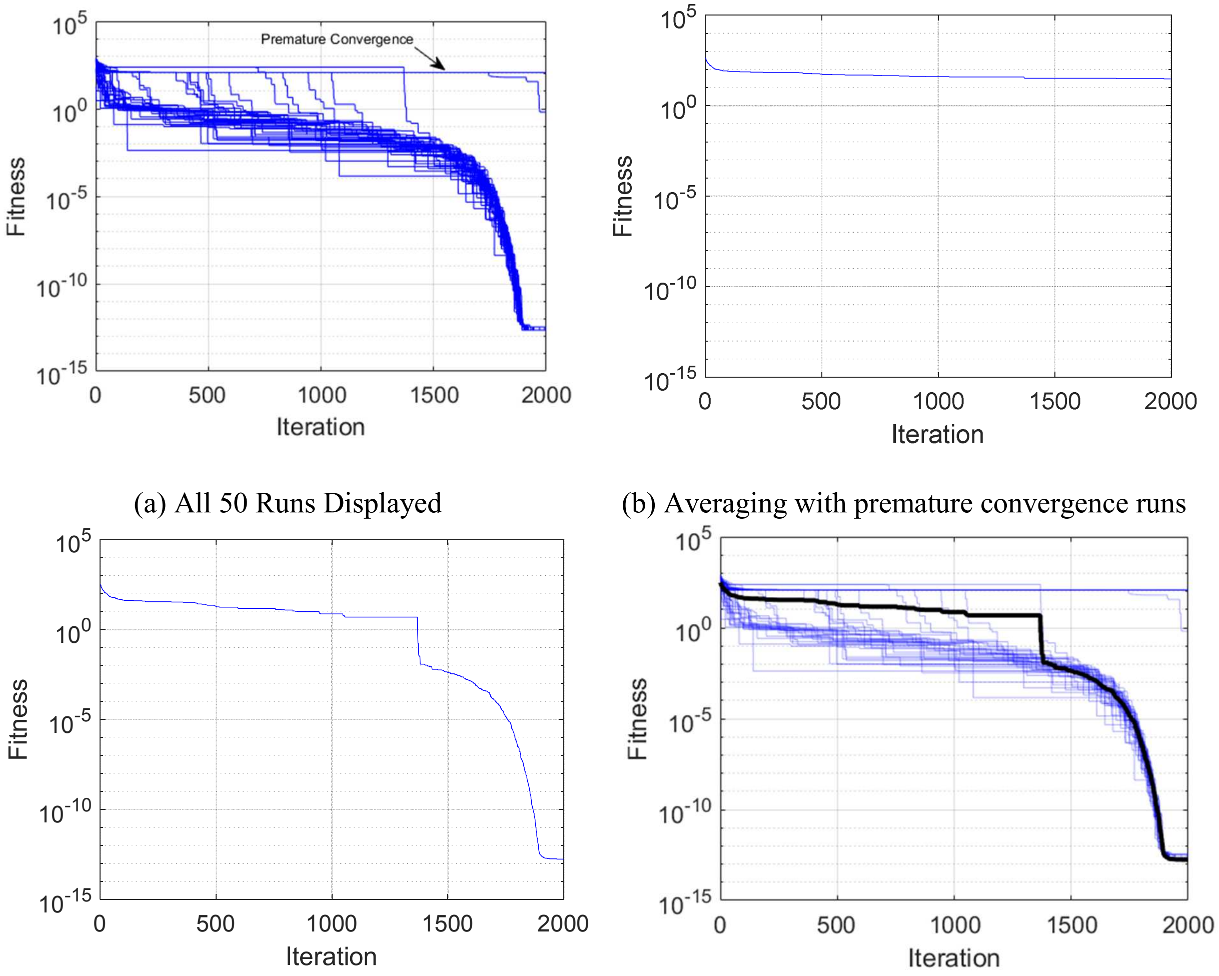


Figure 3-2: Two-Dimensional Schwefel function



(a) All 50 Runs Displayed

(b) Averaging with premature convergence runs

(c) Averaging without the premature runs

(d) All of the 50 runs with the averaging in (c)

Figure 3-1: Comparison of different averaging procedures

are fewer than the converged runs. To remedy this problem, the average fitness of the runs is plotted with omitting the permute runs as shown in Figure 3-1c. Therefore, any outliers with two order of magnitude higher fitness function than the best-found solution are not included when calculating the average fitness. The effect of doing that can be seen in Figure 3-1d where the average fitness provides a better insight of the optimization algorithm combined with the knowledge of the number of premature runs. This method will be used when BSO and PSO are applied to set of mathematical benchmark functions.

The selected test functions in this section are labeled with different properties such as unimodal, multimodal, well-conditioned, ill-conditioned, separable, non-separable, … etc. Table 3-3 shows the selected test functions that are used in this section. The sphere function is the only function that is unimodal while the rest of the functions listed in Table 3-3 are multimodal. It

Table 3-3: Benchmark Testing Functions

Name	*Function*	*Search Space*
Sphere	$f_{sph}(x)=\sum_{i=1}^{D} x_i^2$	$[-100,100]^D$
Rosenbrock	$f_{ros}(x)=\sum_{i=2}^{D}[(1-x_{i-1})^2+100(x_i-x_{i-1}^2)^2]$	$[-10,10]^D$
Rastrigin	$f_{ras}(x)=\sum_{i=1}^{D}[x_i^2-10\cos(2\pi x_i)+10]$	$[-5,5]^D$
Griewank	$f_{gr}(x)=\frac{1}{4000}\sum_{i=1}^{D}[x_i^2]-\prod_{i=1}^{D}[\cos(\frac{x_i}{\sqrt{n}})]+1$	$[-600,600]^D$
Schwefel	$f_{sch}(x)=418.9828872724339\cdot D-\sum_{i=1}^{D}[x_i \sin(\sqrt{\lvert x_i \rvert})]$	$[-500,500]^D$

should be noted that Rosenbrock function is a multimodal function except when it is two dimensional where the Rosenbrock function is unimodal [54]. Figure 3-3 shows each function visualized in a two-dimensional space. The Rosenbrock function has a global minimum at $x_i = 1, \forall i \in 1,\ldots,D$, The Schwefel function has a global minimum at $x_i = 420.96874636$, $\forall i \in 1,\ldots,D$ while the rest of the listed functions have global minima at $x_i = 0, \forall i \in 1,\ldots,D$. All the test functions have the same number of dimensions $D = 50$, the same number of population $N = 100$, and both algorithms will run for 2000 iterations. This will result into 20000 function evaluations. Usually, researchers go for much higher number of function evaluations, 10000*D. However, this luxury cannot be afforded into electromagnetic applications due to the amount of time required to evaluate the fitness function as mentioned before. The lower number of iterations will probably cause the algorithms to converge prematurely towards a local optimum. Nonetheless, the purpose is to compare the performance of BSO with the well-established PSO algorithm. Also, both BSO and PSO will have invisible boundary condition. The parameters used for BSO and PSO algorithms are listed in Table 2-4 and Table 3-4, respectively.

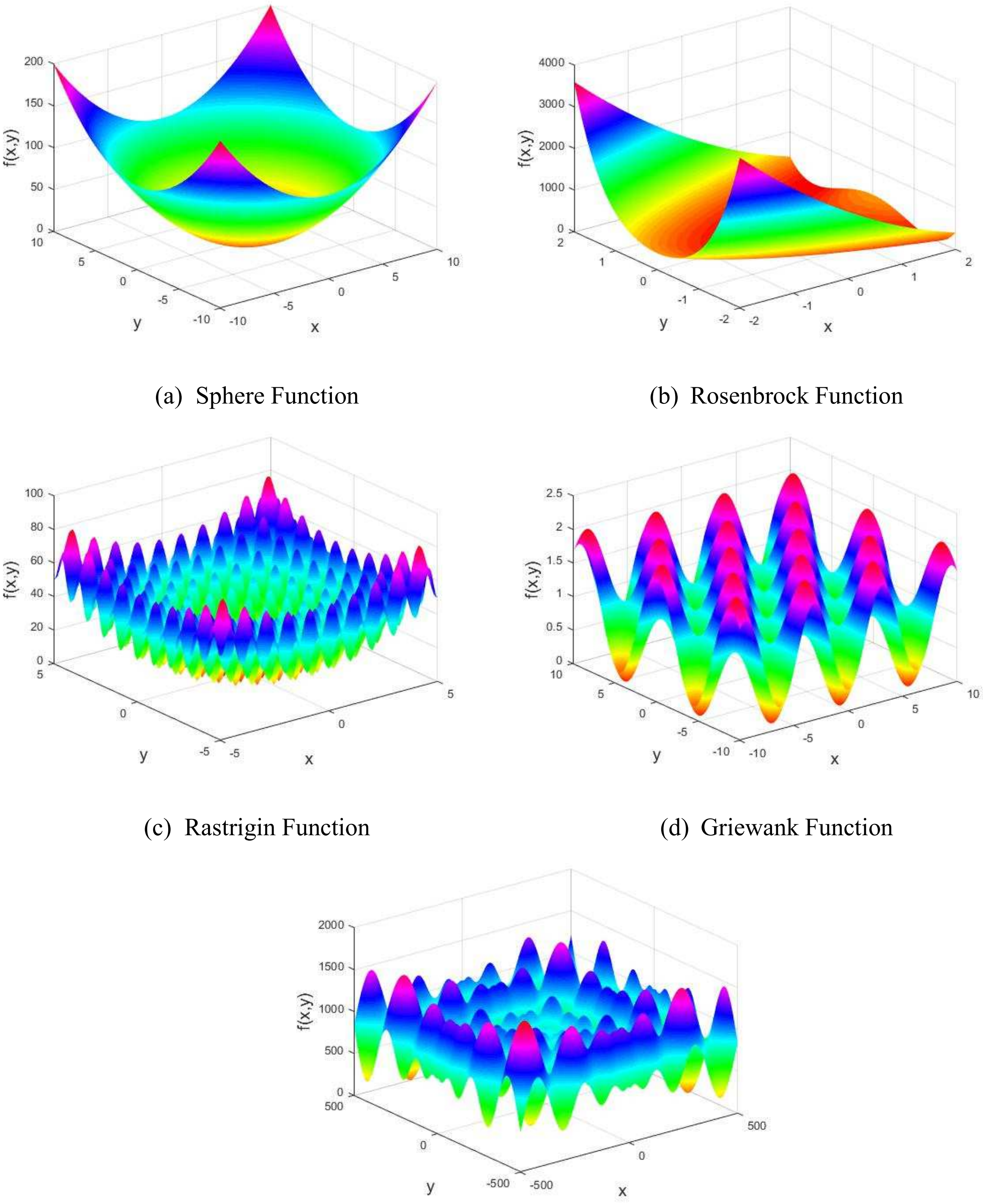
It should be noted that when analyzing at the results in Figure 3-4 that the y-axis is different

Table 3-4: PSO Parameters

Parameter	*Value*
c_1 , c_2	2.0
w	0.9-0.4
$V_{\max}$	$\frac{1}{2}(X_{\max} - X_{\min})$

in each case; the scaling is different in each plot. Generally, it can be seen that the BSO performs the same as PSO for the tested functions. With the exception for the Griewank function, both algorithms achieve optimum values of the same order of magnitude in each tested benchmark functions. Moreover, it can be noticed that both BSO and PSO have identical convergence speed for all functions. The only function where both algorithms perform differently is the Griewank function. BSO found an optimum point with value around 10^{-3} while PSO found an optimum point with value around 10^{-5}; however, 28 runs converged prematurely for PSO optimization.

In general, the results show that BSO can perform at least as good as PSO; both have the same convergence speed. The goal was to test the algorithm using fewer number of function evaluations than the numbers used in the literature. This difference in the optimization environment explains why the overall results of the algorithm are not comparable to the ones published in the field of optimization. In the next sections BSO will be tested using real-world electromagnetic applications where time and resources are limited, and no prior knowledge of the behavior of the fitness function.



(a) Sphere Function

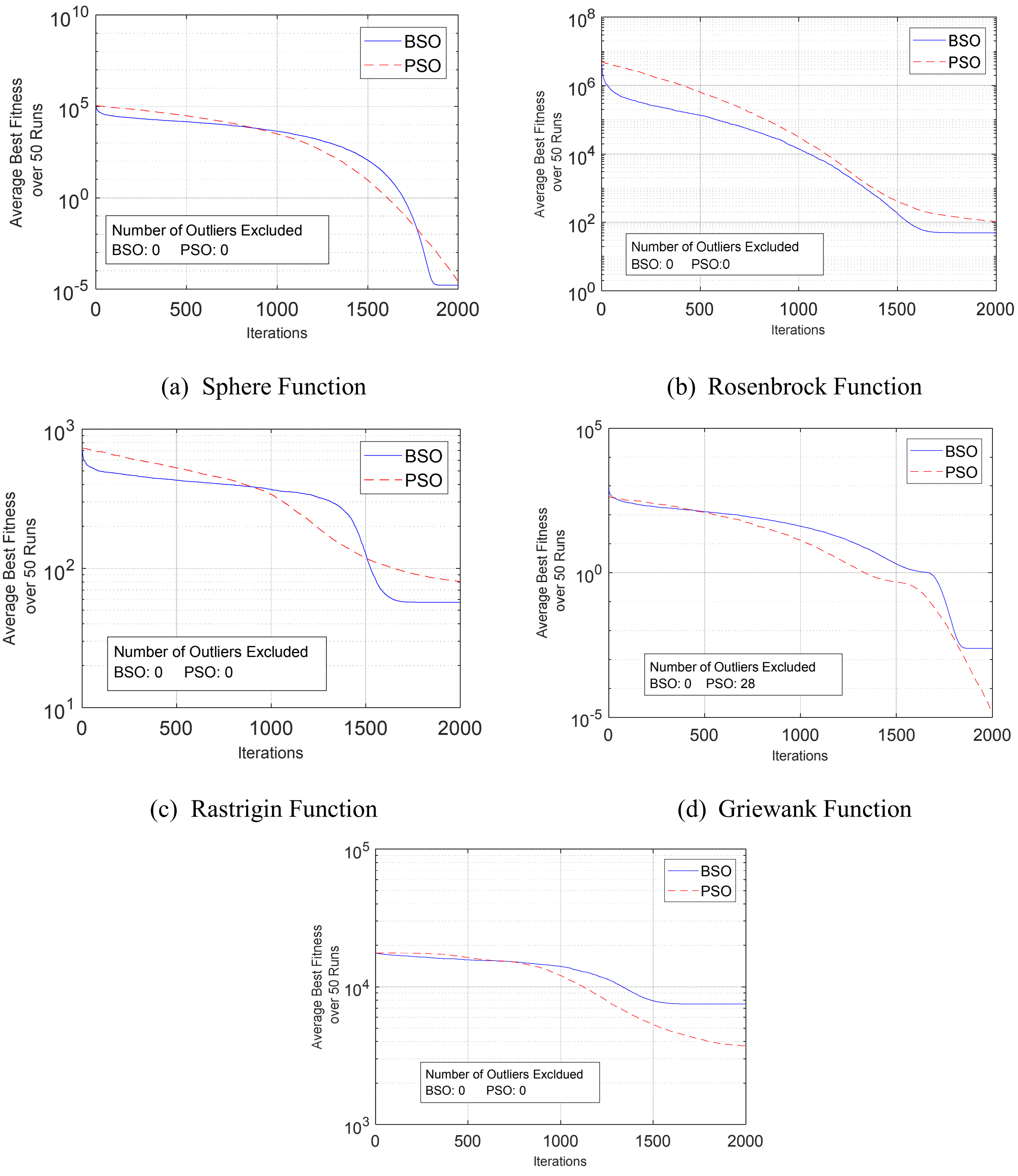
(b) Rosenbrock Function

(c) Rastrigin Function

(d) Griewank Function

(e) Schwefel Function

Figure 3-3: Two-dimension visualization of the testing functions



(a) Sphere Function

(b) Rosenbrock Function

(c) Rastrigin Function

(d) Griewank Function

(e) Schwefel Function

Figure 3-4: Optimization results for the benchmark functions

CHAPTER 4

4 Six-Element Yagi-Uda Optimization

4.1 Theory and Design of Yagi-Uda Antenna

Yagi-Uda antenna is a directional antenna that is practical for a wide range of frequencies (3-3000 MHz). The antenna is famous for its use as a home tv antenna. The antenna was first introduced in Japanese by Shintaro Uda of Tohoku Imperial University, Japan [55]. Then, reintroduced and explained by Hidetsugu Yagi [56], which is considered the more famous paper [57, 58]. It consists of a number of linear elements in parallel only one of which is excited. A folded dipole is usually used as the feed element; the feed element is also called a driven element. The rest of the elements are parasitic radiators that transmit energy by the currents that are induced by the mutual coupling of the driven element with these elements. The purpose of the parasitic elements is to make the antenna have an end-fire radiation pattern. Therefore, the elements that are

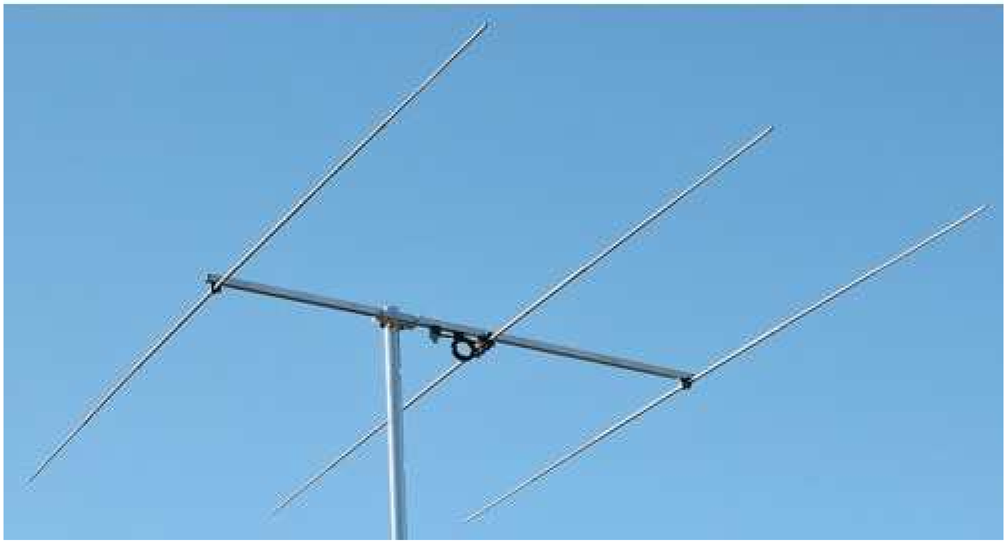


Figure 4-1: Three-element Yagi antenna that operates at 50 MHz. The length of longest element is 3.1 m [80].

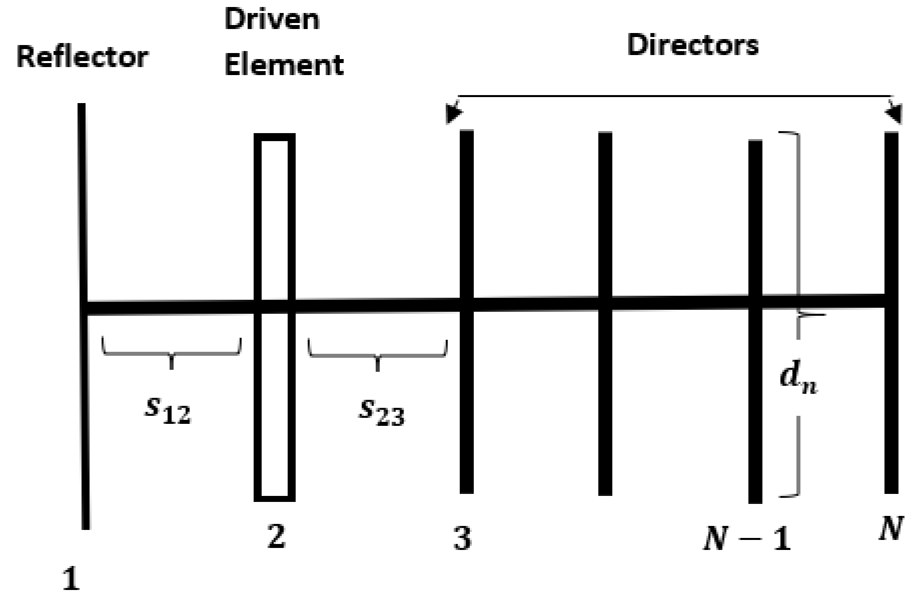


Figure 4-2: Yagi-Uda N-element configuration

at the side of beam from the driven element are called directors while the ones at the opposite side act as reflectors. A typical configuration of Yagi-Uda antenna can be seen in Figure 4-2.

The general design of the Yagi-Uda antenna is that the length of the driven element is slightly less than $\lambda / 2$ to achieve resonance in the presence of parasitic elements. Also, the directors' lengths are smaller than the intended resonant length, in the range of $0.4 - 0.45\lambda$. Hence, the directors will have capacitive impedance i.e. the current leads the voltage. The distance between the directors is in the range of $0.3 - 0.4\lambda$. Moreover, the lengths and the spacing of the direction are not necessarily uniform. Additionally, the reflector's length is slightly greater than $\lambda / 2$, which makes the impedance of the reflector inductive. Hence, the current on the reflector lags the voltage induced. With proper elements spacing, the directors form an array with equal progressive phase shift. Therefore, the Yagi-Uda antenna is designed as a travelling wave antenna [58].

The mathematical analysis of the Yagi-Uda Array is based on solving the Pocklington's Integral Equation for the total field generated by an electric current source radiating in free space using Method of Moment as presented in [58]

$$\int_{-l/2}^{l/2} I(z')\left(\frac{\partial^2}{\partial z'^2}+k^2\right)\frac{e^{-jkR}}{R}dz' = j4\pi\omega\varepsilon_0 E_z^t \tag{4.1}$$

$$R=\sqrt{(x-x')^2+(y-y')^2+(z-z')^2} \tag{4.2}$$

Using integration by part, and assuming the current at the end of each wire vanish, this leads to

$$-\frac{dI(z')}{dz'}\frac{e^{-jkR}}{R}\Bigg|_{-l/2}^{l/2}+\int_{-l/2}^{l/2}\left(\frac{d^2I(z')}{dz'^2}+k^2I(z')\right)\frac{e^{-jkR}}{R}dz' = j4\pi\omega\varepsilon_0 E_z^t \tag{4.3}$$

The essence of MoM is discretizing the geometry of interest into smaller segments. For small dimeter wires, which have been discretized into N small segments, the current on the nth element can be written as

$$I_n(z')=\sum_{m=1}^{M} I_{nm}\cos\left[(2m-1)\frac{\pi z'}{l_n}\right] \tag{4.4}$$

After substituting the above equation and considering the interaction of the discretized elements on the same wire and with elements on the other white, a system of linear equations is created. The total field will take the shape [58]

$$E_\theta=\frac{j\omega\mu e^{-jkr}}{4\pi r}\sin\theta\sum\nolimits_{n=1}^{N}\left\{e^{jk(x_n\sin\theta\cos\phi+y_n\sin\theta\sin\phi}\times\sum\nolimits_{m=1}^{M} I_{nm}\left[\frac{\sin(Z^+)}{Z^+}+\frac{\sin(Z^-)}{Z^-}\right]\right\}\frac{l_n}{2} \tag{4.5}$$

$$Z^+=\left[\frac{(2m-1)\pi}{l_n}+k\cos\theta\right]\frac{l_n}{2} \tag{4.6}$$

$$Z^{-} = \left[\frac{(2m-1)\pi}{l_n} - k\cos\theta \right] \frac{l_n}{2} \tag{4.7}$$

4.2 Setting Up the Optimization Problem

In this section, The Yagi-Uda antenna will serve as a mean of comparison between BSO and PSO. The radiation pattern of a 6-element Yagi-Uda antenna is to be optimized, namely the forward directivity and the front-to-back ratio. Those characteristics are controlled by the lengths of the all elements, as well as their spacing. The Yagi-Uda antenna has been optimized using various optimization techniques [14, 23, 59]. In this optimization, there are two goals to be achieved within the performance of the antenna; maximizing directivity and maximizing the front-to-back ratio. Generally, front-to-back ratio is compromised for forward directivity i.e. receives lower weight in optimization than forward directivity. However, for the purpose of the comparison in this section, the front-to-back ratio will receive high focus in the optimization to see how far each algorithm can go while obtaining higher directivity. Hence, the fitness function to optimize the two parameters is:

$$f_{Yagi} = (13.5 - D)^2 - F/B, \tag{4.8}$$

where D stands for the forward directivity in dB, and F/B stands for the front-to-back ratio. This function will be minimized by both BSO and PSO. The variables that are used for the optimization are: all the six elements' lengths d_n, and their relative spacing from each other s, as shown in Figure 4-2. d_1 refers to the length of the first element, which is the reflector. d_2 refers to the length of the second element, which is the driven element (folded dipole in this case). And, s_{12} refers to the spacing between the first and the second element; the same notation is used for the rest of the elements. Therefore, the optimization will be searching an eleven-dimensional search space (6 variables for lengths, and 5 for spacings). Next, the solution space is limited to

$$\begin{aligned}
&d_i \in (0.42\lambda, 0.52\lambda), \quad i = 1,2 \\
&d_j \in (0.4\lambda, 0.495\lambda), \quad j = 3,4,5,6 \\
&s_{12} \in (0.15\lambda, 0.45\lambda) \\
&s_{kl} \in (0.15\lambda, 0.45\lambda), \quad kl = 23,34,45,56.
\end{aligned} \tag{4.9}$$

As for the parameters used both algorithms, BSO and PSO will have the same number of population of $N = 25$, and will run for same number of iterations of $Max\,Iteration = 200$. Other intrinsic parameters used for BSO and PSO algorithms are listed in Table 2-4 and Table 3-4, respectively. The Yagi-Uda antenna is simulated using MATLAB's Antenna Toolbox™ [60].

4.3 Optimization Results

The optimized design variables of both algorithms are shown in Table 4-1. Due to space limitation the variables are broken into two tables Table 4-1a shows the elements' lengths and Table 4-1b shows the elements' spacings. The convergence of both algorithms is shown in Figure 4-3. Both BSO and PSO had the same convergence speed until the last few iterations where BSO was able to achieve relatively better fitness function. Both designs have almost the same forward directivity of 11.92 dB for BSO and 11.789 dB for PSO. On the other hand, BSO was able to achieve higher front-to-back ratio of 62.46 dB about 3 dB higher than what PSO was able to achieve which is 58.98 dB. Figure 4-4 shows the configurations of both designs relative to each other; the reflector of both design overlaps, so it is hard to distinguish both designs' reflectors. It can be seen that the lengths of both designs are somewhat similar while there are some differences in the spacing between elements. Figure 4-5 and Figure 4-6 shows the radiation pattern in the $xz-$plane and $yz-$plane, respectively.

BSO has demonstrated its potential to be a solid tool for optimization based on its performance with respect to the well-known algorithm PSO. BSO has a similar convergence speed to PSO and was able to achieve better results in the Yagi-Uda example. In the next chapter, BSO will applied to an additional example, and compared with PSO to provide more insight into the performance of BSO.

Table 4-1: Final Design Parameters of the Yagi-Uda Antenna by BSO and PSO. (a) Elements Lengths and (b) Elements Spacings

	d_1 / λ	d_2 / λ	d_3 / λ	d_4 / λ	d_5 / λ	d_6 / λ
BSO	0.48688	0.49017	0.41129	0.42589	0.40806	0.41431
PSO	0.48957	0.48814	0.43899	0.43486	0.42028	0.43238

(a)

	s_{12} / λ	s_{23} / λ	s_{34} / λ	s_{45} / λ	s_{56} / λ
BSO	0.18591	0.18815	0.23285	0.28333	0.33737
PSO	0.16346	0.17453	0.19566	0.33139	0.24800

(b)

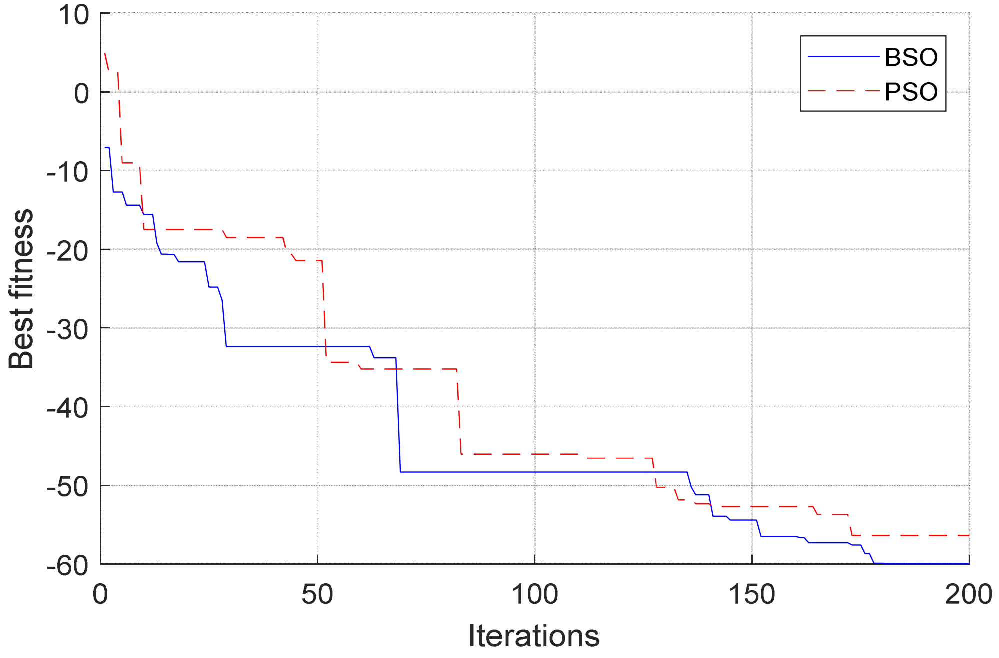


Figure 4-3: Comparison of BSO and PSO convergence

Table 4-2: Pattern Results of Six-Element Yagi-Uda Antenna for Both Designs

	Forward Directivity (dB)	*Front-to-Ratio (dB)*
BSO	11.922	62.46
PSO	11.789	58.98

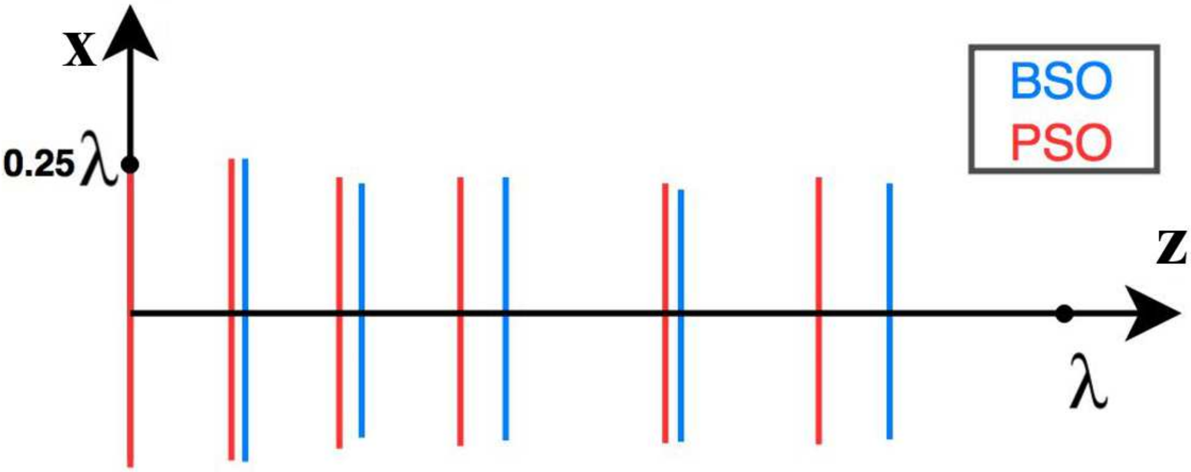
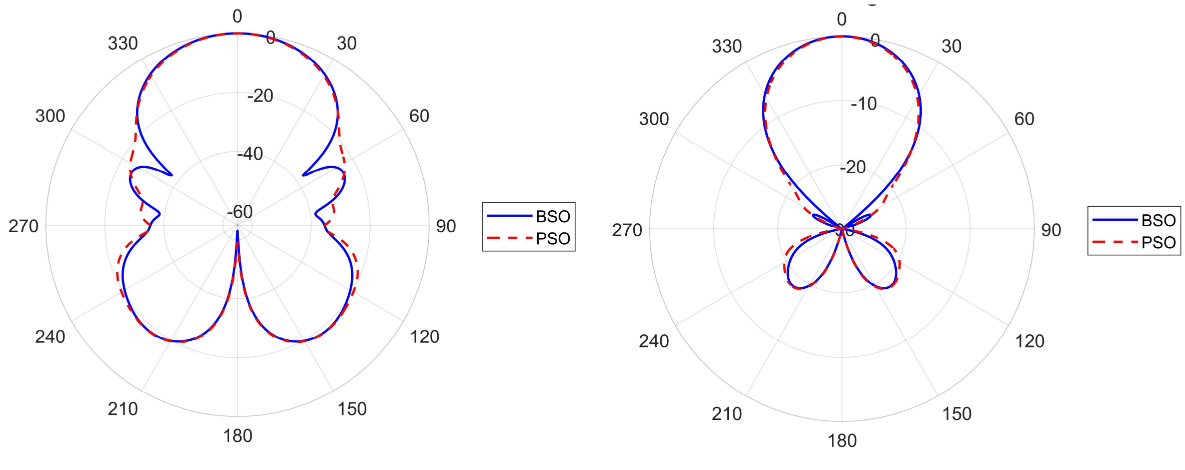
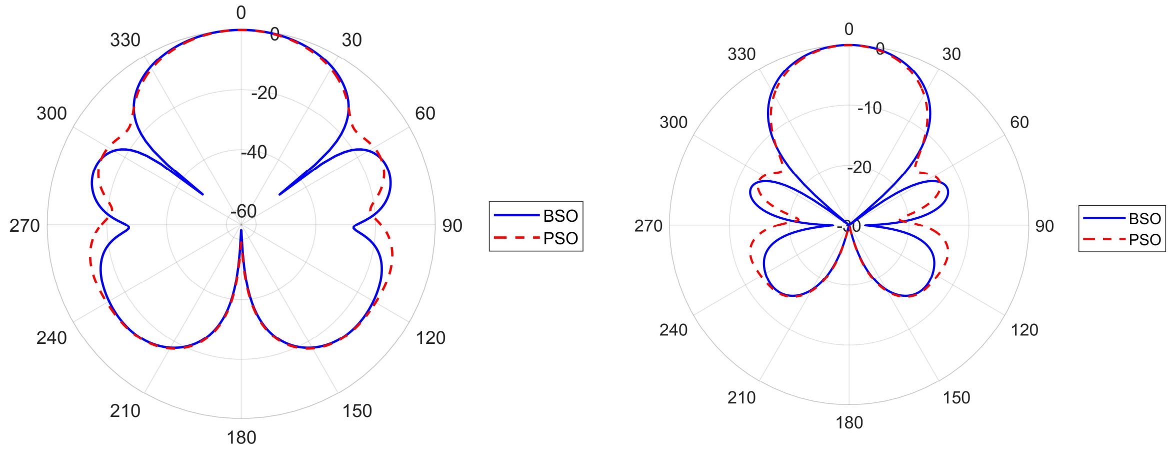


Figure 4-4: Final design configuration of both BSO (blue) PSO (red).



(a) (b)

Figure 4-5: The radiation pattern in the xz-plane. (a) The whole pattern is shown, (b) only including values above -30 dB.



(a) (b)

Figure 4-6: The radiation pattern in the yz-plane. (a) The whole pattern is shown, (b) only including values above -30 dB.

CHAPTER 5

5 Nonuniform Luneburg Antenna

In 1944, R. K. Luneburg proposed a special type of spherical lenses known as Luneburg Lens that has special focusing properties [61]. Luneburg lenses have received the interest of antenna designers; they have been investigated and applied to a variety of applications such as multi-beam scanning [62, 63, 64, 65, 66, 67, 68, 16]. This chapter is inspired by the work in [16], where a nonuniform Luneburg Lens was optimized using Genetic Algorithms (GA) to obtain a desirable radiation characteristic with less number of spherical shells.

An overview of the concept and theory of Luneburg Lens will be discussed as well as applying the mode-matching technique to solve for the electric field in the presence of multilayer dielectric sphere. Then, BSO will be applied to optimize a nonuniform spherical lens antenna in different configurations. Using the same procedure as Chapter 4, PSO will also be applied to the

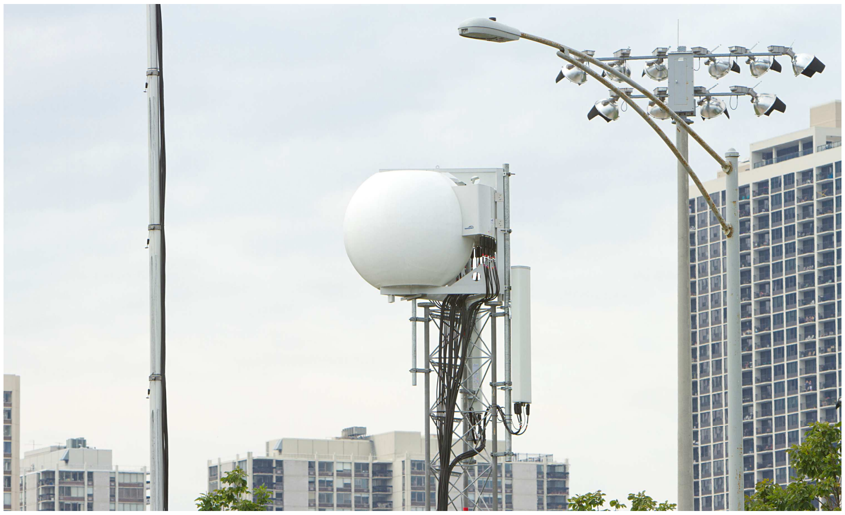


Figure 5-1: AT&T's Giant Eyeball Antenna (also known as the Luneburg Lens Antenna) [79].

same deigns in order to test the performance and the ability of BSO.

5.1 Theory and Application of Luneburg Lens

A Luneburg lens is a symmetrically dielectric sphere that has a decreasing refractive index radially out from its center. The dielectric constant ε_r of an ideal Luneburg lens starts from 2 at the center of the sphere then continuously decreasing to 1 at the outer surface. The behavior of permittivity of the sphere ε_r is expressed as:

$$\varepsilon_r(r) = 2 - \left(\frac{r}{a}\right)^2, \quad (0 \le r \le a), \tag{5.1}$$

where a is the radius of the sphere. This gradual variation of the refractive index allows Luneburg Lens to of transform the point-source radiation into the plane wave as seen in Figure 5-2 and vice versa. The spherical symmetry of Luneburg Lens allows to use a single lens with multiple feeds looking in different direction providing a wide-angle and multi-beam scanning systems. Luneburg lens can also be used as a radar reflector by metalizing part of the surface.

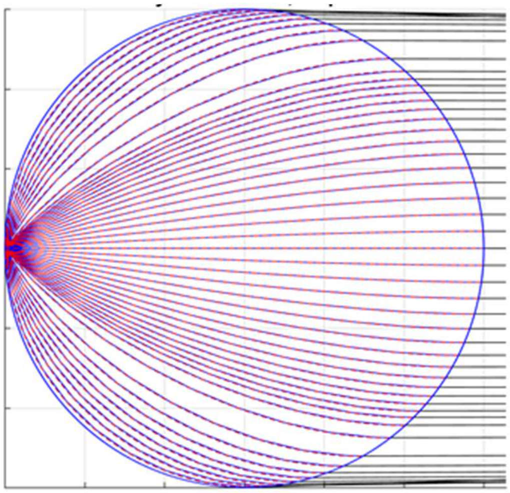


Figure 5-2: Illustration of a ray picture of Luneburg Lens.

The technology of constructing the radial continuous variation of the permittivity lenses does not exist yet. In practice, Luneburg lens is constructed using a discrete number of homogeneous spherical shells that has the same behavior of decreasing permittivity from the most inner shell to the most outer shell as seen in Figure 5-3. Moreover, the permittivity profile of an ideal Luneburg compared with a discrete (5-layer and 10-layer) uniform multi-shell spherical lens is shown in Figure 5-4.

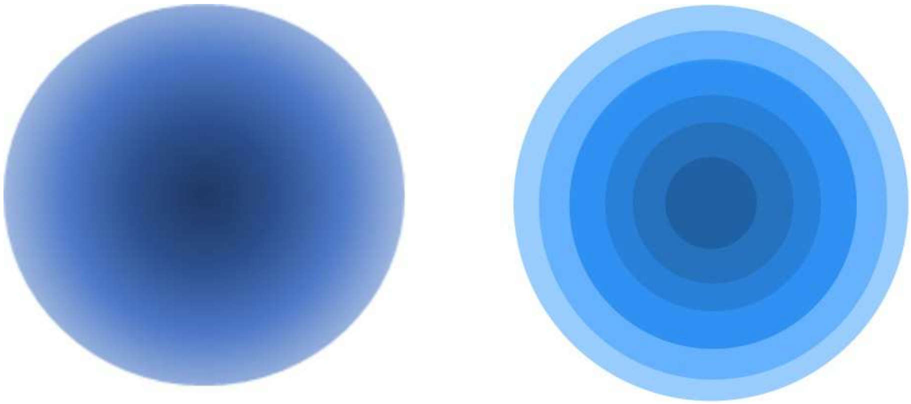


Figure 5-3: Luneburg Lens (Theoretical and its Practical Implementation)

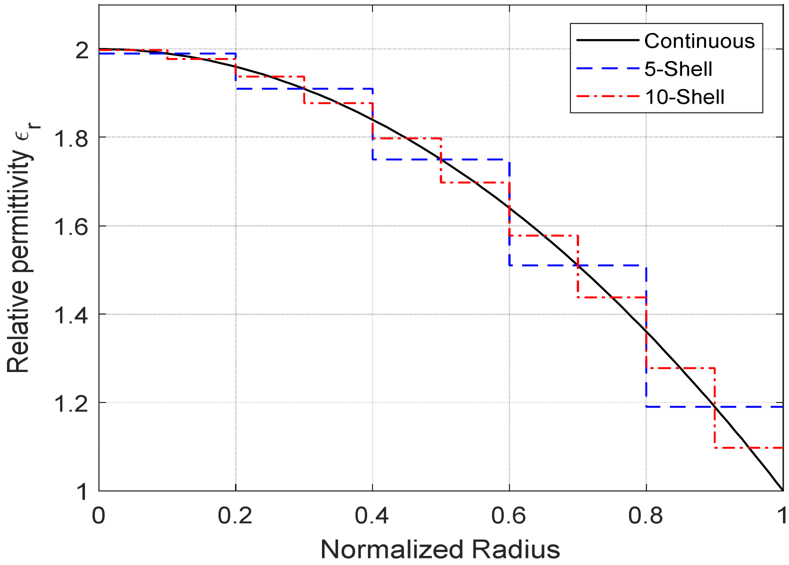


Figure 5-4: Permittivity distribution of ideal Luneburg Lens (continuous "straight line") vs practical approximation (discrete "dashed line")

5.2 Mode-Matching Solution of Luneburg Lens

The properties of Luneburg lens are a result of using the principle of ray-tracing techniques and Fermat's principle or what is known as Geometrical optics (GO) [61]. However, GO is not the optimum method of analyzing when the dimensions of the lens are comparable to the wavelength. Therefore, when designing and analyzing those lenses at microwave frequencies, one needs to perform more accurate electromagnetic analysis. There are multiple studies on analyzing the properties of Luneburg lens using exact electromagnetic solution of inhomogeneous dielectric lenses [69, 70]. Furthermore, the Luneburg Lens model in this chapter is constructed using a discrete number of homogeneous spherical shell, which has been analyzed in [68, 64]. The analysis method in this chapter is the same method that is found in [68], which is based on the spherical vector wave function that uses mode matching technique to solve such a boundary value problem in which the solution of the field is obtained from the dyadic Green's function [71]. The total electric field in the presence of such Luneburg lens is written as

$$\vec{E}\left(\vec{r}\right)=\vec{E}^{inc}\left(\vec{r}\right)+\vec{E}^{scatt}\left(\vec{r}\right)=\int_V \overline{G}_e\left(\vec{r},\vec{r}'\right).\left[-j\omega\mu_s\vec{J}\left(\vec{r}'\right)\right]dv' \tag{5.2}$$

where $\vec{E}^{inc}$ is the incident field; $\vec{E}^{scatt}$ is the scattered field; $\overline{G}_e\left(\vec{r},\vec{r}'\right)$ is the total dyadic Green's function responding to an infinitesimal dipole; $\vec{J}$ is the current density of the source; μ_s permeability of region s containing the current source. The scattered field $\vec{E}^{scatt}$ in the ith region due to the arbitrary infinitesimal dipole with current moment $Il\,\hat{p}$ in the jth region can be expressed as [68, 16]

$$\vec{E}_i^{scatt}\left(\vec{r}\right) = -\left(\frac{\omega\mu_s Il}{4\pi}\cdot\beta_j\right)\sum_{n=1}^{\infty}\sum_{m=0}^{n}\left(2-\delta_{m0}\right)\frac{\left(2n+1\right)\left(n-m\right)!}{n\left(2n+1\right)\left(n+m\right)!}$$
$$\cdot\left[\begin{array}{l}\left(A_{ni}\vec{M}^{(4)}_{\substack{e\\o}mn}(\beta_i)+C_{ni}\vec{M}^{(1)}_{\substack{e\\o}mn}(\beta_i)\right)\left(\vec{M}^{(4)'}_{\substack{e\\o}mn}(\beta_j)\cdot\hat{\beta}\right)\\ +\left(B_{ni}\vec{N}^{(4)}_{\substack{e\\o}mn}(\beta_i)+D_{ni}\vec{N}^{(1)}_{\substack{e\\o}mn}(\beta_i)\right)\left(\vec{N}^{(4)'}_{\substack{e\\o}mn}(\beta_j)\cdot\hat{\beta}\right)\end{array}\right] \tag{5.3}$$

where $\vec{M}$ and $\vec{N}$ are the spherical vector wave function; β_i and β_j are the complex propagation constant of regions i and j, respectively. A_{ni}, C_{ni}, B_{ni}, and D_{ni} are unknown coefficients that are determined by applying boundary conditions at the inferences of the dielectric shells.

5.3 Optimizing Luneburg Lens

The Luneburg lens serves as an interesting example to test the capability of BSO. The structure of the optimization problem is the same exact found in [16] in which Genetic Algorithm is applied. When comparing a five-shell of 30λ - diameter Luneburg lens with a ten-shell lens that has the same diameter, the later design has a superior radiation performance. The thickness and the material permittivity of each shell of a five-shell 30λ - diameter Luneburg lens is shown in Table 5-1. In general, those type of lens antennas is fed by horn antennas due to their end-fire radiation pattern. In this work, the feed is modeled using four infinitesimal dipoles that produce an end-fire radiation pattern. The configuration of a five-shell lens with the feed is shown in Figure 5-5.

There are two drawbacks of the five-shell uniform Luneburg lens compared to the ten-shell lens: (1) the appearance of grating lobes near the main lobe, and (2) lower gain. The radiation patterns of those two lenses are shown in Figure 5-6 where the issue of grating lobes in the five-shell case is very noticeable. Moreover, the gain of the five-shell is about 36.61 dB which is lower than the ten-shell case which is about 37.88 dB. Additionally, different configurations in terms of

Table 5-1: Design Parameters of a Five-Shell Luneburg Lens with a Radis of 15λ

Shell	ε_r	t/λ
1	1.18	3
2	1.50	3
3	1.74	3
4	1.90	3
5	1.98	3

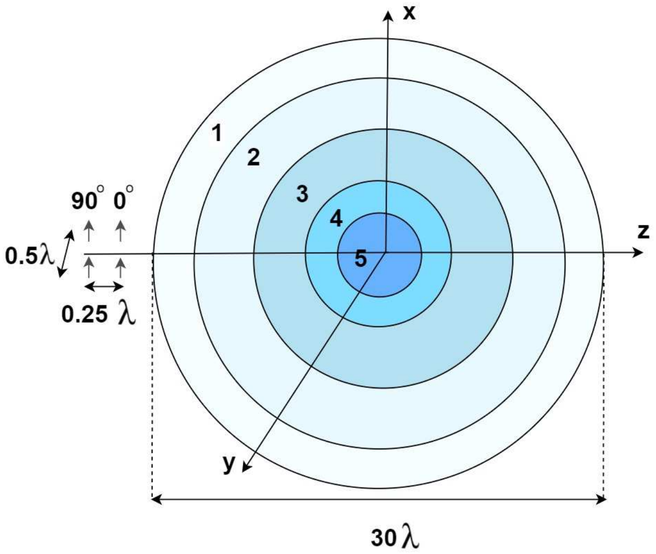


Figure 5-5: Five-shell 30λ diameter Luneburg lens antenna. An end-fire antenna consisting of four infinitesimal dipoles models the actual feed.

the source position and the existence of 0.05λ air gap will be analyzed. The patterns of those different configurations are also shown in Figure 5-6. Those two cases have even worse radiation performance. Besides, having more configurations provides even larger test cases for BSO to give a better overview of BSO performance. The next step is to define the variables that need to be optimized as well as their search space and the fitness function. Then, we define the optimization

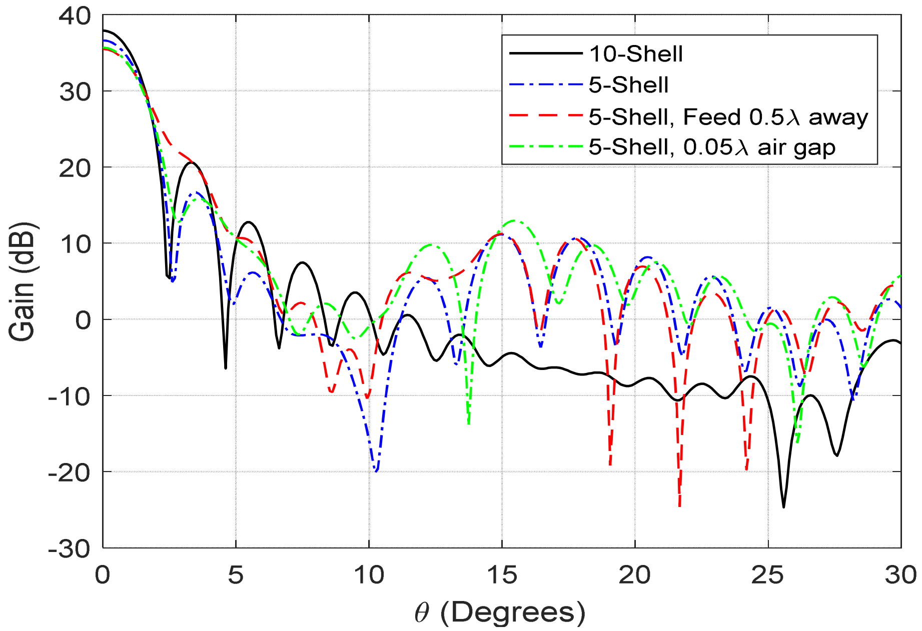


Figure 5-6: Gain pattern of the 30 λ diameter uniform Luneburg lens Antenna in the x-z plane.

parameters of BSO and PSO, and compare the results of the two algorithms.

5.3.1 Developing the Optimization Problem

As discussed, the purpose of the optimization is to design a 5-shell lens with improved radiation performance i.e. lowering the side-lobe level while maximizing the boresight gain. It is desirable to achieve a similar behavior of decreasing sidelobe levels that is found in the ten-shell uniform lens. Thus, an envelope function in the side-lobe region based on the ten-shell lens is developed as shown in Figure 5-7 and is written as

$$f_{env}\left(\theta\right)=12-38\log\left(\frac{\theta^{\text{o}}}{5.8^{\text{o}}}\right)\quad(dB),\quad 2.5^{\text{o}}\leq\theta\leq 25^{\text{o}}. \tag{5.4}$$

The fitness function in this optimization can be written as

$$F=(40-G_0)^2+(E)^2, \tag{5.5}$$

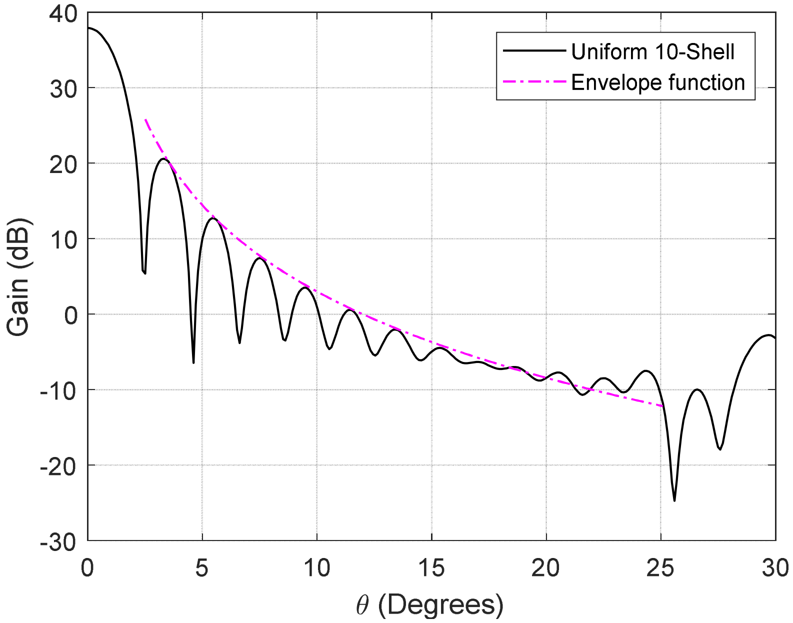


Figure 5-7: Gain pattern of the ten-shell 30λ diameter uniform Luneburg lens antenna with its sidelobe envelope function f_{env} in the x-z plane

where G_0 is the boresight gain, and E is the same parameter that is used in [19]; it represents the goodness of the side lobe level compared to the envelope function. It is calculated as the average difference between the envelope function in Eq.(5.4) and sidelobes that are above the envelope function. Hence, the equation to calculate E can be written as

$$E = \frac{1}{N}\sum_{i=1}^{N}\left[G(\theta_i) - f_{env}(\theta_i)\right] \tag{5.6}$$

where $G(\theta_i)$ is the gain pattern of the lens antenna. If one needs to ensure not losing gain even at the cost of worse grating lobes, the gain term in the fitness function can have a higher weight in the overall sum. It is a trade-off between the two traits and it depends on the desired design requirements. In this optimization, the thickness and the permittivity of each shell are the optimization variables. the permittivity ε_{ri} of each layer of the five-shell lens can take any value from 1 to 2. Furthermore, the thickness of the inner shells can have larger values than the outer shells to achieve better design [16]. Therefore, the search space of the optimization is defined by the set of the following equations.

$$\begin{aligned} &\varepsilon_{ri} \in [1,2], && i \in (1,2,3,4,5) \\ &t_j \in [0, 2.9\lambda], && j \in (1,2) \\ &t_3 \in [0, 5\lambda] && \\ &t_4 \in [0, 4\lambda] && \end{aligned} \tag{5.7}$$

$$t_1 + t_2 + t_3 + t_4 + t_5 = 15\lambda. \tag{5.8}$$

Eq.(5.8) is to keep total dimeter of the sphere fixed at 30λ . Also, the thickness of the most inner shell i.e. the fifth shell is obtained from the same equation.

The five-shell lens will be optimized using both BSO and PSO. The two algorithms will have the same number of population of $N = 25$ and run for 400 iterations. This will result in a total of 10000 function evaluations, which is the same number of evaluations used in [16]. Also, both BSO and PSO will have invisible boundary condition. Other intrinsic parameters that are used for BSO and PSO algorithms are listed in Table 2-4 and Table 3-4, respectively.

Furthermore, there are four different cases of the 5-shell lens, based on the source position and the existence of 0.05λ air gap, that will be optimized to add more room of performance evaluation of BSO. For simplicity, these cases will be referred to by their numbers that are assigned to them in Table 5-2.

5.3.2 Optimization Results

The optimized design variables of Case 1 for both BSO and PSO are shown in Table 5-3a and Table 5-3b, respectively. The two designs' variables are quite different, and this has led to two distinct design performances. In this case, BSO achieved better results than PSO as shown in

Table 5-2: Cases of Different Configuration of the Feed Position and Air Gap in the 5-Shell Lens

Case	*Feed distance from sphere (λ)*	*Air Gap (λ)*
1	0	0
2	0.5	0
3	0	0.05
4	0.5	0.05

Figure 5-8. BSO was able to obtain a gain of 36.11 dB, which is about 3 dB higher than what PSO design was able to achieve. Furthermore, BSO nearly suppressed the grating lobes to the envelope level making the design even better than PSO design where the grating lobes levels are distinguishingly higher than BSO's design sidelobes. This distinction in performance results can be further seen in Figure 5-9 that shows the convergence curve comparison of the fitness function between BSO and PSO. It can be seen that PSO struggled in achieving a good design. PSO seems to have stuck in local optimum trap in a very early stage of the optimization and failed to obtain any better solution after the 200th iteration. On the other hand, BSO was able to find better solutions during the exploration stage. This case demonstrates that PSO is prone to fail in a local optimum trap while BSO was able to maintain a good balance of exploration and exploitation which led BSO to have superior design results.

The optimized design variables of Case 2 for both BSO and PSO are shown in Table 5-4a Table 5-4b, respectively. Both designs achieved almost identical results both in terms of final design radiation performance as seen in Figure 5-10 and convergence speed that is shown in Figure 5-11. Both designs were able to minimize the grating lobe greatly compared to the uniform

Table 5-3: Optimized Variables of case (1) of the Nonuniform Lens Optimization Using (a) BSO and (b) PSO

Shell	ε_r	t / λ
1	1.4483	2.3134
2	1.2525	2.0959
3	1.7537	3.0595
4	1.8665	1.5888
5	1.974	5.9424

(a)

Shell	ε_r	t / λ
1	1.7937	0.7959
2	1.4917	2.0342
3	1.7738	0.7807
4	1.2523	2.4318
5	1.9679	8.9574

(b)

Luneburg lens.

The optimized design variables of Case 3 for both BSO and PSO are shown in Table 5-5a Table 5-5b, respectively. Figure 5-12 shows the gain pattern of both designs, and Figure 5-13 shows the convergence curve of both algorithms. PSO's design has 1.6 dB higher gain than BSO's; however, PSO' design has worse grating lobes. Based on the designated fitness function, BSO has a slightly better design as shown in the convergence curve.

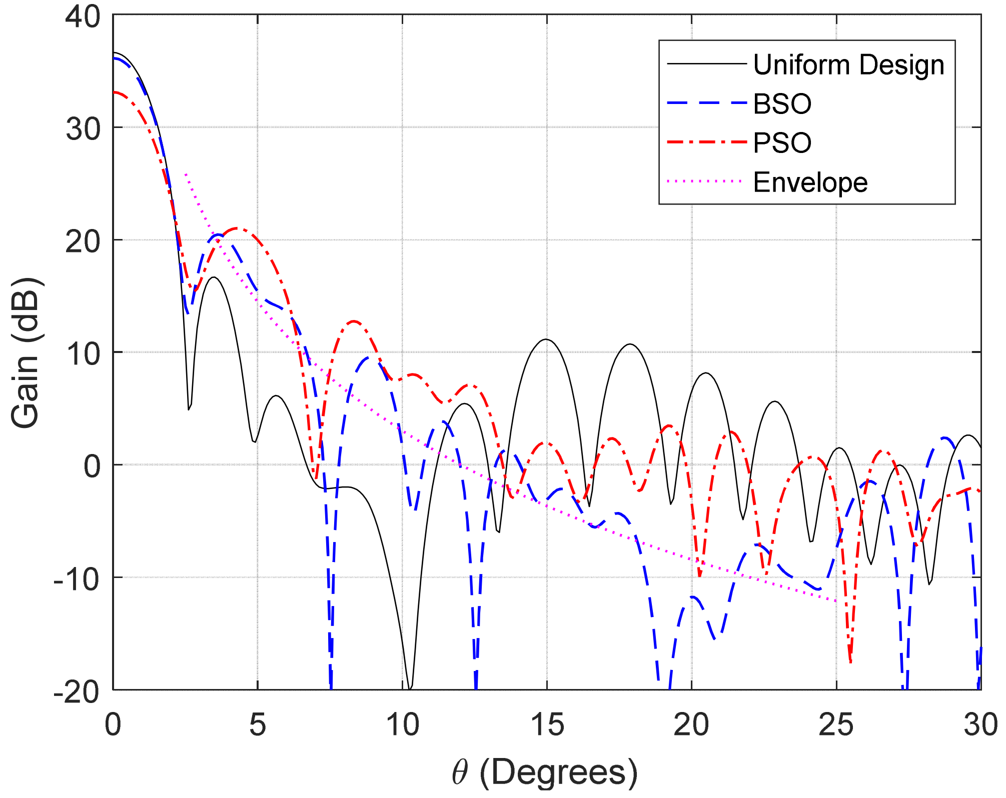


Figure 5-8: Gain pattern comparison between BSO and PSO of case (1) of the five-shell nonuniform optimized lens

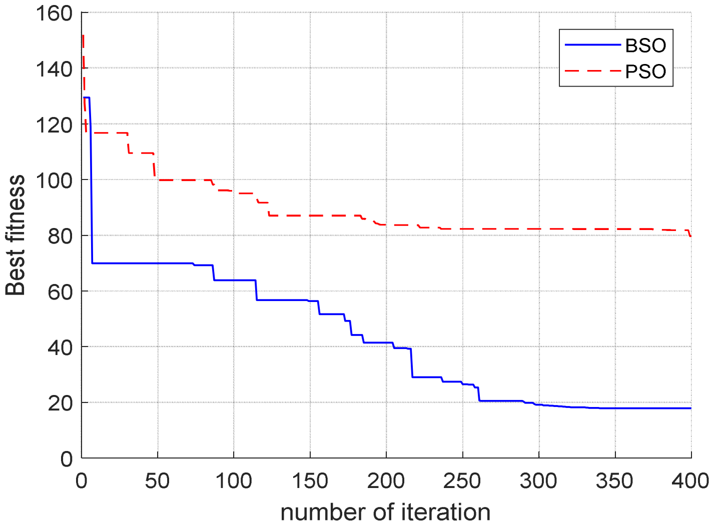


Figure 5-9: Convergence curve comparison between BSO and PSO of case (1) of the five-shell nonuniform optimized lens

Table 5-4: Optimized Variables of case (2) of the Nonuniform Lens Optimization Using (a) BSO (b) PSO

Shell	ε_r	t / λ
1	1.4854	2.0668
2	1.7563	0.1452
3	1.2459	2.5
4	1.867	3.7387
5	1.9607	6.5475

(a)

Shell	ε_r	t / λ
1	1.5717	2.0787
2	1.0723	0.3203
3	1.2672	2.5832
4	1.7373	3.1411
5	1.8839	6.8767

(b)

The optimized design variables of Case 4 for both BSO and PSO is shown in Table 5-6a Table 5-6b, respectively. Figure 5-14 shows the gain pattern of both designs, and Figure 5-15 shows the convergence curve of both algorithms. BSO’s design has a better gain, more than 1 dB higher than PSO’s while both designs have similar grating lobes level. Hence, in terms of overall performance, BSO has a relatively better design.

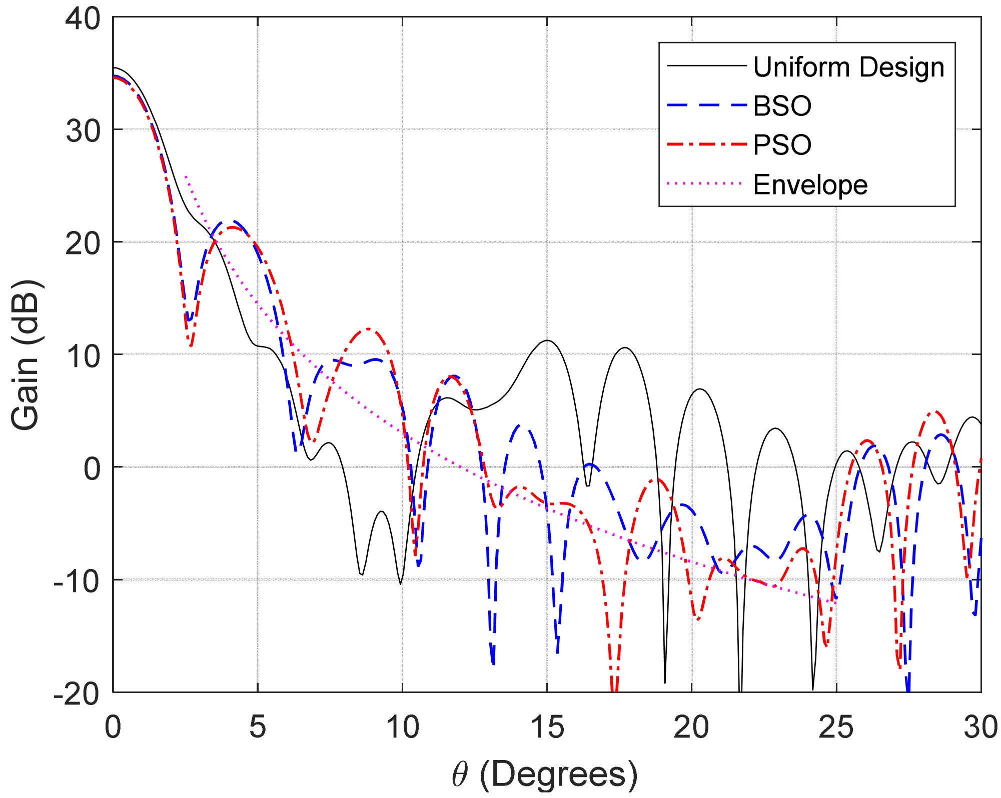


Figure 5-10: Gain pattern comparison between BSO and PSO of case (2) of the five-shell nonuniform optimized lens

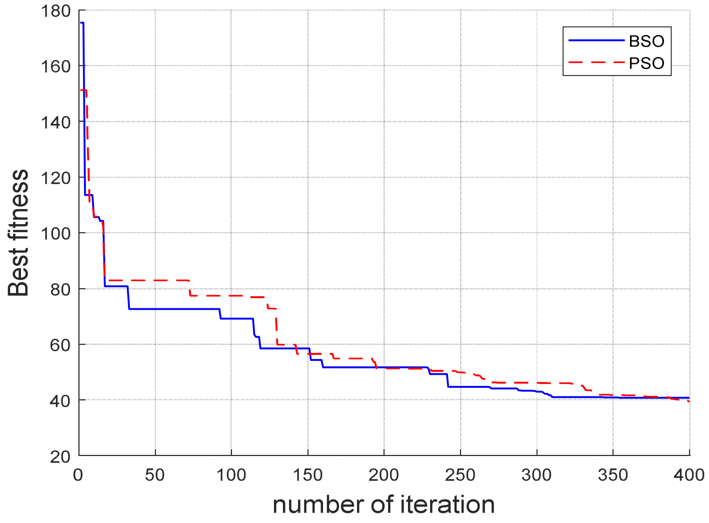


Figure 5-11: Convergence curve comparison between BSO and PSO of case (2) of the five-shell nonuniform optimized lens

Table 5-5: Optimized Variables of case (3) of the Nonuniform Lens Optimization Using (a) BSO (b) PSO

Shell	ε_r	t / λ
1	1.2316	1.1697
2	1.4021	0.6680
3	1.3407	3.3015
4	1.8153	2.4563
5	1.9293	7.2045

(a)

Shell	ε_r	t / λ
1	1.1825	2.5024
2	1.3776	1.1830
3	1.7038	0.6734
4	1.7945	3.3755
5	1.9745	7.0657

(b)

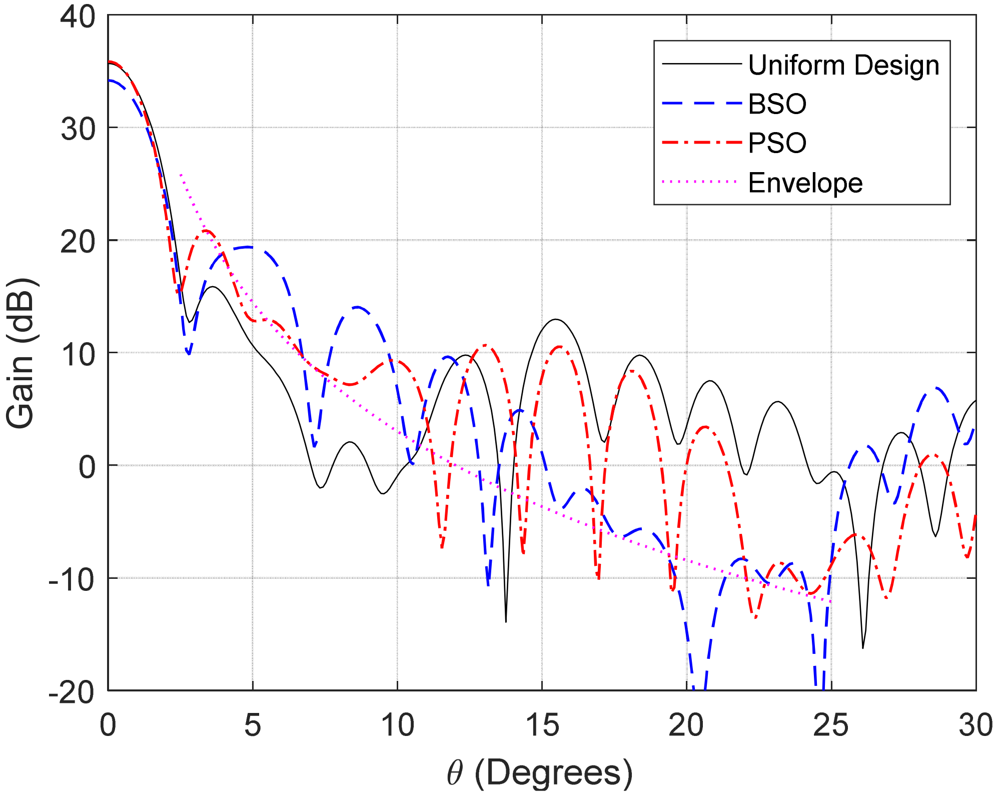


Figure 5-12: Gain pattern comparison between BSO and PSO of case (3) of the five-shell nonuniform optimized lens

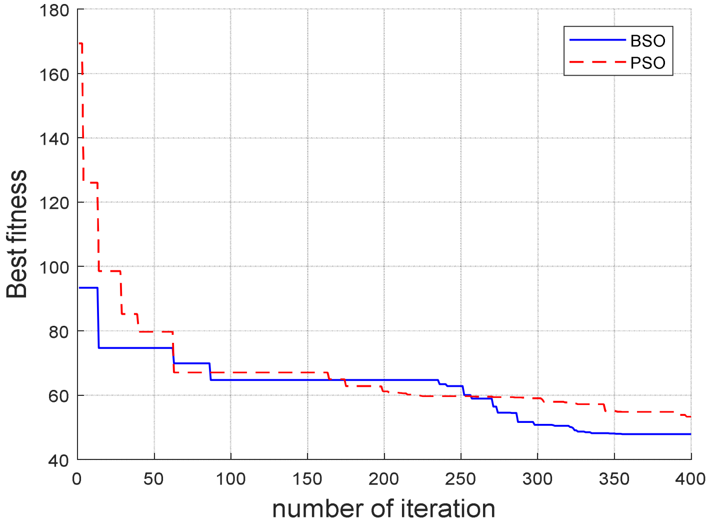


Figure 5-13: Convergence curve comparison between BSO and PSO of case (3) of the five-shell nonuniform optimized lens

Table 5-6: Optimized Variables of case (4) of the Nonuniform Lens Optimization Using (a) BSO (b) PSO

Shell	ε_r	t / λ
1	1.1489	2.556
2	1.5629	2.3548
3	1.6714	1.5055
4	1.787	1.3696
5	1.8851	7.004

(a)

Shell	ε_r	t / λ
1	1.3811	1.0029
2	1.0558	2.264
3	1.5445	3.0220
4	1.6686	1.5218
5	1.7882	6.9893

(b)

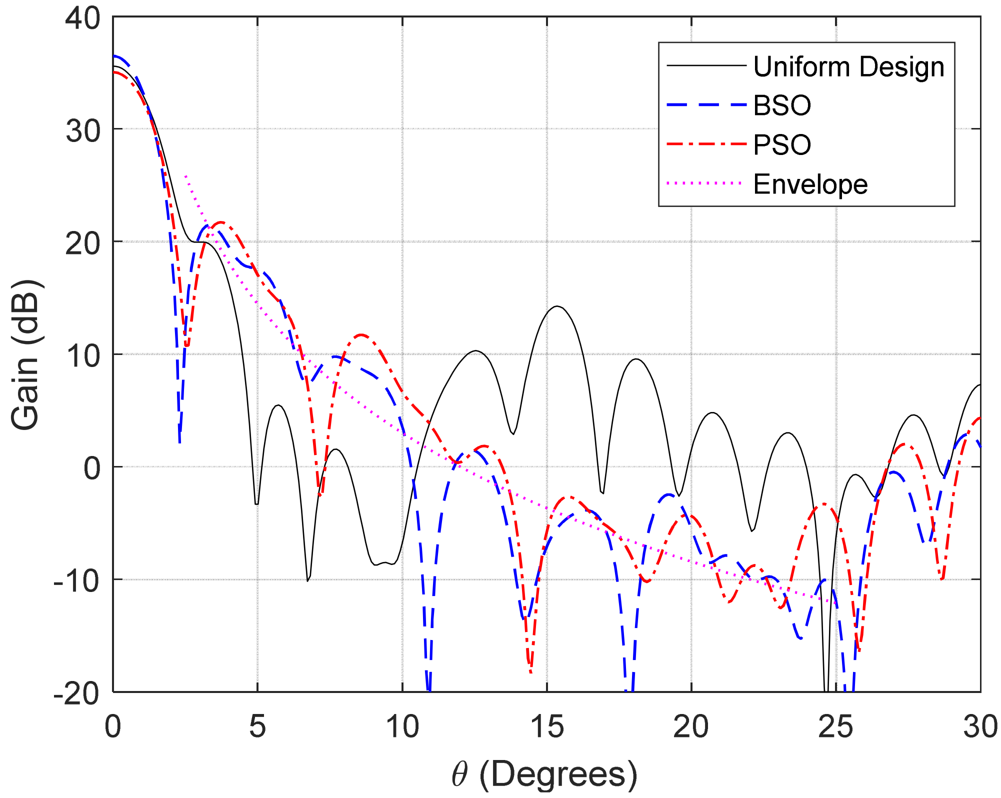


Figure 5-14: Gain pattern comparison between BSO and PSO of case (4) of the five-shell nonuniform optimized lens

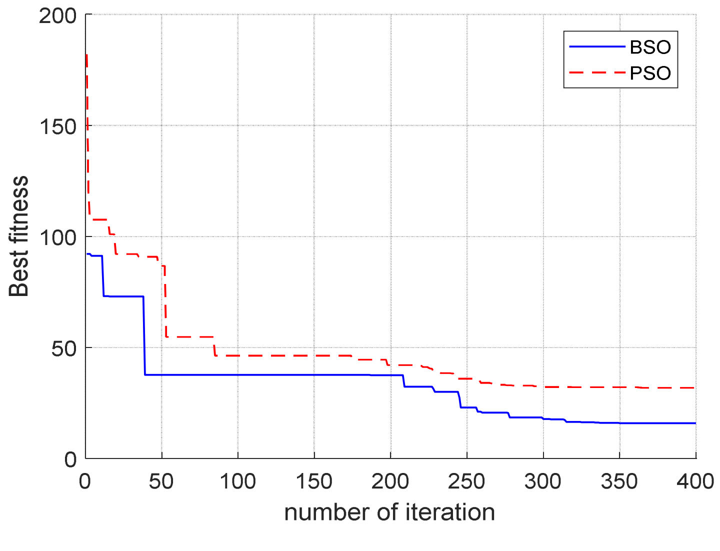


Figure 5-15: Convergence curve Comparison between BSO and PSO of case (4) of the five-shell nonuniform optimized lens

5.3.3 BSO and GA Comparison Using Different Fitness Function

Since this chapter is inspired by [16], it is great to compare the results of BSO with the reported results of GA for extra level of testing. The reason for doing this comparison in a separate section is due to the fact that the fitness function used in their work is different. Their fitness function is defined as follows

$$F = \alpha \max(G_0) + \beta \min(E) \tag{5.9}$$

where α and β are a weighting coefficient of the gain and grating lobes level. α is set to 1 and β is set to 0.5, to achieve better gain. Furthermore, it should be noted that in the referred paper, the fitness function is used as an adaptive fitness function. This means α and β can change during the course of optimization. In their work, β is set to 0 during the first 20% number of fitness function evaluations. This method is used to keep the antenna gain in a high level while decreasing the sidelobe levels. After that, β is set to 0.5. This method is beyond the focus of the study, and BSO is run with static values of α and β in this section. This might put BSO at a disadvantage, but it will still be interesting how BSO preforms compared to GA. Also, Case (2) in Table 5-2 is chosen in this comparison because it serves as more applicable design in real world application.

The optimized variables are shown in Table 5-7. Figure 5-16 shows the gain pattern of both designs. BSO's design has a slightly better gain, 0.3 dB higher, although the adaptive method was used for GA to maintain a high gain. The side lobes levels of both designs are fairly similar.

Table 5-7: Optimized Variables of Nonuniform Lens Optimization of Case (2) Using the New Fitness Function

Shell	ε_r	t / λ
1	1.1226	1.7424
2	1.2047	1.1660
3	1.7608	3.9464
4	1.8549	1.6727
5	1.9776	6.4725

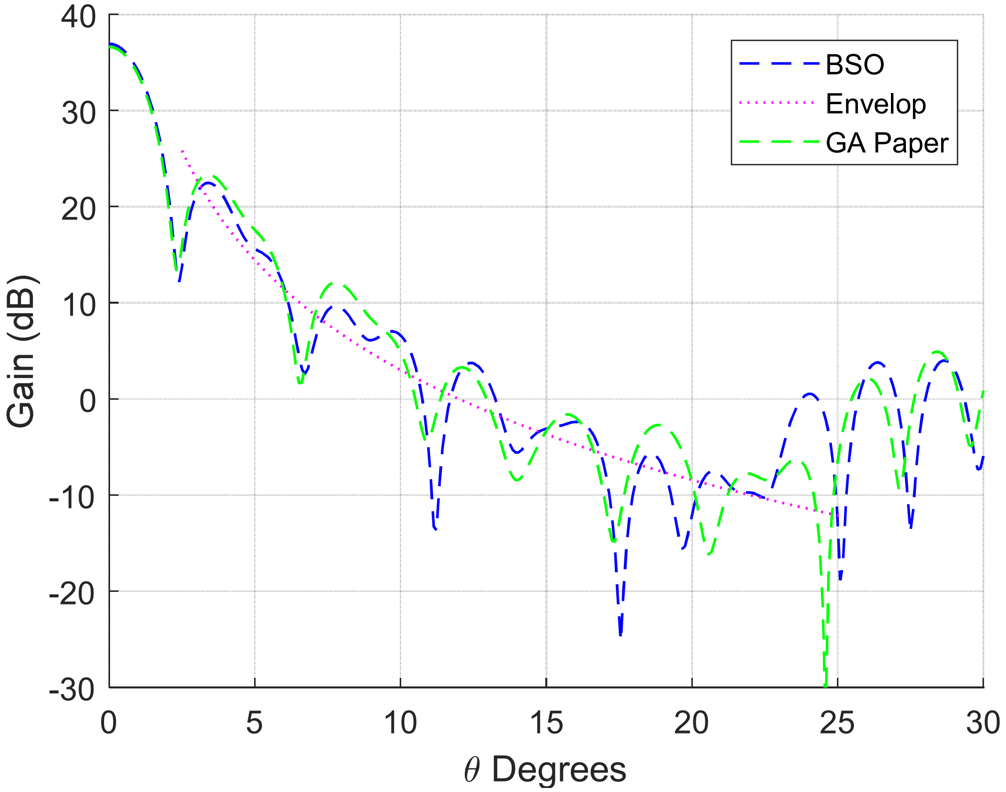


Figure 5-16: Gain pattern comparison between BSO and GA in [16] of case (2) of the five-shell nonuniform optimized lens

5.4 Further Discussion

In this chapter, the importance of optimization tools for electromagnetic applications is demonstrated by improving the radiation performance of a five-shell 30λ -diameter spherical lens. That was motivated by comparing a five-shell 30λ -diameter Luneburg Lens with a ten-shell lens that has the same diameter; the later design has a superior radiation performance. Furthermore, the performance of BSO has been examined using four different configurations of Luneburg Lens by comparing BSO with PSO. In one case, BSO showed significantly better results than PSO; a case where PSO fell in a local minimum trap during the early stage of optimization. Compared to the results obtained by BSO, it is reasonable to say the PSO failed this optimization test. In other two cases, BSO was able to attain slightly better designs than PSO. And, in one case both BSO and PSO had similar performance. The four cases showed that BSO at worst is as good as PSO, and better than PSO at best. PSO showed to be prone to fall in a local optimum trap. Since there is no prior knowledge of the fineness function behavior in electromagnetic problems most of the time, BSO is a better candidate than PSO as an optimization tool based on the results presented in this chapter.

CHAPTER 6

6 Binary BSO

All the previous discussions were based on applying BSO using a real-valued vector representation of the optimized variables such as length, thickness, permittivity…etc. often times, the variables of optimization are discrete values. An example of problems that have discrete variables is array thinning [15]. Other examples are shown in [18] where PSO is used to optimize binary-valued problems. Motivated by the need of binary optimization, and the performance of real-valued BSO, we introduce a novel binary brainstorm optimization (BBSO) for discrete optimization. It should be noted that to the best of our knowledge this is the first attempt to convert BSO into a binary-valued algorithm.

In this chapter, a novel binary brainstorm optimization (BBSO) is presented in detail by transforming BSO different operations from its real-valued context into a binary context. Also, each binary operation is investigated for a better overall performance of BBSO. Then, a two-dimensional SINC function is optimized using BBSO as a simple introductory problem of optimization.

6.1 Transforming BSO Into Binary Coding

The BSO algorithm and its all operations are based on solutions that are evaluated using real-valued variables in a continuous search space $X \in \mathbb{R}^D$. Therefore, to use BSO with binary encoding, those operations need to be transformed to consider that the solutions are in binary strings in which each bit can take either two values of 0 and 1. The operations that need to be modified are: the grouping operation, combining two ideas, and solution generation by adding

Gaussian noise.

In real-valued BSO, the measure of similarity among solutions are simple and intuitive. The grouping was based on the distance between those points, in which the City Block distance was used. The City Block distance between two points, a and b, in D dimensions is calculated as:

$$\sum_{i=1}^{D} \left| a_i - b_i \right|. \tag{6.1}$$

Since in binary space, each bit can take either 0 and 1, the Hamming distance is used to determine the similarity among solutions. the Hamming distance between two binary strings of equal length is the number of positions at which the corresponding bits are different. For example, the Hamming distance between the two binary strings ‘110’ and ‘111’ are 1 since the third bit in the two strings are different. In BBSO, the *k*-medoids clustering based on the Hamming distance algorithm that is part of the Statistics and Machine Learning Toolbox in MATLAB is used to group the binary ideas [72]. Since this is a first attempt into using BSO in binary form, the clustering technique could be a potential area where BBSO can be improved.

Another modification is needed during the idea selection process, specifically, when two ideas are selected as the source of the next idea generation. In real-valued BSO, the two ideas are combined using random weighted sum as shown in Eq. (2.5). In binary, this operation is analogous to crossover in Genetic Algorithm, where in BBSO a random integer is chosen that corresponds to the position of a bit in the binary string. The selected point is used to break the binary strings of the two ideas, and the selected parts are combined to form one idea. This process is shown in Figure 6-1.

Lastly, a new idea in real-valued BSO is generated by adding a Gaussian noise to the

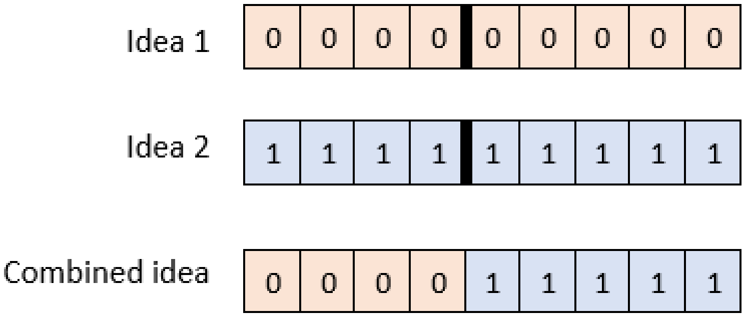


Figure 6-1: Single-Point idea combination in BBSO

selected idea, which is translated in Eq. (2.3). The purpose of this noise is to search for ideas around the selected idea with high probability of the new idea being close to the selected idea in the search space. Also, the value of the noise added is controlled by the step function i.e. the added noise to the selected idea is decreasing after each iteration. In BBSO, the previous statements need to the translated to the binary world. As previously discussed, each dimension i.e. bit in the binary string can take either of two values of 0 or 1. In the proposed BBSO, adding noise to an idea that is a string of binary bits is basically changing bits from 0 to 1 and 1 to 0. Determining how many and which bits are changed should be controlled by similar behavior of decreasing values of the added noise in real-valued BSO. The maximum probability of a bit is flipped should be decreasing after each iteration. Hence, the term $\xi * \mathrm{N}(\mu,\sigma)$ in the idea generation in real-valued BSO needs to be restructured in BBSO to accommodate this behavior. In BBSO, the probably of a bit is flipped in the selected idea is calculated using

$$P_{flip} = \frac{Q}{D}, \tag{6.2}$$

where D is length of the binary string i.e. the dimension of the selected idea, and Q is a real number calculated using the modified step function for BBSO

$$Q = \xi \cdot |\,\mathrm{N}(0,1)\,|. \tag{6.3}$$

It should be noted that the Gaussian noise can return negative values. Hence, the absolute value is taken. And the modified step function ξ is calculated

$$\xi(t) = \gamma \cdot (D-1) \cdot \exp\left(1 - \frac{T}{T-t+1}\right) + 1, \tag{6.4}$$

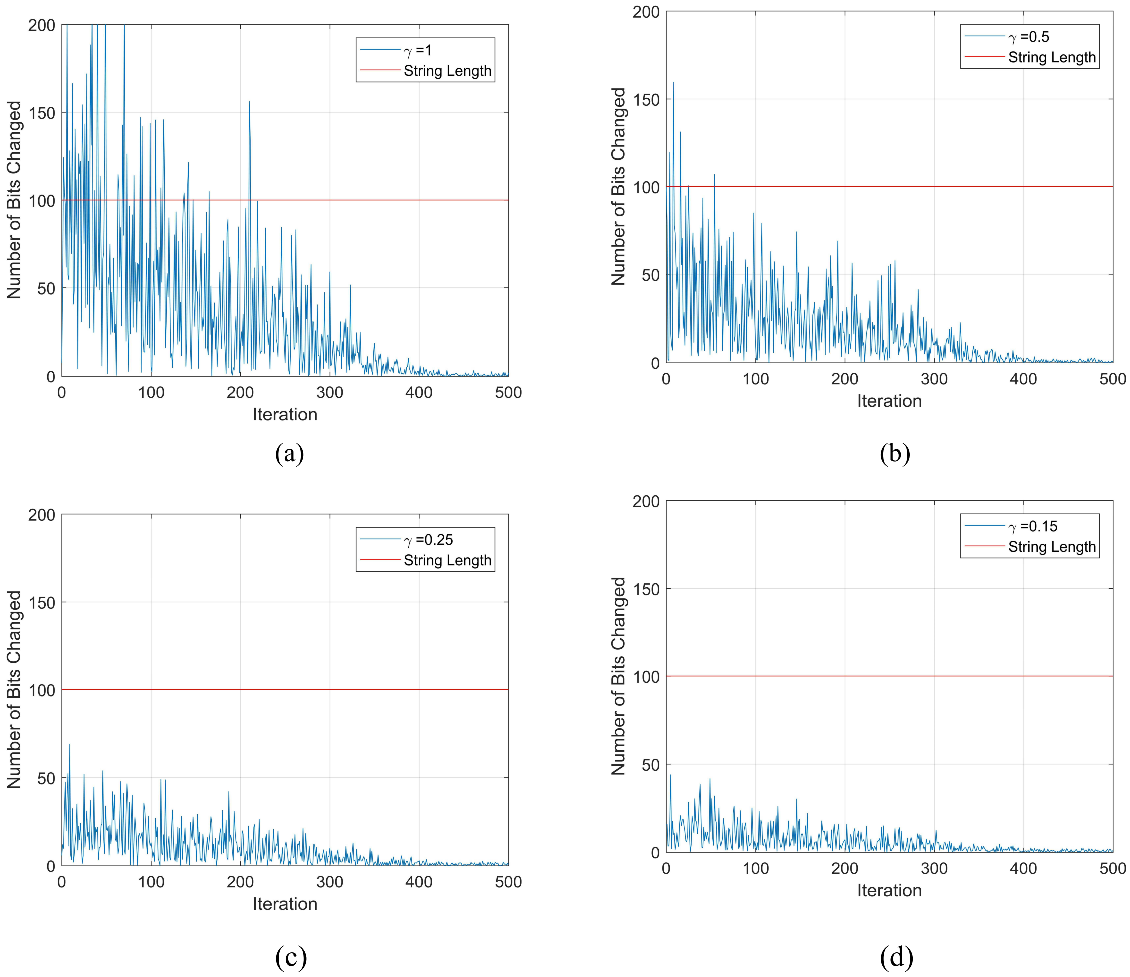
where D is length of the binary string, and γ is a scaling factor. Since the whole notion of adding noise is to explore ideas in the neighborhood of the selected idea, it would be contradictory if every bit in the binary string is flipped, hence the scaling factor γ is introduced. Additionally, the algorithm ensures that at least one bit is flipped to ensure a new solution is generated. Figure 6-2 shows a simulation of values of Q in the optimization process for a binary string with length of 100 bits using different values of γ. Figure 6-2c shows that $\gamma = 0.25$ returns a reasonable behavior of Q.

6.2 Optimizing Two-Dimensional SINC Function Using BBSO

A simple way to demonstrate the capability of BBSO is to use a mathematical function as they are very fast in computation time. Hence, BBSO can run dozens of time to avoid any statistical error in the results. The selected function is a SINC function that has the form

$$f_{SINC}(x) = \prod_{i=1}^{D} \left| \frac{\sin(\pi(x_i - 3))}{\pi(x_i - 3)} \right|, \quad x \in [0,8]^D. \tag{6.5}$$

A 2D f_{SINC} is chosen to be optimized, which has a maximum at $x_1 = 3.0, x_2 = 3.0$. The behavior of 2D f_{SINC} is shown in Figure 6-3. Since f_{SINC} is evaluated in a real-valued search space, a decoding method is needed to transfer the binary data in BBSO to real decimal numbers to be evaluated by f_{SINC}. A binary decoding method from an N bit string to a set of real decimal numbers



(a)

(b)

(c)

(d)

Figure 6-2 Simulated value of Q in each iteration with different value of γ for binary string of length of 100 bits

is given by [13]

$$x_i = \left(\frac{x_{\max} - x_{\min}}{2^N - 1} \right) g \sum_{n=0}^{N-1} \left(2^n b_n \right) + x_{\min} \tag{6.6}$$

An example of decoding a binary idea to a pair of decimals numbers in a real-valued, and two-dimensional search space is shown in Figure 6-4. In this optimization, each real dimension is represented by 16 binary bits which results into each idea in BBSO is a 32-bit binary string. Also, BBSO is run with a popualtion of 20 ideas and 4 clusters for a maximum number of iteration of

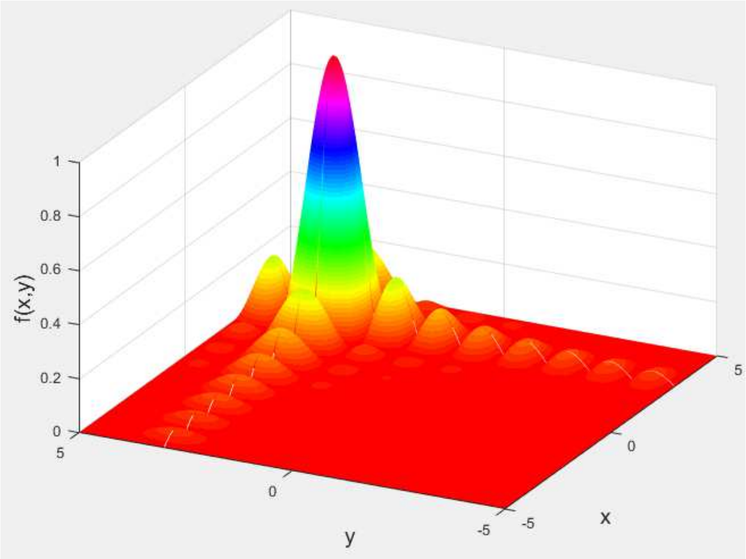
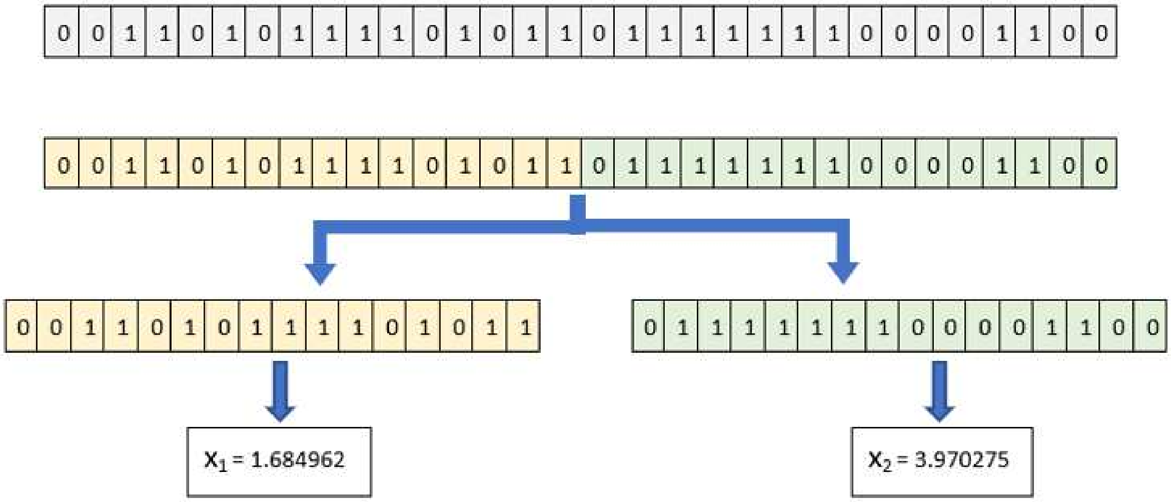


Figure 6-3: Two-Dimensional SINC function shown in Eq. (6.5)



0 0 1 1 0 1 0 1 1 1 1 0 1 0 1 1 0 1 1 1 1 1 1 1 0 0 0 0 1 1 0 0

0 0 1 1 0 1 0 1 1 1 1 0 1 0 1 1 0 1 1 1 1 1 1 1 0 0 0 0 1 1 0 0

0 0 1 1 0 1 0 1 1 1 1 0 1 0 1 1

0 1 1 1 1 1 1 1 0 0 0 0 1 1 0 0

$X_1 = 1.684962$

$X_2 = 3.970275$

Figure 6-4: Decoding a 32-bit binary string into a 2D real-valued search space. Each real dimension is represented by 16-bit binary string and $x \in [0,8]^D$

50. Moreover, to further investigate the effect of the scaling factor γ, the function is optimizaed with different values of γ. Furthermore, BBSO is written as a minimzer algorithm, therefore, the fitness function is multiplyed by -1.

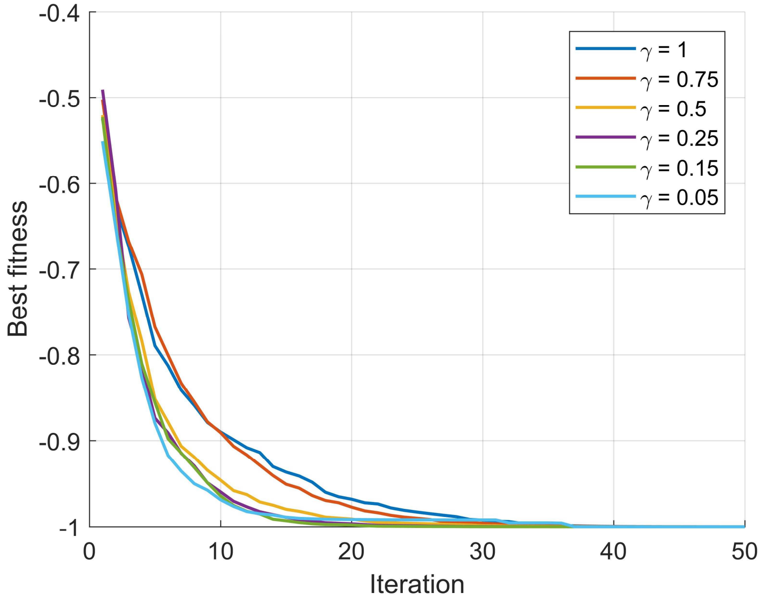


Figure 6-5: Optimization convergence of averaged 100 runs of BBSO of 2D SINC function with different values of γ.

To overcome statistical errors, BBSO is applied to the fitness function 100 times. The average of those 100 independent runs is shown in Figure 6-5. The results demonstrates that BBSO has a fast convergence. Also, the effect of the scaling factor γ is notiecble; the smaller the value the better. However, there is no signifact adavantgae for $\gamma \leq 0.15$. This example demonstrates the ability of BBSO to have a good performance and a fast convergence for binary-valued optimization problems. Armed with the obtained information in this example, BBSO is ready to be applied to electromagnetic applications where the variables of optimization are discrete.

CHAPTER 7

7 Array Thinning Using BBSO

In chapter 4, the Yagi-Uda was an example of antenna arrays where certain directivity characteristics can be achieved by increasing the number of radiative elements. Different patterns of an array with identical elements can be achieved by changing the type of element (dipole, patch, etc.), the geometrical configuration, spacing between elements, and the excitation of the individual elements. The total far-field of an array of identical elements can be calculated using pattern multiplication

$$\vec{E}_{tot}(\theta,\phi) = \vec{E}_{src}(\theta,\phi)\cdot AF(\theta,\phi) \tag{7.1}$$

where $\vec{E}_{src}(\theta,\phi)$ is the far-field radiation patter of a single element in the array, and $AF(\theta,\phi)$ is the array factor. Generally, the array factor is a function of the excitation magnitude, phase of each elements, and the spacing between each element. The feature of array factor allows designers to synthesize desired patterns by manipulating those variables. Therefore, antenna arrays have been widely solved by global optimization algorithms [15, 18, 12, 30]. One of the desired characteristics in antenna arrays is lower sidelobe levels (SLL). In periodic arrays where the spacing of the elements are identical, the sidelobe levels are suppressed by optimizing amplitude and/or phase excitation of the individual elements. Another approach to minimize the SLL is to use aperiodic arrays where the spacing between the elements is not uniform. Such aperiodicity can be achieved by either nonuniform array or thinned array. In nonuniform arrays, the number of the total elements in the array is fixed while changing the spacing between the elements is changed such that the elements have unequal spacing. In thinned arrays, starting from a periodic array, different elements in the array are turned OFF, which results in lower number of elements needed. The two different approaches are shown in Figure 7-1. The thinned array approach requires a discrete optimization where each element can have either of two states. Hence, array thinning serves a good candidate to test BBSO. The array factor for a planar array in the x-y plane is calculated by [15]

$$AF(\theta,\phi)=4\sum_{n=1}^{N}\sum_{m=1}^{M}a_{mn}\cos[(2m-1)\pi d_x\sin\theta\cos\phi]\times\cos[(2n-1)\pi d_y\sin\theta\sin\phi] \tag{7.2}$$

where

a_{mn}= Amplitude of element mn , 1 for ON or 0 for OFF

M = number of elements in the y-direction,

N = number of elements in the x-direction,

d_y = spacing in the y-direction,

d_x = spacing in the x-direction,

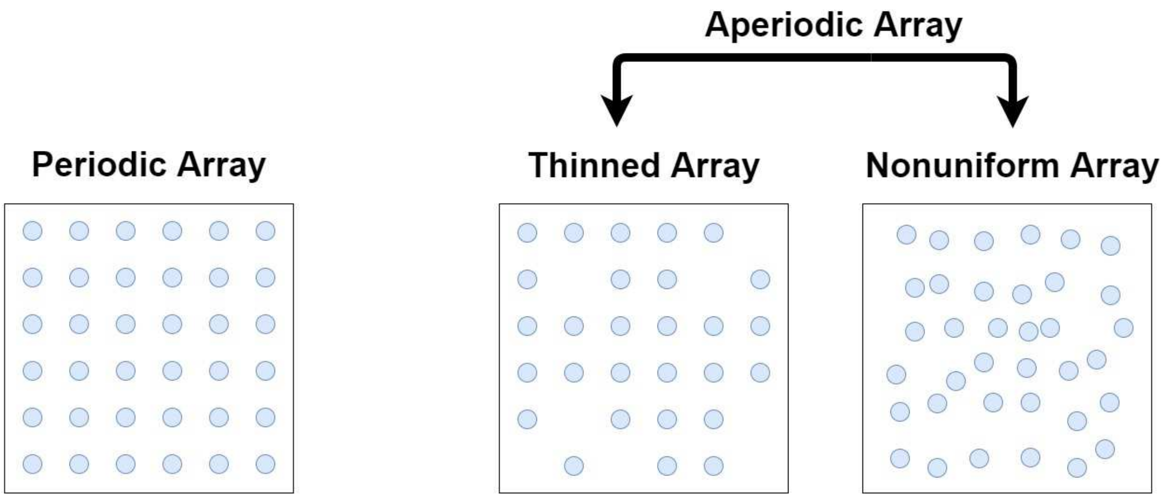
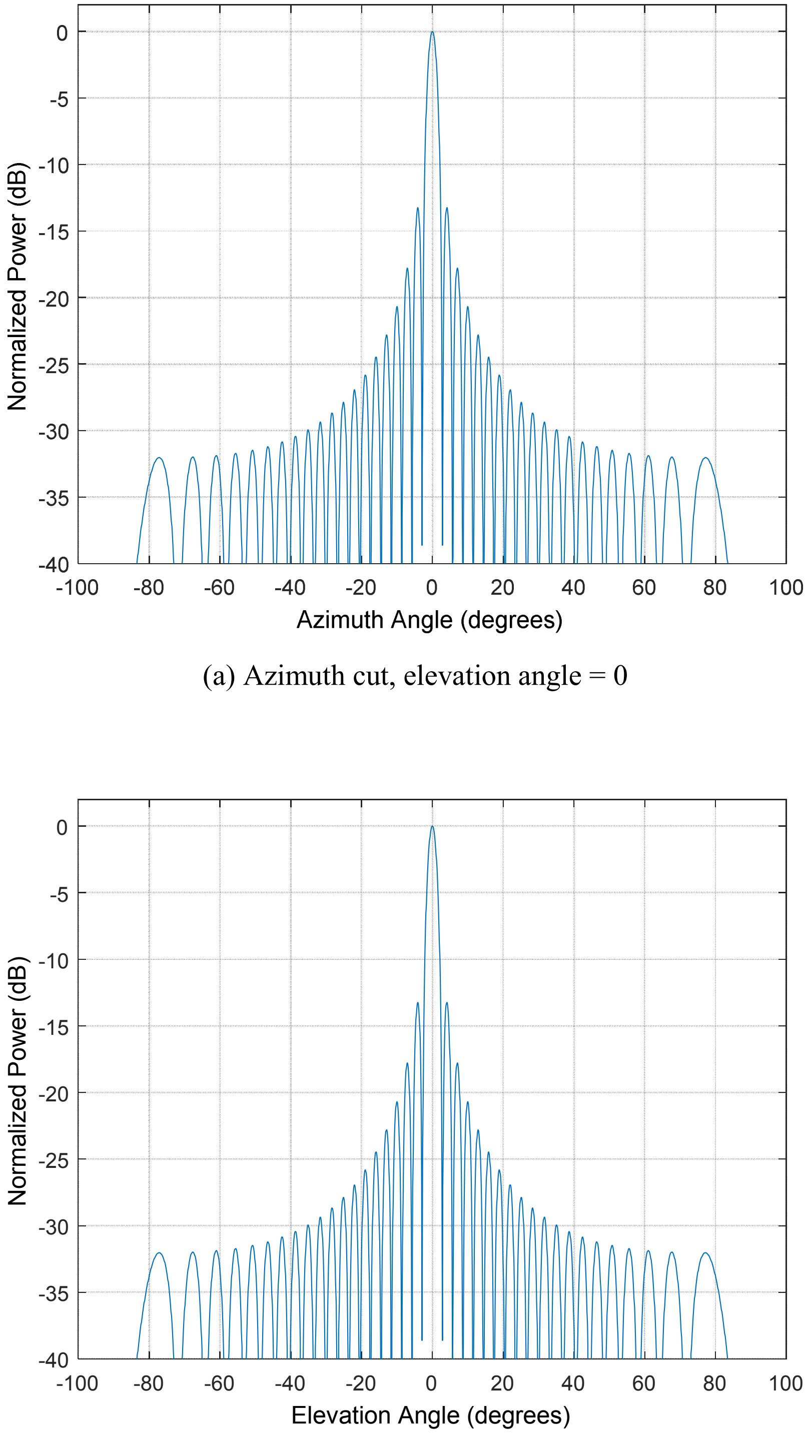


Figure 7-1: Different types of arrays based on elements spacing

7.1 Developing the optimization problems

In this example, BBSO is used to thin a 40×40 uniform periodic planar array of isotropic sources with spacing of $\lambda/2$ between each element with the purpose of achieving lower relative SLL. Thus, the fitness function in the optimization is simply the worse SLL in the two orthogonal planes (azimuth and elevation). The array is simulated using MATLAB's Phased Array System Toolbox [73]. Figure 7-2 shows the radiation pattern of the full array placed in the yz-plane simulated in both planes (azimuth and elevation cut). The SLL is about 13 dB for the full 40×40 planar array.

In BBSO, an idea is a binary string, therefore, a planar array is transformed into a linear array; if an element has the ON state, it will take the value 1 in the binary string, and 0 if it has the OFF state. Since each element can be either of the two states, this results into 2^{1600} different possible combinations. One can take the advantage of symmetry in planar arrays; the search space is reduced by applying the thinning procedure to only quarter of the array, therefore, this results in 2^{400} different possible combinations. Also, a proper initialization is a big factor of the convergence rate when optimizing thinned arrays by having more elements in the center than at the edges [13, 15]. Hence, the BBSO will starts with partially pre-initialed ideas that have their center elements turned ON in the beginning of search. BBSO parameters used in this optimizing are: population number is 200, number of clusters is 4, and $\gamma = 0.25$. The optimization will run for 100 iterations.



(a) Azimuth cut, elevation angle = 0

(b) Elevation cut, azimuth angle = 0

Figure 7-2: Radiation pattern of 40x40 uniformly spaced planar array: (a) azimuth cut (a) elevation cut

7.2 Optimization Results

A relative SLL of -23.86 dB is achieved with 47.25% fill rate. Figure 7-3 shows BBSO convergence where the fitness function is the relative SLL. BBSO was able to achieve a relative SLL of -20 dB in 21 iterations. Figure 7-4 shows the optimized design of a thinned 40x40 planar array. The elements with ON state are represented by white squares while elements with OFF state are represented by black squares. Figure 7-5 shows the difference in the radiation patterns between a full 40x40 uniformly spaced planar array with relative SLL of about -13 dB and optimized thinned array with relative SLL of -23.86 dB. This example clearly shows the ability of BBSO in optimizing discrete problems with a good convergence rate.

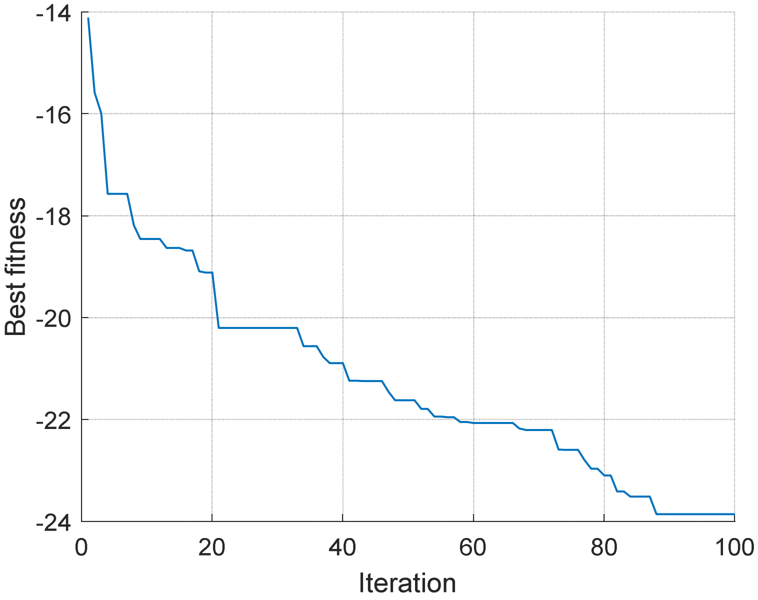


Figure 7-3: Convergence curve of BBSO where the fitness function corresponds to the relative SLL.

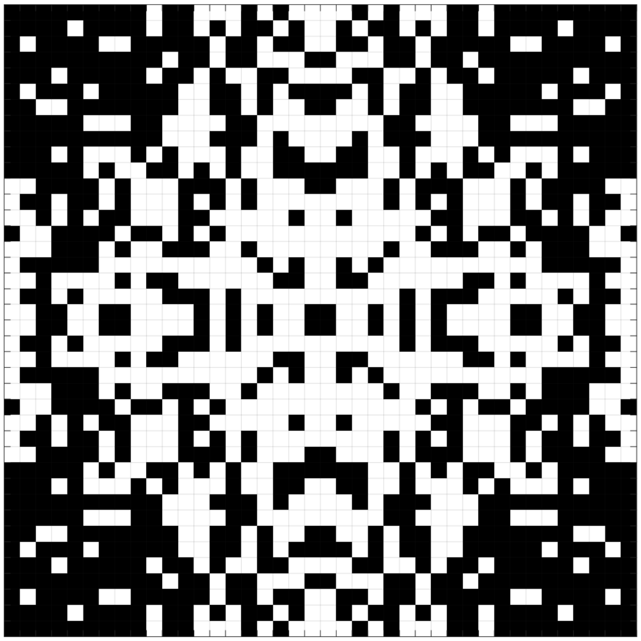
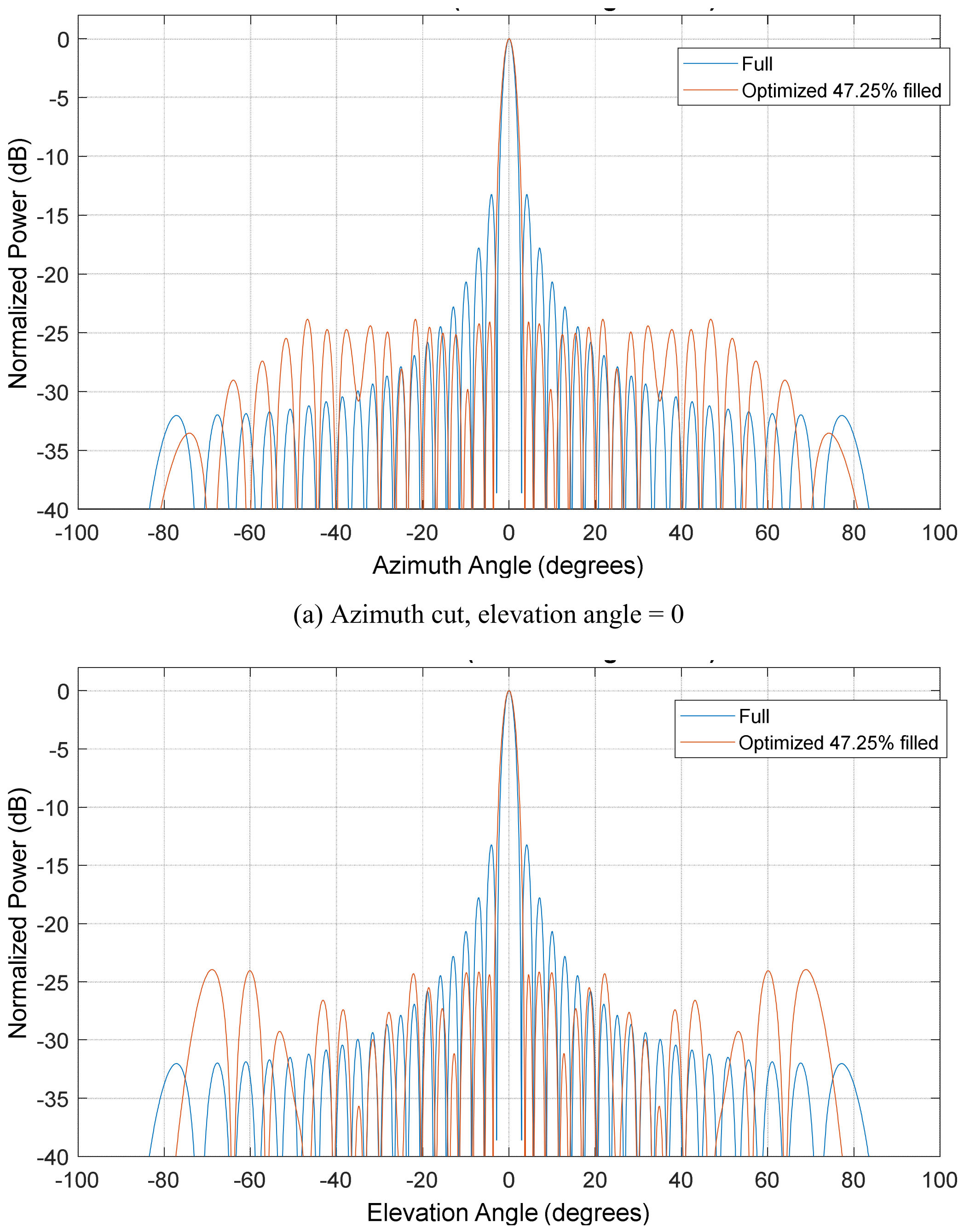


Figure 7-4: Optimized design of a thinned 40x40 planar array. elements with ON state are represented by white squares while elements with OFF state are represented by black squares.



(a) Azimuth cut, elevation angle = 0

(b) Elevation cut, azimuth angle = 0

Figure 7-5: Radiation pattern comparison between a full 40x40 uniformly spaced planar array and optimized thinned array: (a) Azimuth Cut (a) Elevation Cut

CHAPTER 8

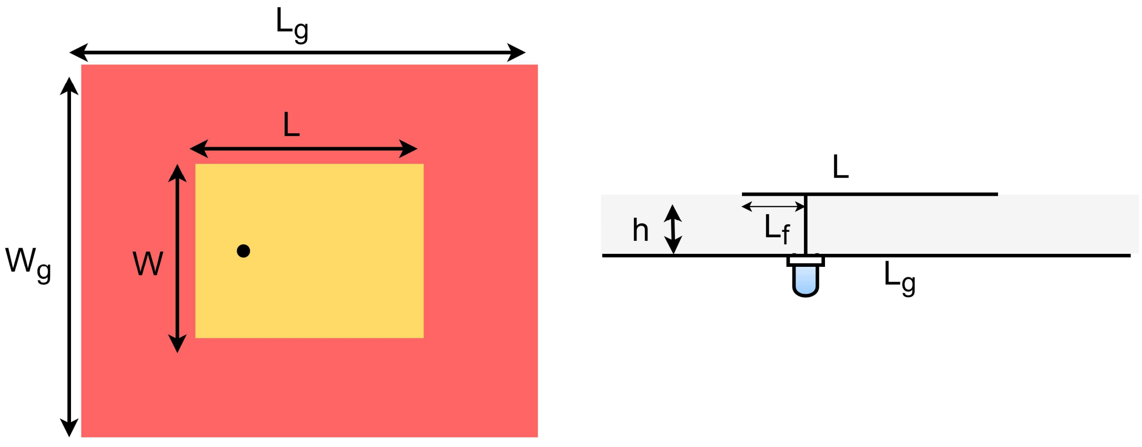
8 Pixel Patch for Dual-Band Applications

8.1 Introduction to Patch Antennas

A planar rectangular sheet mounted over a large sheet of metal is all what is needed to create one of if not the simplest antenna i.e. patch antenna. This type of antenna is used a lot in various types of applications such as satellite, missile, aircraft, mobile, and wireless communications applications. Some of the features that led to the widespread of the antenna are [58]

- The low cost of manufacturing this antenna. This feature makes them attractive for applications that require large quantity.
- The antenna is low profile and can be mounted to planar and non-planar surfaces. Therefore, the integration of the antenna with MMIC designs is simple.
- The antenna can be designed with different resonant frequency, polarization, impedance, which makes it very versatile.

Patch antennas can be classified based on the feeding method, with the probe-fed being the simplest one as shown in Figure 8-1. Nonetheless, those advantages of patch antennas come at a cost; patch antennas have disadvantages such as low efficiency, high cross polarization, and low bandwidth [58]. In wireless communications, the higher the bandwidth the more information that can be transmitted. There are various methods to increase the bandwidth and introduce new resonance frequencies of patch antennas. Increasing the height of substrate is a simple way of increasing the frequency bandwidth but to a certain limit [58]. Another way of achieving



(a) (b)

Figure 8-1: Coaxial probe-fed rectangular patch. (a) top view, (b) side view

broadband patch antennas is by adding passive resonating elements to the antenna; however, this means also increasing the size of the antenna, which can be undesirable in certain applications [13]. Another common method of adding more bandwidth is by introducing slots to the antenna in which a new resonant frequency is added [13]. A popular application of the use of slots into patch antenna is E-patch antenna [74, 25]. Optimization tools can be utilized by randomly adding slots to the patch [13, 75, 22], which is the method that will be used in this chapter. In the next section, BBSO will be used to find the slots placements of a simple rectangular patch antenna by discretizing the patch into smaller elements ‘pixels’.

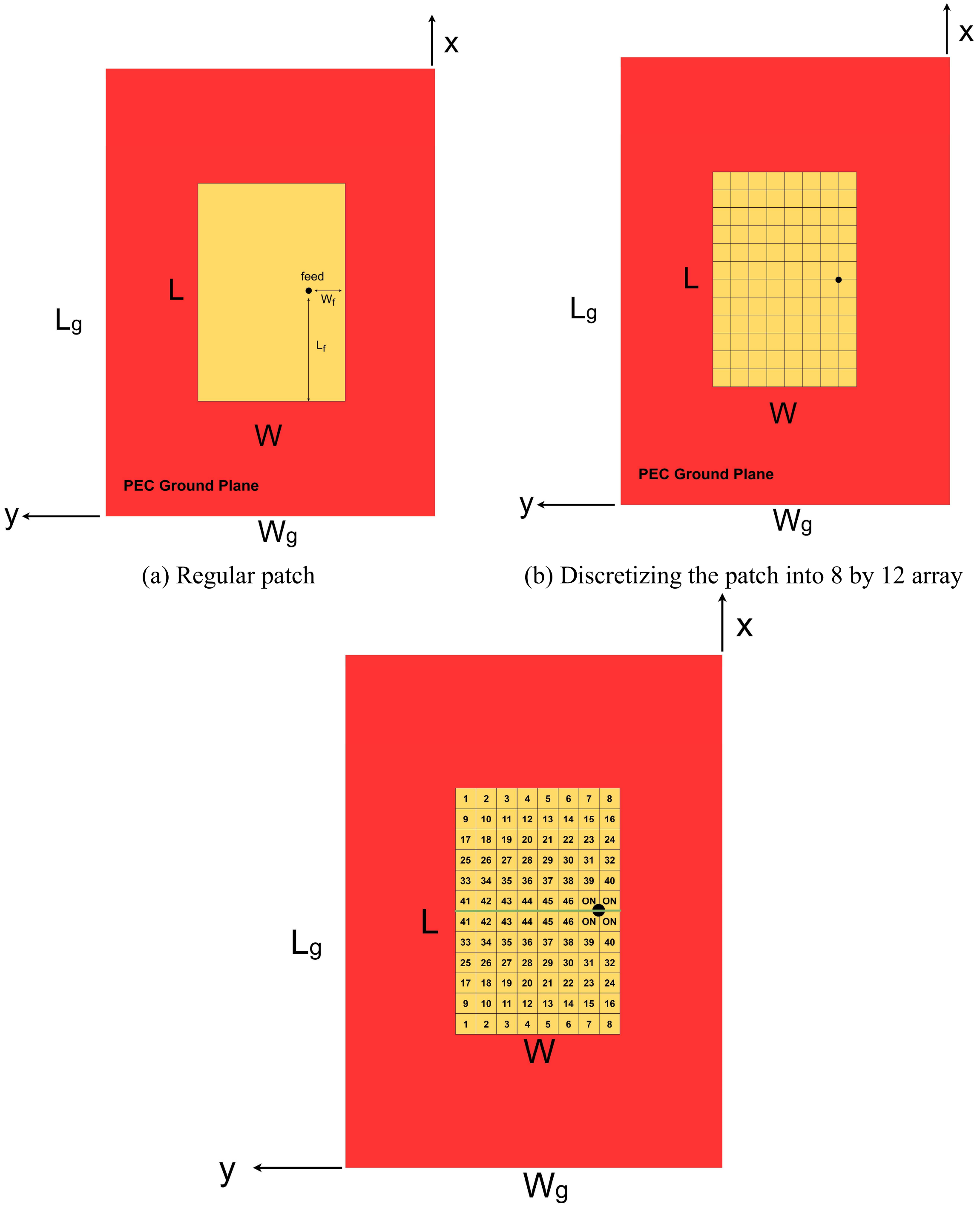
8.2 Developing the Optimization Problem

To demonstrate its capability in optimizing binary-valued problems, BBSO will used to design a dual-band antenna element for wireless communication (1.9 GHz and 2.4 GHz) applications. The design is inspired by the work documented in [75] where the authors utilize Genetic Algorithm (GA) to design the same antenna. However, the authors use method of moments

(MoM) to evaluate the structure while in this work the High Frequency Spectrum Simulation (HFSS) is used [76]. HFSS uses the finite element method (FEM) as a numerical solver. The authors start form E-patch antenna design to take the relevant dimension of the representative patch antenna. Then, the rectangular patch is discretized into 8 by 12 array with each cell in the array is a $6\,\mathrm{mm}\times 6\,\mathrm{mm}$ square. The cell can take either of two values in the optimization (ON/OFF), thus BBSO is used. To ensure that the probe is electrically connected to the patch, the 4 cells that are connected to the probe will always stay ON. Additionally, to avoid high cross-polarization, structure symmetry is imposed along the E-plane. Hence, the number of pixels that will be optimized by BBSO is 46 pixels (reduced to half). This will lead to 2^{46} possible solutions. This process is illustrated in Figure 8-2. The structure dimensions are: $L_g = 150\,\mathrm{mm}$ and $W_g = 110\,\mathrm{mm}$ for the ground, $L = 72\,\mathrm{mm}$, $W = 48\,\mathrm{mm}$, $h = 15\,\mathrm{mm}$ for the patch, and the feed location is $L_f = \frac{L}{2} = 36\,\mathrm{mm}, W_f = 5\,\mathrm{mm}$. As stated, the goal is to obtain a design that operates in the desired bands of interest (1.9 GHz and 2.4 GHz). Therefore, the fitness function for the dual-band antenna is written as

$$f = \max\left(S_{11}\,|_{f=1.9GHz}\,,\, S_{11}\,|_{f=2.4GHz}\right). \tag{8.1}$$

An idea in BBSO will be a string of 46 binary bits to represent a solution. BBSO parameters used in this optimizing are: population number is 25, number of clusters 4, and $\gamma = 0.25$. The optimization will run for only 100 iterations due to time constraints as a single function evaluation can take from 7 to 15 minutes using HFSS. This will result into 2500 function evaluation, which is only 5% of what is used in [75] where a parallelization scheme was used to speed up the whole optimization process. In this work, the luxury of parallelization is not used so that BBSO will have to find a good design in much less number of function evaluations.



(a) Regular patch

(b) Discretizing the patch into 8 by 12 array

1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16
17	18	19	20	21	22	23	24
25	26	27	28	29	30	31	32
33	34	35	36	37	38	39	40
41	42	43	44	45	46	ON	ON
41	42	43	44	45	46	ON	ON
33	34	35	36	37	38	39	40
25	26	27	28	29	30	31	32
17	18	19	20	21	22	23	24
9	10	11	12	13	14	15	16
1	2	3	4	5	6	7	8

(c) Imposing structure symmetry

Figure 8-2: Process of establishing the optimization problem starting from a regular patch ending with a 2-D rectangular array of 46 (ON/OFF) metallic elements with field symmetry condition along the E-plane.

8.3 Optimization Results

Figure 8-3 shows BBSO's convergence where the fitness function is the worst return loss of the two frequencies of interest. BBSO was able to achieve a satisfactory design ($S_{11} < -10 \text{ dB}$) within 41 iterations. The return loss of the optimized design is shown Figure 8-4. The return loss at the two frequencies (1.9 GHz and 2.4 GHz) are nearly the same with $S_{11} \cong -11 \text{ dB}$. Although it was completely unintentional, the design has two resonance frequencies close to each other (1.96 GHz and 2.28 GHz) which led to a wide bandwidth (from roughly 1.9 GHz to 2.4 GHz) where $S_{11} < -10 \text{ dB}$. Figure 8-5 shows how the design looks in terms of pixels. The elements with ON state are represented by yellow squares while elements with OFF state are removed.

BBSO was able to perform well even though the maximum number of function evaluations is much less than the number used in [75]. BBSO needed only 41 iterations (1025 function evaluations) to find a satisfactory design out of 2^{46} possible solutions. Moreover, this example shows the beauty of global optimization tools, a design was created from what it can be considered a completely bad design. Figure 8-4 shows the return loss of the original patch (all pixels 'ON') for comparison. Additionally, the example has shown the integration of BBSO with a common EM solver, HFSS. The two tools combined can be used for additional designs for more interesting results.

The radiation characteristics of the antenna at frequencies 1.9 GHz and 2.4 GHz are shown in Figure 8-6 and Figure 8-7, respectively. Figure 8-6a shows 3D radiation pattern of the optimized antenna; the antenna has directivity of 7.35 dB at 1.9 GHz and 8.64 dB at 2.4 GHz. Figure 8-6a shows the normalized H-plane (xz-plane in Figure 8-2) co-pol (E_ϕ) and cross-pol (E_θ) far field patterns. The simulated cross-pol levels are roughly 5.5 dB down from the co-pol peak. This is

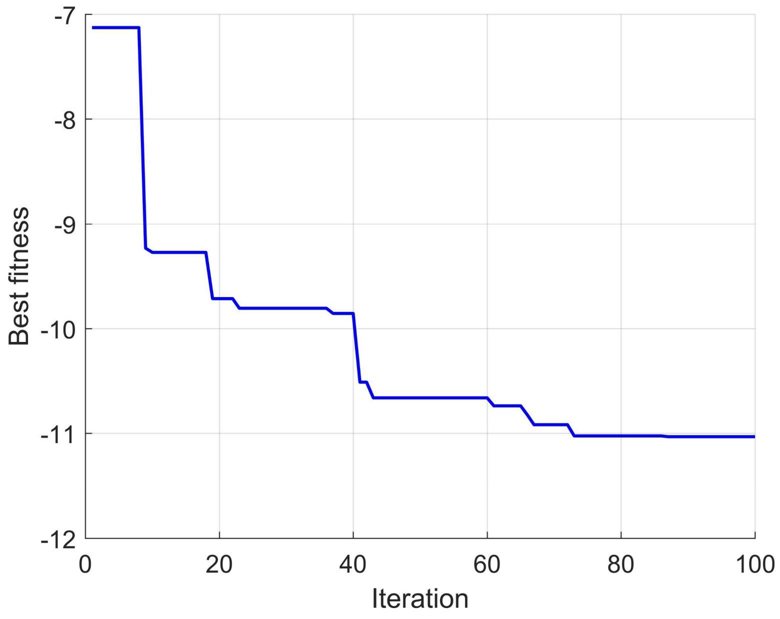


Figure 8-3: BBSO Convergence

expected since rectangular patch antennas generally suffer from high cross-pol levels in the H-plane [77] in addition to the optimization was done solely for matching. Since the cross-polarization is very low (at least 45 dB below) in the other orthogonal plane i.e. E-plane, the E-plane patterns are not shown.

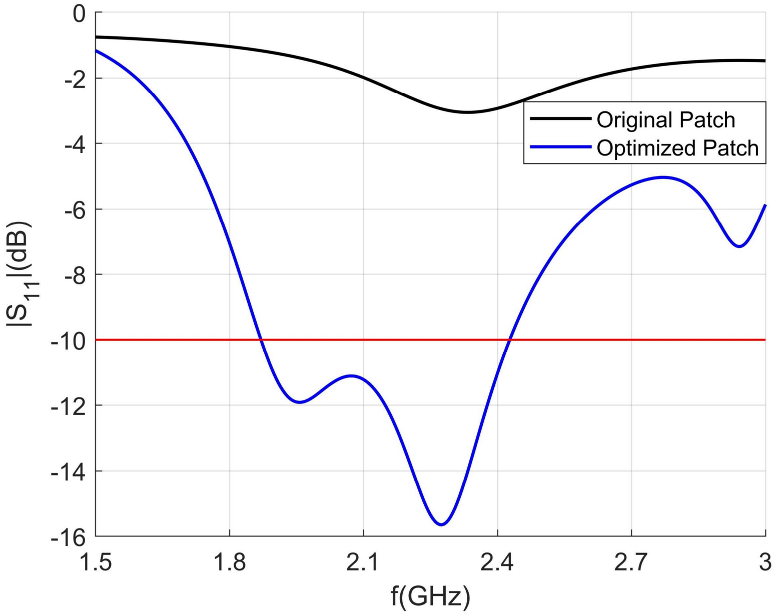


Figure 8-4: Input return loss of the optimized patch. The original patch where all the pixels are "ON" is shown for comparison.

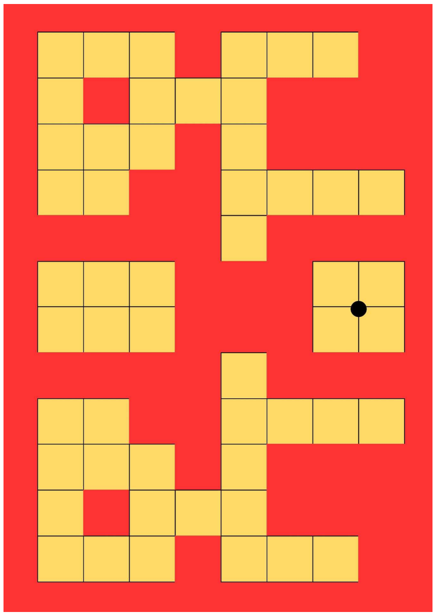
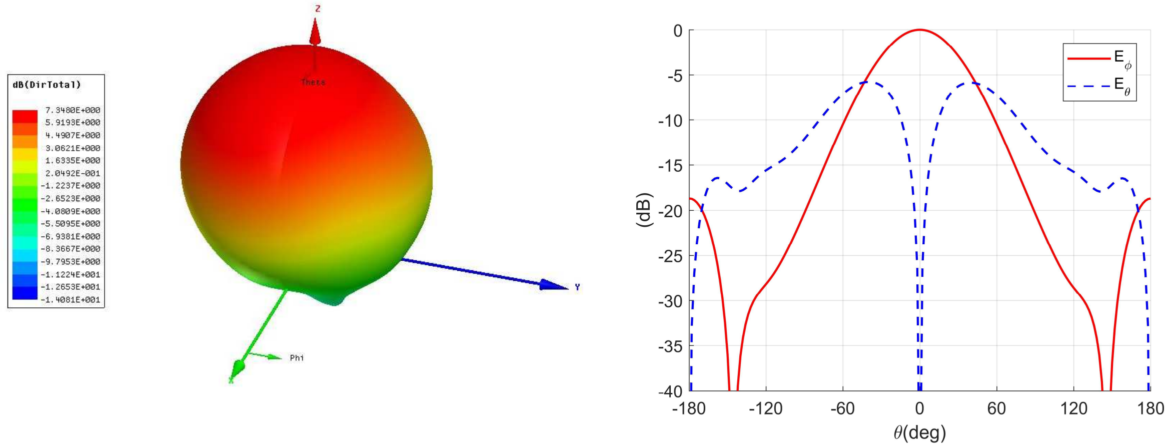


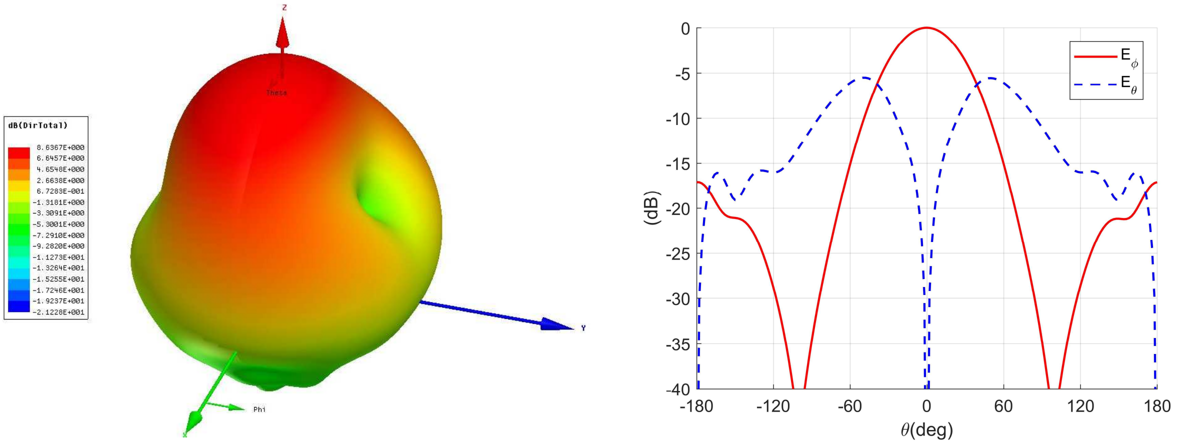
Figure 8-5: The optimized design after applying symmetry. The yellow squares are the patch pixels.



(a) 3-D radiation pattern

(b) Normalized co-pol and cross-pol far-field components (xz-plane)

Figure 8-6: Radiation characteristics of the optimized patch at $f = 1.9GHz$



(a) 3-D radiation pattern

(b) Normalized co-pol and cross-pol far-field components (xz-plane)

Figure 8-7: Radiation characteristics of the optimized patch at $f = 2.4GHz$

CHAPTER 9

9 Slotted Patch Antenna for Dual-Band Applications: Simulation And Measurement

9.1 Introduction

Chapter 8 covered the discussion on the utility of patch antennas, and how their low profile makes them a common candidate in various applications. Also, the concept of adding more bandwidth to a patch antenna is by introducing slots to the antenna in which a new resonant frequency is added has been demonstrated. The approach taken in that chapter is to convert the patch into discrete number of elements i.e. pixels, and different designs are realized by removing pixels from the patch. That approach required the use of BBSO to perform a binary optimization. In this chapter, the same problem will be tackled with the difference being is that the problem will have real valued variables that need to be optimized. This requires the use of BSO to optimize the real-valued variables. Additionally, the optimized design will be fabricated and measured to demonstrate how the optimization tool can be a part of the full design process.

The design concept is proposed by [78], where slots are introduced by cutting a half-U-slot and a rectangular slot on the edges of the patch as shown in Figure 9-1. The two slots add additional resonance frequencies to the patch. BSO will be used to design a dual-band antenna element for wireless communication (1.9 GHz and 2.4 GHz) applications. In this work, the High Frequency Spectrum Simulation (HFSS) is used to evaluate the structure [76], which uses the Finite Element Method (FEM) as a numerical solver.

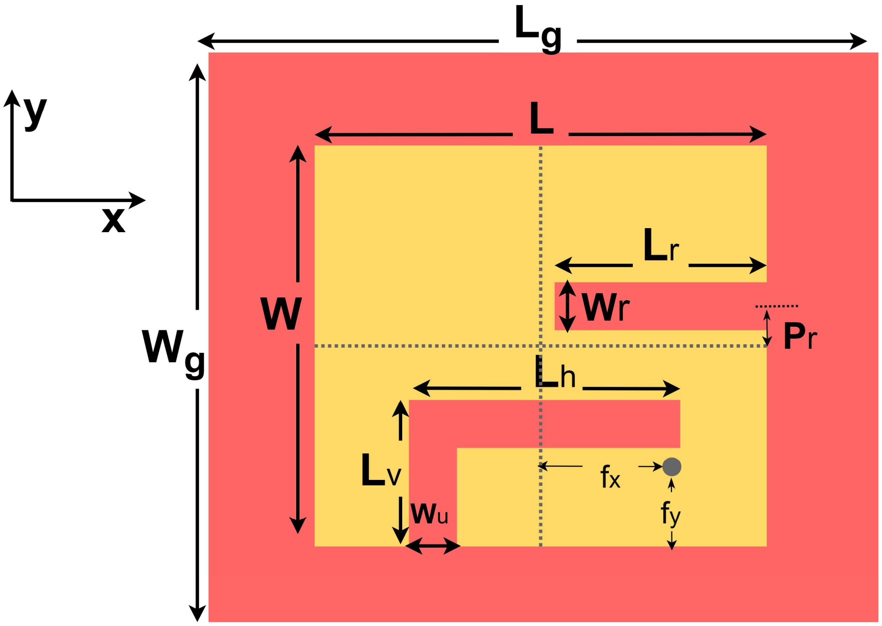


Figure 9-1: Coaxial probe-fed slotted rectangular patch.

9.2 Developing the Optimization Problem

The slotted patch is designed on RT/duroid 5880 substrate with thickness of $h = 3.175\,\text{mm}$ and $\varepsilon_r = 2.2$. As stated, the goal is to obtain a design that operates in the desired bands of interest (1.9 GHz and 2.4 GHz). Therefore, the fitness function for the dual-band antenna is written as

$$f = \max\left(S_{11}\,|_{f=1.9GHz}\,,\; S_{11}\,|_{f=2.4GHz}\right). \qquad (9.1)$$

BSO parameters used in this optimization are: population number is 20, number of clusters is 4, and. The optimization will run for 300 iterations. A single function evaluation can take from 7 to 20 minutes using HFSS. Termination criterion is added since time was a big factor in this design. Next, the optimization variables and their ranges are defined. There is a total of ten variables in this optimization that define the ten-dimensional search space. These variables are taken with reference to Figure 9-1. Two variables that define the outer geometry of the patch which are the length L and the width W and their ranges in (mm) are defined as:

$$W \in [30, 70], \tag{9.2}$$

$$L \in [30, 70], \tag{9.3}$$

rectangular slot variables and their ranges are defined as:

$$W_r \in [1, 10], \tag{9.4}$$

$$L_r \in [0, 50], \tag{9.5}$$

$$P_r \in [-10, 10], \tag{9.6}$$

P_r is the offset the center of the rectangular slot from the center along the y-axis, with positive values to indicate that the slot is offset above the center, and negative is below the center. The half-U slot variables and their ranges are defined as:

$$W_u \in [1, 10], \tag{9.7}$$

$$L_h \in [7.5, 70], \tag{9.8}$$

$$L_v \in [5, 35], \tag{9.9}$$

feed location variables and their ranges are defined as:

$$f_x \in [0, 30], \tag{9.10}$$

$$f_y \in [0, 10]. \tag{9.11}$$

The ground size will be relative to the size of the patch i.e.

$$W_g = W + 6h \tag{9.12}$$

$$L_g = L + 6h \tag{9.13}$$

Moreover, sets of constrains need to be enforced in this design such that any design would be meaningful in a physical sense. For example, not only does the feed need to be inside the patch geometry defined in Eq.(9.2) and Eq.(9.3), but also on the conductive part i.e. outside the slot region; these two constraints are written as equalities such that:

$$f_x < \frac{L}{2}, \tag{9.14}$$

$$f_y < L_v - W_u. \tag{9.15}$$

Additional constraints are enforced as shown below.

$$L_h < L, \tag{9.16}$$

$$L_r < L, \tag{9.17}$$

$$L < \frac{W}{2} - \frac{W_r}{2} + P_r. \tag{9.18}$$

The number of constraints in this optimization adds a layer of difficulty on top of the fact that this problem is a highly dimensional optimization problem. Both properties combined make this optimization problem more difficult.

9.3 Optimization Results

Figure 9-2 shows BSO convergence where the fitness function is the worst return loss of the two frequencies of interest. BSO was able to achieve a ($S_{11} = -25.5\text{ dB}$) within 187 iterations only, which led to the decision to stop the optimization as the obtained value is good enough to save time to start the fabrication process and then start the measurement. Table 9-1 shows the slotted patch antenna optimized dimensions (in mm) with reference to Figure 9-1.

Figure 9-3 shows the two sides of fabricated slotted patch antenna. Figure 9-4a shows the measurement setup of the fabricated antenna to obtain the return loss using Vector Network Analyzer (VNA). Figure 9-4b shows the measurement setup of the fabricated antenna inside a spherical near-field antenna chamber that uses automated software performing near-field to far-field transformation.

Table 9-1: Slotted Patch Antenna Optimized Dimensions (in mm)

W	L	W_r	L_r	P_r	W_u	L_h	L_v	f_x	f_y
65.96	60.45	6.45	6.82	-5.21	7.04	54.10	22.85	22.43	7.69

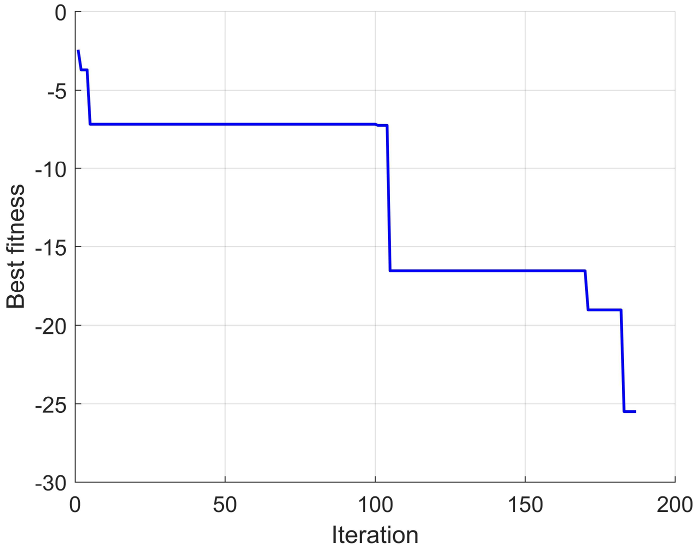
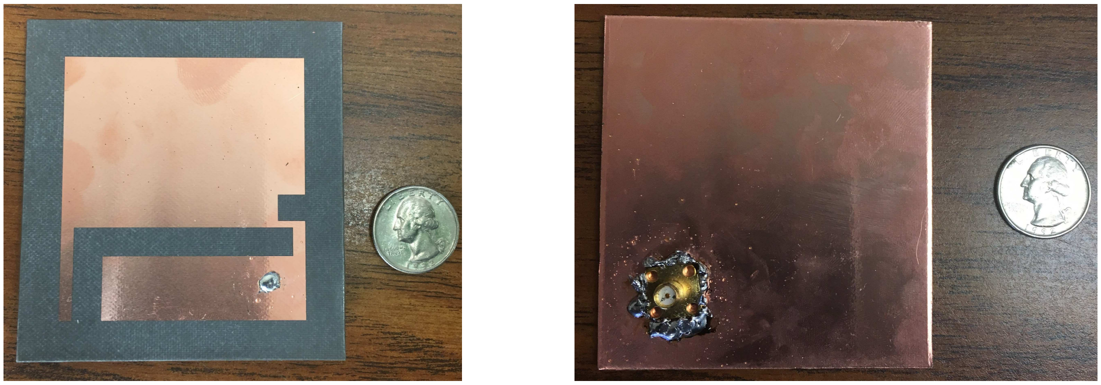


Figure 9-2: BSO Convergence

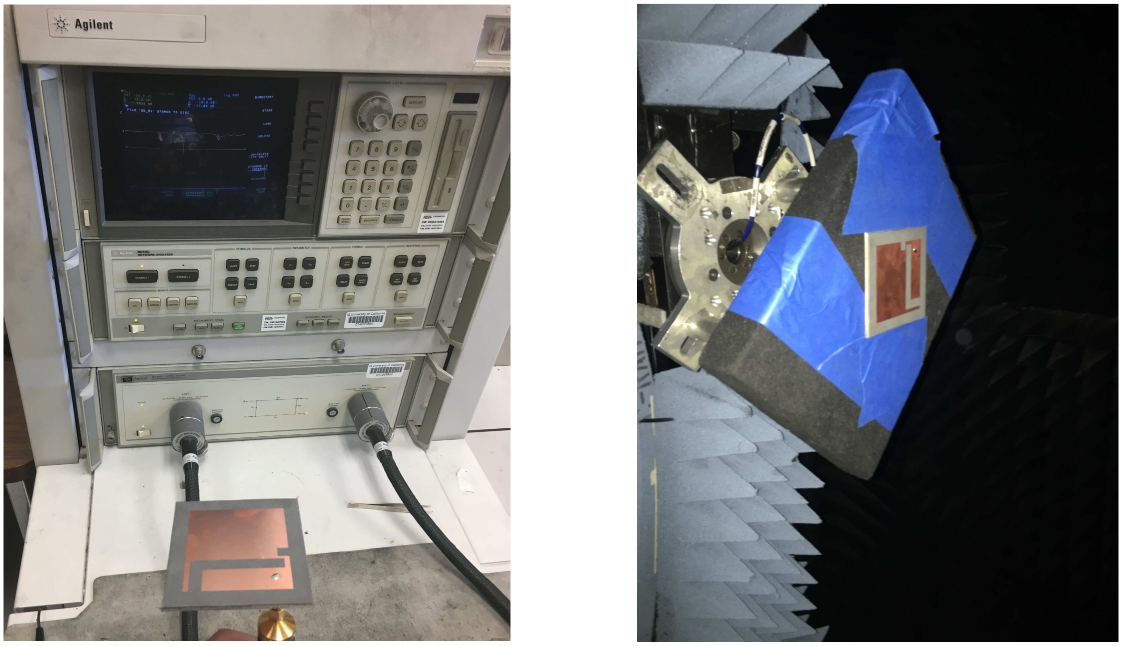
The return loss of the optimized design is shown in Figure 9-5, which shows a good agreement with simulated design. The measured directivity of the fabricated antenna at frequencies 1.9 GHz and 2.4 GHz are 5.8 dB and 7.8 dB, and their radiation characteristics are shown in Figure 9-6 and Figure 9-7, respectively.



(a) Top

(b) Bottom

Figure 9-3: Picture of the fabricated slotted patch antenna.



(a) S_{11} measurement

(b) Pattern measurement

Figure 9-4: Fabricated slotted patch antenna for port and radiation pattern measurements

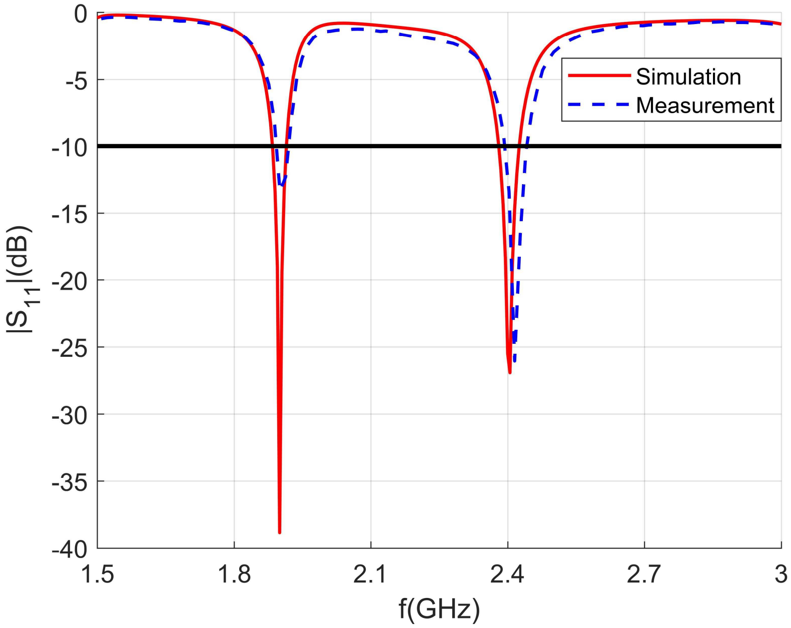
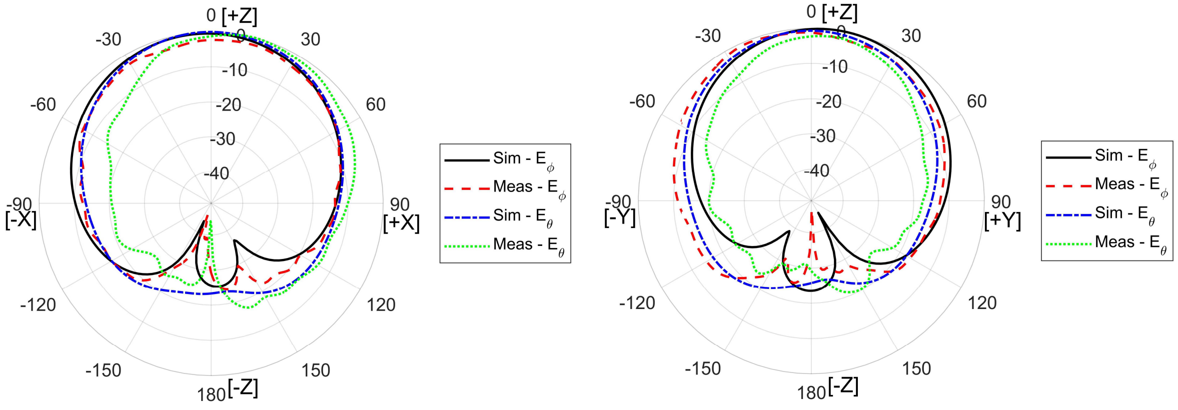
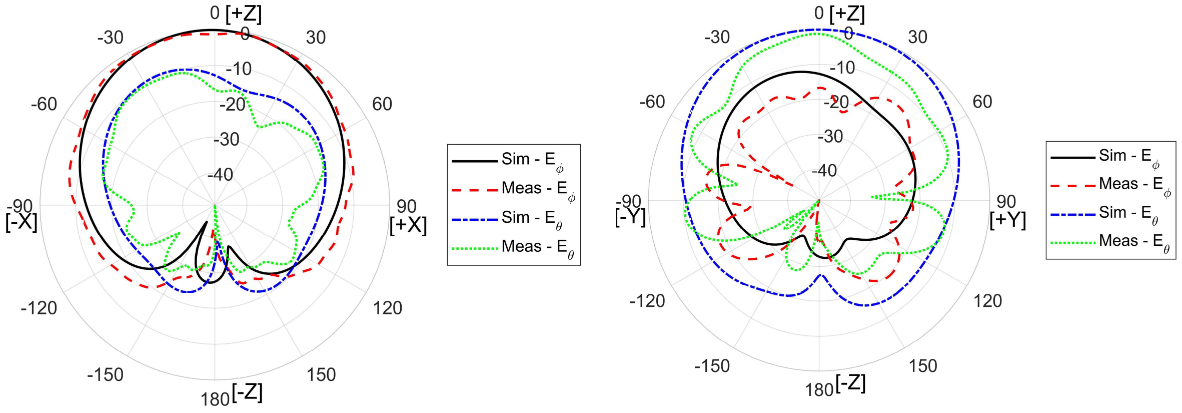


Figure 9-5: Input return loss of the optimized patch.



(a) *xz*-plane (b) *yz*-plane

Figure 9-6: Normalized co-pol and cross-pol far-field components in dB of the optimized patch at $f = 1.9GHz$



(a) *xz*-plane (b) *yz*-plane

Figure 9-7: Normalized co-pol and cross-pol far-field components in dB of the optimized patch at $f = 2.4GHz$

CHAPTER 10

10 Conclusions

Throughout this thesis, BSO has demonstrated great results to be a powerful optimization tool that can be used in microwave and antenna engineering. A detailed discussion on the concept of BSO was provided as well as how it is compared to PSO. Then, both algorithms were applied to common mathematical benchmark functions with different characteristics to test their convergence and robustness. Both algorithms had similar performance results and convergence speed. The goal was to test the algorithm using fewer number of function evaluations than the numbers used in the literature. This difference in the optimization environment explains why the overall results of the algorithm are not comparable to the ones published in the field of optimization.

Then, BSO and PSO were applied to two electromagnetic problems, which is one of the main objectives of this work. As already mentioned, to the best of our knowledge, this is the first time that BSO is applied to electromagnetic problems. First, a six-element Yagi-Uda antenna was optimized. The goal was to achieve higher forward directivity and front-to-back ratio. BSO has demonstrated its potential to be a promising tool for optimization based on its performance with respect to the well-known algorithm PSO. BSO has a similar convergence speed to PSO and was able to achieve better results in the Yagi-Uda example. The second problem was optimizing Luneburg Lens, where the objective was to the lower the sidelobes level while maximizing the boresight gain. Different configurations of the design were optimized to provide more insight into their performance. In one configurations of Luneburg Lens, BSO showed significantly better results than PSO; a case where PSO fell in a local trap during the early stage of optimization. Compared to the results provided by BSO, it is reasonable to say the PSO failed this optimization test. In other two cases, BSO was able to attain slightly better designs than PSO. And, in one case both BSO and PSO had similar performance. The four cases showed that BSO at worst is as good as PSO, and better than PSO at best. PSO showed to be prone to fall in a local optimum trap. Since there is no prior knowledge of the fineness function behavior in electromagnetic problems most of the time, BSO might be a better candidate than PSO as an optimization tool based on the results presented in the two examples.

The next main part of this work was the introduction of a novel binary version of BSO that can be used for problems where variables of the optimization have discrete representation. To the best of our knowledge this is the first attempt to transform BSO to a binary BSO. The BBSO parameters were discussed, and the reasons behind the choice of their values demonstrated. Then, BBSO was applied with success to a two-dimensional array thinning problem where the goal was to minimize the sidelobe. The test showed BBSO to be a potential candidate for binary optimization. After that, BBSO is applied to a pixel patch antenna, where the patch is divided into discrete number of elements. The goal was to achieve a dual-band patch antenna by removing those elements from the patch. This example showed the beauty of global optimization tools, where a design was created from a non-working design. Additionally, the example demonstrated the integration of BBSO with a common electromagnetic solver, HFSS.

Finally, we provided a demonstration of how BSO as an optimization tool can be used in the full design process. The process started with selecting a slotted patch antenna as a representative design example. BSO was used as a global optimization tool in conjunction with HFSS as a numerical electromagnetic solver and a simulation tool. Finally, the optimized design was fabricated and measured.

References

[1] T. Weise, Global Optimization Algorithms – Theory and Application –, Thomas Weise, 2009.

[2] S. Boyd and L. Vandenberghe, Convex Optimization, Cambridge: Cambridge University Press, 2004.

[3] D. M. Pozar, Microwave Engineering, 4th ed., Hoboken, NJ: John Wiley & sons, 2011.

[4] J. Robinson and Y. Rahmat-Samii, "Particle swarm optimization in electromagnetics," *IEEE Transactions on Antennas and Propagation,* vol. 52, no. 2, pp. 397 - 407, Feb. 2004.

[5] M. N. O. Sadiku, Numerical Techniques in Electromagnetics with MATLAB, 3rd ed., Boca Raton, FL: CRC Press, 2009.

[6] J. Volakis, A. Chatterjee and L. Kempel, Finite Element Method Electromagnetics: Antennas, Microwave Circuits, and Scattering Applications, Piscataway, NJ: IEEE-Wiley, 1998.

[7] P. P. Silvester and G. Pelosi, Finite Elements For Wave Electromagnetics, IEEE Press, 1994.

[8] K. Kunz and R. Luebbers, The Finite Difference Time Domain Method for Electromagnetics, Boca Raton, FL: CRC Press, 1993.

[9] W. Yu, R. Mittra, T. Su, Y. Liu and X. Yang, Parallel Finite-Difference Time Domain Method, Norwood, MA: Artech House Publishers, 2006.

[10] R. F. Harrington, Field Computation by Moment Methods, Wiley-IEEE Press, 1993.

[11] W. C. Gibson, The Method of Moments in Electromagnetics, Boca Raton, FL : Chapman & Hall/CRC, 2008.

[12] R. L. Haupt, "An Introduction to Genetic Algorithms for Electromagnetics," *IEEE Antennas and Pmpagatiin Magazine,* vol. 37, no. 2, pp. 7-15, April 1995.

[13] Y. Rahmat-Samii and E. Michielssen, Eds., Electromagnetic Optimization by Genetic Algorithms, New York: Wiley, 1999.

[14] B. Austin and W.-C. Liu, "An optimized shaped Yagi-Uda array using the genetic algorithm," in *IEE National Conference on Antennas and Propagation*, York, UK, 1999.

[15] R. Haupt, "Thinned arrays using genetic algorithms," *IEEE Transactions on Antennas and Propagation,* vol. 42, no. 7, pp. 993 - 999, 1994.

[16] H. Mosallaei and Y. Rahmat-Samii, "Nonuniform Luneburg and Two-Shell Lens Antennas: Radiation Characteristics and Design Optimization," *IEEE Transactions on Antennas and Propagation,* vol. 49, no. 1, pp. 60 - 69, Jan 2001.

[17] H. Mosallaei and Y. Rahmat-Samii, "RCS reduction of canonical targets using genetic algorithm synthesized RAM," *IEEE Transactions on Antennas and Propagation,* vol. 48, no. 10, pp. 1594 - 1606, Oct. 2000.

[18] N. Jin and Y. Rahmat-Samii, "Advances in Particle Swarm Optimization for Antenna Designs: Real-Number, Binary, Single-Objective and Multiobjective Implementations," *IEEE Transactions on Antennas and Propagation,* vol. 55, no. 3, pp. 556 - 567, 2007.

[19] Y. Rahmat-Samii, D. Gies and J. Robinson, "Particle swarm optimization (PSO): A novel paradigm for antenna designs," *URSI Radio Science Bulletin,* vol. 2003, no. 306, pp. 14 - 22, Sept. 2003.

[20] Y. Rahmat-Samii, "Modern Antenna Designs using Nature Inspired Optimization Techniques: Let Darwin and the bees help designing your Multi band MIMO antennas," in *2007 IEEE Radio and Wireless Symposium*, Long Beach, 2007.

[21] Y. Rahmat-Samii and C. G. Christodoulou, "Guest Editorial for the Special Issue on Synthesis and Optimization Techniques in Electromagnetics and Antenna System Design," *IEEE Transactions on Antennas and Propagation,* vol. 55, no. 3, pp. 518-522, March 2007.

[22] Y. Rahmat-Samii, J. M. Kovitz and H. Rajagopalan, "Nature-Inspired Optimization Techniques in Communication Antenna Designs," *Proceedings of the IEEE,* vol. 100, no. 7, pp. 2132-2144, July 2012.

[23] M. Behera, A. B. Sahoo, H. Pradhan and B. Mangaraj, "Performance comparison of PSO optimized mutually coupled linear array antenna with Yagi-Uda antenna," in *IEEE Conference on Information and Communication Technologies*, Thuckalay, Tamil Nadu, India, 2013.

[24] J. M. Kovitz, H. Rajagopalan and Y. Rahmat-Samii, "A novel optimized broadband reconfigurable RHCP/LHCP E-shaped patch antenna," in *2012 IEEE Antennas and Propagation Society International Symposium (APSURSI)*, Chicago, 2012.

[25] H. Rajagopalan, J. M. Kovitz and Y. Rahmat-Samii, "MEMS Reconfigurable Optimized E-Shaped Patch Antenna Design for Cognitive Radio," *IEEE Transactions on Antennas and Propagation,* vol. 62, no. 3, pp. 1056-1064, March 2013.

[26] L. Lizzi, F. Viani, R. Azaro and A. Massa, "Optimization of a Spline-Shaped UWB Antenna by PSO," *IEEE Antennas and Wireless Propagation Letters,* vol. 6, pp. 182-185, April 2007.

[27] S. Karimkashi and A. A. Kishk, "Invasive Weed Optimization and its Features in Electromagnetics," *IEEE Transactions on Antennas and Propagation,* vol. 58, no. 4, pp. 1269-1278, April 2010.

[28] S. H. Sedighy, A. R. Mallahzadeh, M. Soleimani and J. Rashed-Mohassel, "Optimization of Printed Yagi Antenna Using Invasive Weed Optimization (IWO)," *IEEE Antennas and Wireless Propagation Letters,* vol. 9, pp. 1275 - 1278, 2010.

[29] A. Dastranj, H. Abiri and A. Mallahzadeh, "Design of a Broadband Cosecant Squared Pattern Reflector Antenna Using IWO Algorithm," *IEEE Transactions on Antennas and Propagation,* vol. 61, no. 7, pp. 3895-3900, July 2013.

[30] L. Pappula and D. Ghosh, "Large array synthesis using Invasive Weed Optimization," in *2013 International Conference on Microwave and Photonics (ICMAP)*, Dhanbad, India, 2013.

[31] Y. Shi, "Brain Storm Optimization Algorithm," in *Advances in Swarm Intelligence.*, vol. 6728, Y. Tan, Y. Shi, Y. Chai and G. Wang, Eds., Springer, Berlin, Heidelberg, 2011, pp. 303-309.

[32] C. Sun, H. Duan and Y. Shi, "Optimal Satellite Formation Reconfiguration Based on Closed-Loop Brain Storm Optimization," *IEEE Computational Intelligence Magazine,* vol. 8, no. 4, pp. 39-51, November 2013.

[33] H. Duan, S. Li and Y. Shi, "Predator–Prey Brain Storm Optimization for DC Brushless Motor," *IEEE Transactions on Magnetics,* vol. 49, no. 10, pp. 5336-5340, October 2013.

[34] H. T. Jadhav, U. Sharma, J. Patel and R. Roy, "Brain storm optimization algorithm based economic dispatch considering wind power," in *2012 IEEE International Conference on Power and Energy (PECon)* , Kota Kinabalu, 2012.

[35] H. Zhu and Y. Shi, "Brain storm optimization algorithm for full area coverage of wireless sensor networks," in *Eighth International Conference on Advanced Computational Intelligence (ICACI)*, Chiang Mai, 2016.

[36] A. F. Osborn, Your Creative Power, New York: Charles Scribner's Sons, 1940.

[37] A. F. Osborn, Applied Imagination, 3rd ed., New York: Charles Scribner's Sons, 1979.

[38] Z.-h. Zhan, J. Zhang, Y.-h. Shi and H.-l. Liu, "A modified brain storm optimization," in *2012 IEEE Congress on Evolutionary Computation (CEC)*, Brisbane, QLD, 2012.

[39] S. Cheng, Y. Sun, J. Chen, Q. Qin, X. Chu, X. Lei and Y. Shi, "A comprehensive survey of brain storm optimization algorithms," in *2017 IEEE Congress on Evolutionary Computation (CEC)*, San Sebastian, 2017.

[40] S. Cheng, Y. Shi, Q. Qin and S. Gao, "Solution clustering analysis in brain storm optimization algorithm," in *2013 IEEE Symposium on Swarm Intelligence (SIS)*, Singapore, 2013.

[41] H. Zhu and Y. Shi, "Brain storm optimization algorithms with k-medians clustering algorithms," in *2015 Seventh International Conference on Advanced Computational Intelligence (ICACI)*, Wuyi, 2015.

[42] Z.-h. Zhan, W.-n. Chen, Y. Lin, Y.-j. Gong, Y.-l. Li and J. Zhang, "Parameter investigation in brain storm optimization," in *2013 IEEE Symposium on Swarm Intelligence (SIS)*, Singapore, 2013.

[43] J. Xue, Y. Wu, Y. Shi and S. Cheng, "Brain Storm Optimization Algorithm for Multi-objective Optimization Problems," in *Advances in Swarm Intelligence*, vol. 7331, Y. Tan, Y. Shi and Z. Ji, Eds., Springer, Berlin, Heidelberg, 2012, pp. 513-519.

[44] D. Zhou, Y. Shi and S. Cheng, "Brain Storm Optimization Algorithm with Modified Step-Size and Individual Generation," in *Advances in Swarm Intelligence*, vol. 7331, Y. Tan, Y. Shi and Z. Ji, Eds., Berlin, Springer, 2012, pp. 243-252.

[45] S. Cheng, Y. Shi, Q. Qin, T. O. Ting and R. Bai, "Maintaining population diversity in brain storm optimization algorithm," in *2014 IEEE Congress on Evolutionary Computation (CEC)*, Beijing, 2014.

[46] Z. Yang and Y. Shi, "Brain storm optimization with chaotic operation," in *2015 Seventh International Conference on Advanced Computational Intelligence (ICACI)*, Wuyi, 2015.

[47] M. El-Abd, "Brain storm optimization algorithm with re-initialized ideas and adaptive step size," in *2016 IEEE Congress on Evolutionary Computation (CEC)*, Vancouver, BC, 2016.

[48] Z. Cao, X. Hei, L. Wang, Y. Shi and X. Rong, "An Improved Brain Storm Optimization with Differential Evolution Strategy for Applications of ANNs," *Mathematical Problems in Engineering,* vol. 2015, pp. 1-18, 2015.

[49] Z. Cao, Y. Shi, X. Rong, B. Liu, Z. Du and B. Yang, "Random Grouping Brain Storm Optimization Algorithm with a New Dynamically Changing Step Size," in *ICSI 2015: Advances in Swarm and Computational Intelligence*, Cham, 2015.

[50] Z. Zhou, H. Duan and Y. Shi, "Convergence analysis of brain storm optimization algorithm," in *2016 IEEE Congress on Evolutionary Computation (CEC)*, Vancouver, BC, 2016.

[51] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *IEEE International Conference on Neural Networks, 1995. Proceedings.*, Perth, 1995.

[52] Y. Rahmat-Samii, "Genetic algorithm (GA) and particle swarm optimization (PSO) in engineering electromagnetics," in *17th International Conference on Applied Electromagnetics and Communications, 2003. ICECom 2003.*, Dubrovnik, Croatia, 2003.

[53] M. Jamil and X.-S. Yang, "A Literature Survey of Benchmark Functions For Global Optimization Problems," *Int. Journal of Mathematical Modelling and Numerical Optimisation,* vol. 4, no. 2, pp. 150-194, 2013.

[54] Y.-W. Shang and Y.-H. Qiu, "A Note on the Extended Rosenbrock Function," *Evolutionary Computation,* vol. 14, no. 1, pp. 119 - 126, 2006.

[55] S. Uda, "On the Wireless Beam of Short Electric Waves," *Journal of IEE of Japan,* pp. 273-282, 1926.

[56] H. Yagi, "Beam Transmission Of Ultra Short Waves," *Proc. IRE,* vol. 16, pp. 715-741, June 1928.

[57] D. Pozar, "Beam Transmission Of Ultra Short Waves: An Introduction To The Classic Paper By H. Yagi," *Proceedings of the IEEE,* vol. 85, no. 11, p. Proceedings of the IEEE, Novermber 1997.

[58] C. A. Balanis, Antenna Theory. Analysis and Design, 3rd ed., New York: Wiley, 2005, pp. 577-600.

[59] C. Chen and D. Cheng, "Optimum element lengths for Yagi-Uda arrays," *IEEE Transactions on Antennas and Propagation,* vol. 23, no. 1, pp. 8 - 15, Jan 1975.

[60] MathWorks™, "Antenna Toolbox™ User's Guide," 2017. [Online]. Available: https://www.mathworks.com/help/pdf_doc/antenna/antenna_ug.pdf. [Accessed August 2017].

[61] R. K. Luneburg, "Mathematical Theory of Optics," University of California Press, Berkeley, 1964.

[62] J. Sanford, "A Luneberg-lens update," *IEEE Antennas and Propagation Magazine,* vol. 37, no. 1, pp. 76-79, Feb. 1995.

[63] C. S. Silitonga, Y. Sugano, H. Sakura, M. Ohki and S. Kozaki, "Optimum variation of the Luneberg lens for electromagnetic scattering calculations," *International Journal of Electronics,* vol. 84, no. 6, pp. 625-633, 1998.

[64] J. R. Sanford, "Scattering by Spherically Stratified Microwave Lens Antennas," *IEEE Transactions on Antennas and Propagation,* vol. 42, no. 5, pp. 690 - 698, May 1994.

[65] S. Morgan, "Generalizations of spherically symmetric lenses," *IRE Transactions on Antennas and Propagation,* vol. 7, no. 4, pp. 342 - 345, October 1959.

[66] G. Peeler and H. Coleman, "Microwave stepped-index luneberg lenses," *IRE Transactions on Antennas and Propagation,* vol. 6, no. 2, pp. 202 - 207, 1958.

[67] J. Hollis and M. Long, "A Luneberg lens scanning system," *IRE Transactions on Antennas and Propagation,* vol. 5, no. 1, pp. 21 - 25, January 1957.

[68] K. W. Kim and Y. Rahmat-Samii, "Spherical Luneburg lens antennas: Engineering characterizations including air gap effects," in *IEEE Antennas and Propagation Society International Symposium*, Atlanta, GA, USA, 1998.

[69] A. Greenwood and J.-M. Jin, "A field picture of wave propagation in inhomogeneous dielectric lenses," *IEEE Antennas and Propagation Magazine,* vol. 41, no. 5, pp. 9 - 18, Oct 1999.

[70] C. T. Tai, "The electromagnetic theory of the spherical luneberg lens," *Applied Scientific Research, Section B,* vol. 7, no. 1, p. 113–130, Dec 1959.

[71] C. T. Tai, Dyadic Green Functions in Electromagnetic Theory, New York: IEEE Press, 1994.

[72] MathWorks™, "Statistics and Machine Learning Toolbox," [Online]. Available: https://www.mathworks.com/help/pdf_doc/stats/stats.pdf. [Accessed 16 January 2018].

[73] MathWorks™, "Phased Array System Toolbox™ User's Guide," 2017. [Online]. Available: https://www.mathworks.com/help/pdf_doc/phased/phased_ug.pdf. [Accessed 06 Feb. 2017].

[74] F. Yang, X. Zhang, X. Ye and Y. Rahmat-Samii, "Wide-band E-shaped patch antennas for wireless communications," *IEEE Transactions on Antennas and Propagation,* vol. 49, no. 7, pp. 1094 –1100,, July 2001.

[75] F. Villegas, T. Cwik, Y. Rahmat-Samii and M. Manteghi, "A parallel electromagnetic genetic-algorithm optimization (EGO) application for patch antenna design," *IEEE Transactions on Antennas and Propagation,* vol. 52, no. 9, pp. 2424 - 2435, Sept. 2004.

[76] Anosft, "Ansoft HFSS," 2016.

[77] S. Bhardwaj and Y. Rahmat-Samii, "Revisiting the generation of cross-polarization in rectangular patch antennas: A near-field approach," *IEEE Antennas and Propagation Magazine,* vol. 56, no. 1, pp. 14-38, Feb 2014.

[78] A. A. Deshmukh and K. P. Ray, "Compact Broadband Slotted Rectangular Microstrip Antenna," *IEEE Antennas and Wireless Propagation Letters,* vol. 8, pp. 1410 - 1413, 2009.

[79] AT&T, "http://about.att.com," AT&T, [Online]. Available: http://about.att.com/newsroom/music_in_grant_park.html. [Accessed 07 May 2018].

[80] Antennas-Amplifiers.com, "6meter 3 Elements Yagi Antenna PA50-3-1.5," Antennas-Amplifiers, [Online]. Available: https://www.antennas-amplifiers.com/50mhz-3-element-antenna-6-meter. [Accessed 07 May 2018].