# Butterfly Mating Optimization

*in fulfilment of the requirements*

*for the completion of Robotics Project*

*by*


K.Vineetha(N150727)

G.Naga Laxmi(N150390)

S.Pavani(N150288)

M.Nagamani (N150458)

*Department of Electronics and Communication Engineering*

*Rajiv Gandhi University of Knowledge Technologies*

*Nuzvid, India - 521202*

# Certificate

It is certified that the work contained in this term project entitled ’Butterfly Mating Optimization’ by ’K.Vineetha,G.Naga Laxmi,S.Pavani,M.Nagamani’ has been carried out under the supervision of Chakravarthi Jada and that it has not been submitted elsewhere for a degree.

Project Guide

Chakravarthi Jada

Asst.Professor

P. Vinod babu

Asst. Professor

RGUKT NUZVID

# *Abstract*

---

Name of the student:

**K.Vineetha**

**G.Naga Laxmi**

**S.Pavani**

**M.Nagamani**

Degree for which submitted: **B.Tech** Department: **ECE**

Term project title: **Butterfly Mating Optimization**

Project guide: **Chakravarthi Jada, Vinod babu**

Month and year of thesis submission: **June 2021**

---

Today the present world will running on new technologies.With the technology human work will also be reduced rapidly.One of them is Robotics.This paper presents a novel swarm intelligence algorithm named as Butterfly Mating Optimization (BMO) which is based on the mating phenomena occurring in butterflies. The BMO algorithm is developed with novel concept of dynamic local mate selection process which plays a major role in capturing multiple peaks for multi modal search spaces. This BMO algorithm was tested on 3-peaks function and various convergence plots were drawn from it. Also, BMO was tested on other benchmark functions to check and discuss thoroughly its capability in terms of capturing the local peaks.

**Key Words::** Butterflies ·Patrolling ·Mating ·UV distribution ·Multimodal Optimization ·Glowworms ·Local mate selection ·Bfly

# *Acknowledgements*

I would extend my sincerest gratitude... to my project guide who guides and give best instructions to do this project and make it successful.

# Contents

# List of Figures

# Chapter 1

# Introduction

Nature is inhabitant for enormous number of species. These species exhibit and follow different customs and strategies to sustain in their life. For instance, though they are tiny in size, ants do apply simple cooperative rules among themselves for foraging, building nests and other social activities in their daily lives. Similarly, fish schooling will do extraordinary maneuverings to avoid the predators. Not only the above said species but also many others such as birds for travelling longer distances, honey bees for collecting nectar, termites for building complex apartments etc. All these species perform these complex activities using simple and elegant rules. All these species perform these activities using their very limited body shape, sensorimotor parts, and communication systems. For instance, mosquitoes know the human presence by sensing heat released by human, using their antennas. Similarly, bats sense environment movements of the prey, obstacles etc using reflection of the ultrasonic phenomena.   Various research groups are inspired from the strategies of biological species and applied at daily technological aspects and succeeded to a large extent. Specifically, the present fast growing embedded technology is giving the boost. Recently developed and easy to use embedded processors such as Raspberry pi, beaglebone etc and sensors such as kinect depth sensor, hokuyo laser range finder, gas sensor, humidity sensor are some among those. Added to this the advancement in the miniaturization of processors are making the devices into more proximate to the reality in imitating the nature and biological species behavior. Some of the recent inventions includes painless needle from mosquitoes biting, morphing wing UAV, humanoids etc.

## 1.1 Related Works

People also have inspired from the swarm intelligence behavior of the species and designed various artificial metaphors such as Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO), Glowworm Swarm Optimization (GSO) etc to multimodal search spaces. And implemented these models into multi-robotic systems. For various applications and they have shown multiple ways of solving the problems. Recently, by inspiring from the bizarre flight of the butterflies, Our research group (2016) has proposed a swarm intelligence algorithm name as ”Butterlfly Mating Optimization (BMO)” which would simultaneously capture all the local optima of a multimodal search functions. This paper deals with the hardware architecture of the mobile robotic platform which is designed to the requirements of the BMO algorithm in view of using it to detect multiple signal sources in a 2-D environment and various individual sensing experimental results also presented and discussed various present and prospective works. Next part of this paper presents the brief survey on the major swarm algorithms of similar to BMO and the architectures of mobile robotic platforms designed to implement those algorithms.

# Chapter 2

# Butterfly Mating Optimization

## 2.1 Description of BMO Algorithm

While patrolling, butterflies mainly use UV reflectance (by males) and absorbance (by females) mechanism for mating. This proposed "Butterfly Mating Optimization" assumes that there is no differentiation among males and females and every butterfly reflects UV it has got and absorbs UV that it receives from all the remaining butterflies simultaneously. Hence this algorithm suggests a meta-butterfly model which replaces "Butterfly in the natural environment" with"Bfly in the search space". BMO aims at solving multi modal optimization problems.In this each Bfly chooses its mate anywhere in the search space adaptively in each iteration, we call this mate as a local mate i.e. l-mate. This adaptive selection process of l-mate plays a key role in the BMO algorithm. Our algorithm starts with a well random dispersion of Bflies on the search space. Then each Bfly updates its UV, distributes to and accessesUV from all the others, chooses l-mate and moves towards that. The algorithm's description is given below:

### 2.1.1 UV Updation Phase

In this phase, each B-fly UV is updated in proportion to its fitness function value at the present location of B-fly according to

$$UV_i = max\{0, b_1 * UV_i(t-1) + b_2 * f(t)\} \tag{2.1}$$

UV is updated at time index 't' to give more priority to the present fitness value compared to the past UV. To satisfy this choose the $b_1$ and $b_2$ constants such that $0 \leq b_1 \leq 1$ and $b_2 > 1$.

### 2.1.2 UV Distribution Phase

In this phase, each B-fly distributes its own *UV* to the remaining B-flies such that the nearest B-fly get more share compared to farthest one. The below approach is followed for this distribution. An $i^{th}$ B-fly having $UV_i$ reflects its *UV* value to the $j^{th}$ B-fly at a distance $d_{ij}$ which is given by

$$UV_{i \to j} = UV_i * \frac{d_{ij}^{-1}}{\sum_k d_{ik}^{-1}} \tag{2.2}$$

Where i=1,2,....,$N$ is number of B-flies; j=1,2,...,$N$ j=i; K=1,2,...,$N$ k/=i; $UV_{i \to j}$ is *UV* absorbed by $j^{th}$ B-fly from $i^{th}$ B-fly; $d_{ij}$ is euclidean distance between $i^{th}$ and $j^{th}$ B-fly; $d_{ik}$ is euclidean distance between $i^{th}$ and $k^{th}$ B-fly.

### 2.1.3 Local Mate(l-mate) selection Phase

The *l*-mate selection is done in the following manner. Initially an $i^{th}$ B-fly arranges itself all remaining B-flies in the descending order based on the *UV* values they have reflected toward it. Based on the descending order each Bfly compares its fitness value to the corresponding Bfly fitness. Now, which Bfly reflects more fitness towards it then it choose that Bfly as its local mate. Now, if every B-fly chooses one B-fly in its descending order as its *l*-mate and move towards it, it leads to some kind of localization and sensing of peaks. But to capture local peaks simultaneously an $i^{th}$ B-fly should also consider the *UV* of remaining B-flies and choose its l-mate which satisfies the below condition

$$UV(i^{th} Bfly) < UV(j^{th} Bfly) \tag{2.3}$$

Where i=1,2,...,$N$; j=1,2,...,$N-1$; j is the index of B-flies in the descending order of $i^{th}$ B-fly.

### 2.1.4 Movement Phase

Each B-fly move towards its mate as following.

$$x_i(t+1) = x_i(t) + B_s * \frac{x_{l-mate}(t) - x_i(t)}{||x_{l-mate}(t) - x_i(t)||} \tag{2.4}$$

where $B_s$ is B-fly step size; $x_i$(t) is position of $i^{th}$ B-fly at time t. Below is the pseudo code for the algorithm and Fig. 2, 3, 4 shows 3-D plot of three peaks function, emergence of the Bflies at various iterations and UV convergence.

## 2.2 Pseudo code of the BMO algorithm

```
While
{
    for each Bfly
    {
        UV Updation;
        UV Distribution;
    }
    for each Bfly
    {
        Select l-mate;
        Update position;
    }
}
```

# Chapter 3

# Hardware Implementation

## 3.1 Arduino Uno

The Arduino Uno is an open-source microcontroller board based on the Microchip ATmega328P microcontroller and developed by Arduino.cc.[2][3] The board is equipped with sets of digital and analog input/output (I/O) pins that may be interfaced to various expansion boards (shields) and other circuits.[1] The board has 14 digital I/O pins (six capable of PWM output), 6 analog I/O pins, and is programmable with the Arduino IDE (Integrated Development Environment), via a type B USB cable.[4] It can be powered by the USB cable or by an external 9-volt battery, though it accepts voltages between 7 and 20 volts. It is also similar to the Arduino Nano and Leonardo.[5][6] The hardware reference design is distributed under a Creative Commons Attribution Share-Alike 2.5 license and is available on the Arduino website. Layout and production files for some versions of the hardware are also available.

The word ”uno” means ”one” in Italian and was chosen to mark the initial release of Arduino Software.[1] The Uno board is the first in a series of USB-based Arduino boards;[3] it and version 1.0 of the Arduino IDE were the reference versions of Arduino, which have now evolved to newer releases.[4] The ATmega328 on the board comes preprogrammed with a bootloader that allows uploading new code to it without the use of an external hardware programmer.[3]

While the Uno communicates using the original STK500 protocol,[1] it differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it uses the Atmega16U2 (Atmega8U2 up to version R2) programmed as a USB-to-serial converter.[7]


FIGURE 3.1: Uno Board

## 3.2 Light Dependent Resistor(LDR)

A light dependent resistor works on the principle of photo conductivity. Photo conductivity is an optical phenomenon in which the materials conductivity is increased when light is absorbed by the material.LDR's are light dependent devices whose resistance is decreased when light falls on them and that is increased in the dark. When a light dependent resistor is kept in dark, its resistance is very high. This resistance is called as dark resistance.

Photocells or LDR's are non linear devices. There sensitivity varies with the wavelength of light incident on them. Some photocells might not at all response to a certain range of wavelengths. Based on the material used different cells have different spectral response curves.When light is incident on a photocell it usually takes about 8 to 12 ms for the change in resistance to take place, while it takes one or more seconds for the resistance to rise back again to its initial value after removal of light.

## 3.3 Motor Shield

The Motor Shield is a driver module for motors that allows you to use Arduino to control the working speed and direction of the motor. Based on the Dual Full-Bridge Drive Chip L298, it is able to drive two DC motors or a step motor.

## 3.4 Gyro Scope Sensor

Gyroscope sensor is a device that can measure and maintain the orientation and angular velocity of an object. These are more advanced than accelerometers. These can measure the tilt and lateral orientation of the object whereas accelerometer can only measure the linear motion.Measured in degrees per second, angular velocity is the change in the rotational angle of the object per unit of time.

Besides sensing the angular velocity, Gyroscope sensors can also measure the motion of the object. For more robust and accurate motion sensing, in consumer electronics Gyroscope sensors are combined with Accelerometer sensors.

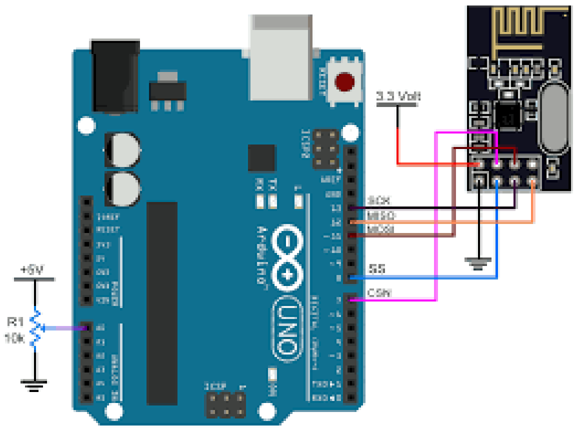
## 3.5 NRF Module

Having two or more Arduino boards be able to communicate with each other wirelessly over a distance opens lots of possibilities like remotely monitoring sensor data, controlling robots, home automation and the list goes on.

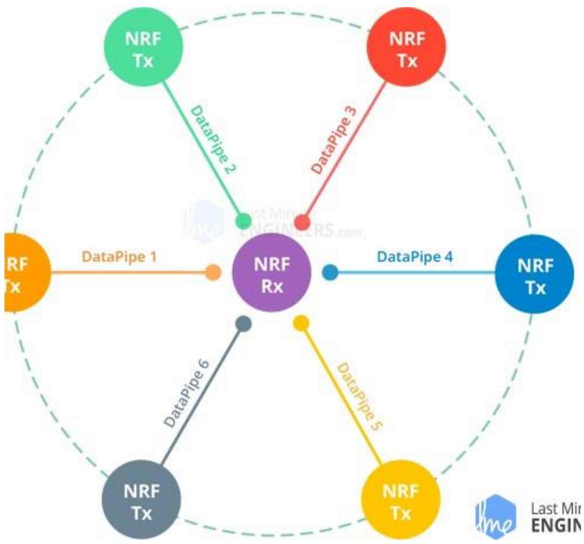
The nRF24L01+ provides a feature called Multiceiver. It's an abbreviation for Multiple Transmitters Single Receiver. In which each RF channel is logically divided into 6 parallel data channels called Data Pipes. In other words, a data pipe is a logical channel in the physical RF Channel. Each data pipe has its own physical address (Data Pipe Address) and can be configured. This can be illustrated as shown below.
Components

## 3.6 Components Interfacing(BOT making)

By integrating all the components above mentioned we get the bot




In-Depth How nRF24L01 Wireless Module Works Interface with Arduino.png In-Depth How nRF24L01 Wireless Module Works




Interface with Arduino.png

FIGURE 3.2: nrf module

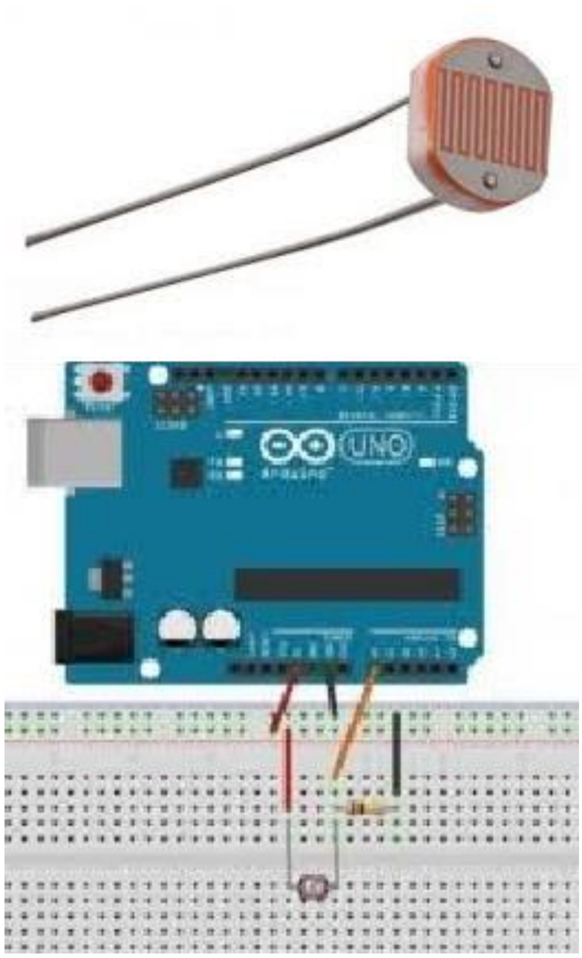


FIGURE 3.3: a. ldr b. ldr interfacing

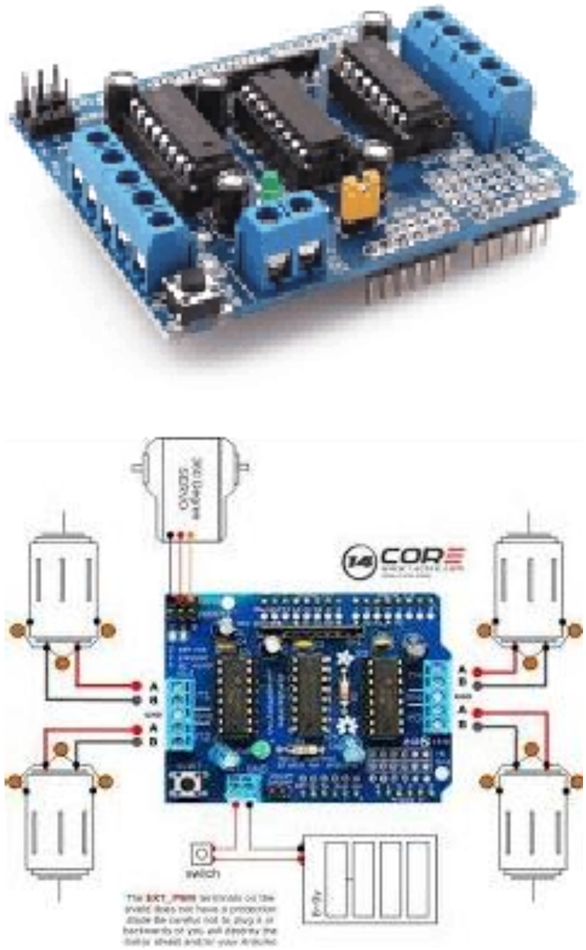



FIGURE 3.4: a. motor sheild b. motor sheild interfacing uno

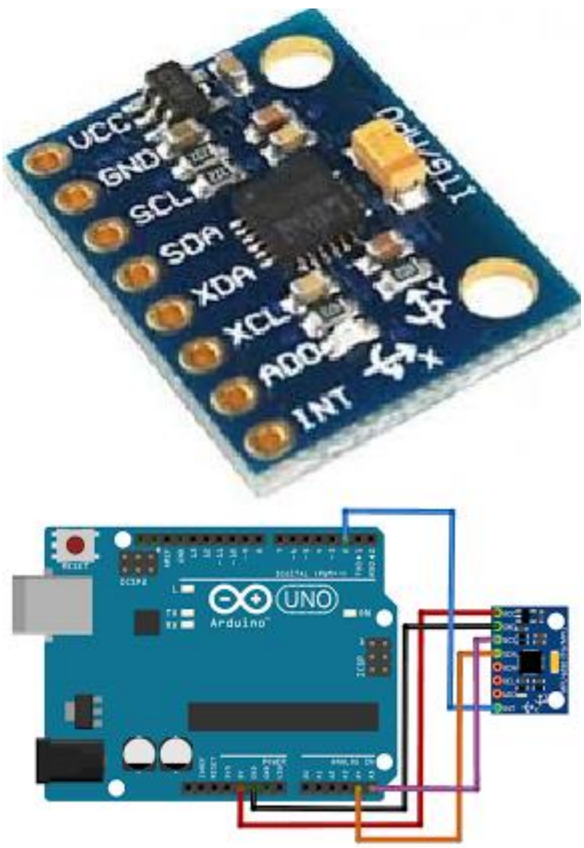



FIGURE 3.5: a. gyro b. gyro interfacing uno


FIGURE 3.6: ps2 mouse,battery,bot basic

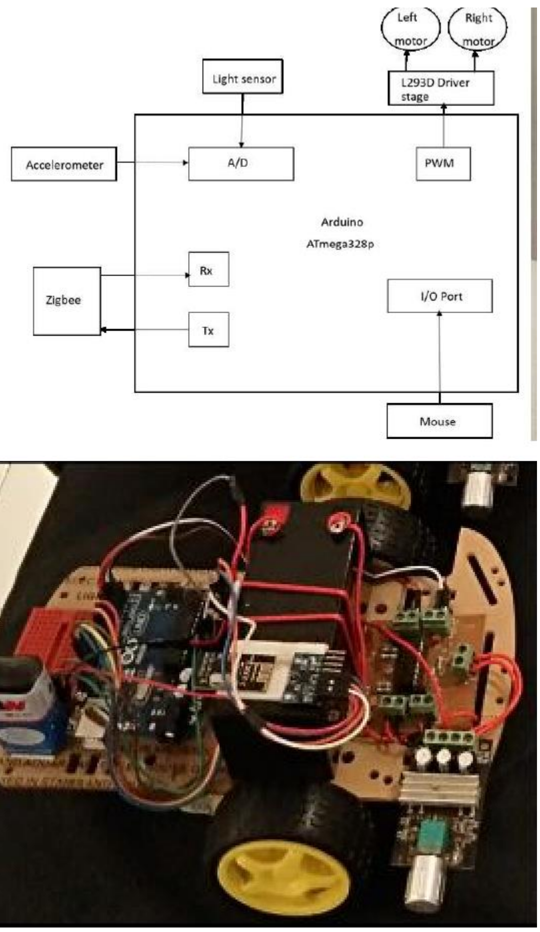



FIGURE 3.7: bot

# Chapter 4

# Results and Simulations

## 4.1 Software

Using the Matlab 2011b we have simulated the following results

### 4.1.1 3 peak Function

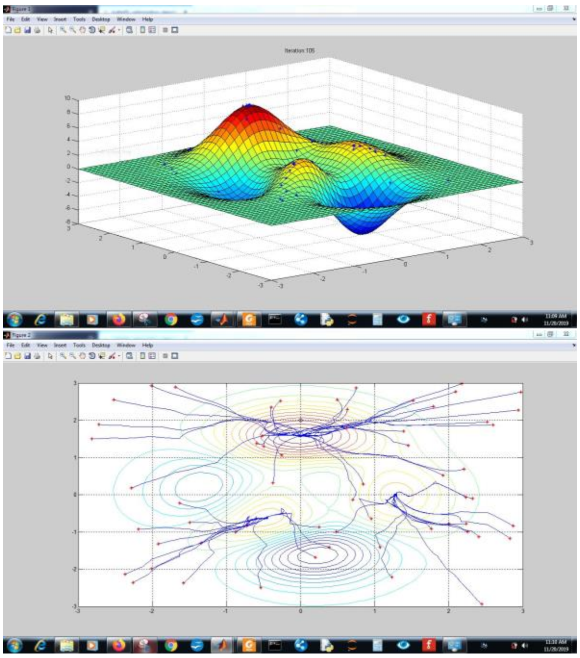


FIGURE 4.1: a. 3peaks function b. convergence

## 4.2 Rastrigin Function

In mathematical optimization, the Rastrigin function is a non-convex function used as a performance test problem for optimization algorithms. It is a typical example of non-linear multimodal function. It was first proposed by Rastrigin [1] as a 2-dimensional function and has been generalized by Mühlenbein et al.[2] Finding the minimum of this function is a fairly difficult problem due to its large search space and its large number of local minima. Parameters Required **number of butterflys=1000 stepsize=0.009 iterations=120 peaks=100**

Test functions for optimization - Wikipedia.png Test functions for optimization - Wikipedia.png

$$f(\mathbf{x}) = An + \sum_{i=1}^{n} \left[x_i^2 - A\cos(2\pi x_i)\right]$$

$$\text{where: } A = 10$$

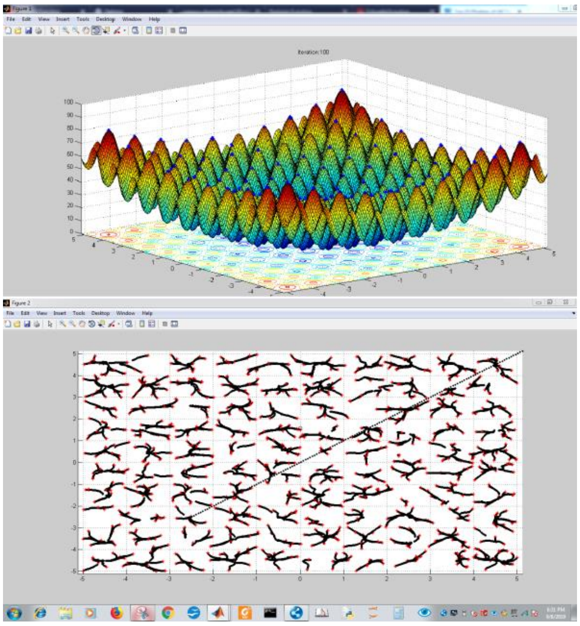
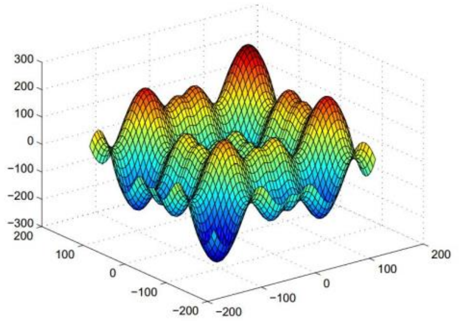


FIGURE 4.2: a. Rastrigin function b. Rastrigin convergence

## 4.3 Schwefel Function

We have simulated for following parameters **number of Butterflys=500 stepsize=0.5 iterations=400 peaks=16**



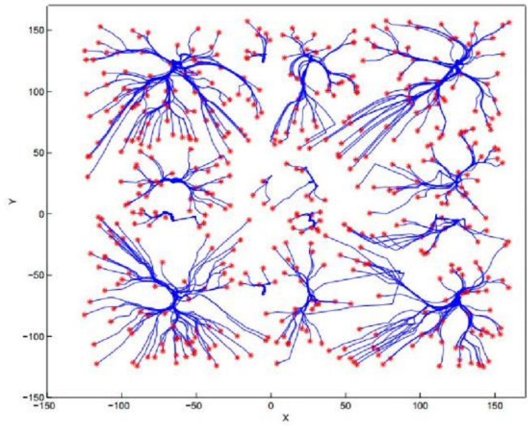
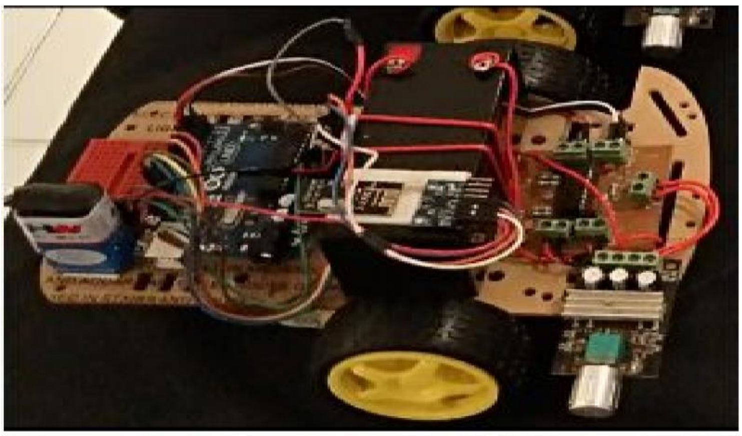
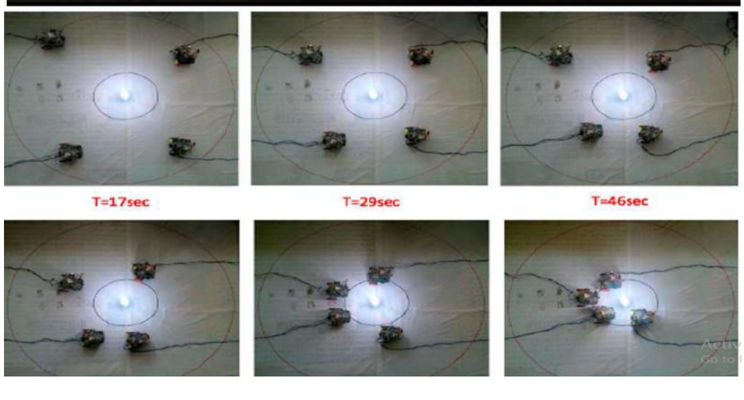




FIGURE 4.3: Schwefel convergence FIGURE 4.4: Schwefel convergence

## 4.4 Hardware





# Chapter 5

# Conclusions and Future works

## 5.1 Conclusion

The experimental results indicating the efficacy of the BMO in detecting the light source. Initial experiments are conducted with varying initial locations of the four bots, also validated the effect of step-size in the process of detection and found that smoothness and time of co-location are effecting. After, the number of bots are increased and conducted experiments to see the efficacy of the paradigm in detecting multiple light sources in both static and dynamic environments. Here it is observed that, the localization of Bflybots depend on the initial placements was observed. These trails on dynamic sources could lead to tsunami location identification. We also planned to take a step, to make represent the on-going and prospective works in those, planning to embedded the model into multi-quad rotor swarm, which could be the ultimate reality and suits to many practical applications. BMO can be applicable to both water bot and Earthquake prediction scenarios and some kind of object detection applications by considering the object as peak.

## 5.2 Future Works

BMO is an extremely simple algorithm that shows effective for the localization of multiple signal sources (same and different color). The convergence of Bflybot swarm is confined to a local region but the communication is done globally. By increasing the number of Bflybots, the probability of the convergence increases. Multi-source localization for various time intervals also discussed. In this paper,preparation of the workspace arena for the dynamically varying source detection is also presented. We can also extend this work to detect multiple dynamically varying sources.

## 5.3 REFERENCES


### 5.3.1 Simultaneous Localization of Signal Sources Inspired by Butterfly Paradigm By Chakravarthi Jada*, Vinod Pusuluri, Pavan Baswani, Ashok Urlana, Mahima Kumari Devineni, Padma Priya Motepalli, Ganesh Bodakurthi, Sumanth Motupalli

### 5.3.2 Butterfly Communication Strategies: A Prospect for Soft-Computing Techniques By Sowmya Ch, Anjumara Shaik, Chakravarthi Jada, Anil Kumar Vadathya

### 5.3.3 Butterfly Mating Optimization By Chakravarthi Jada, Anil Kumar Vadathya, Anjumara Shaik, Sowmya Charugundla, Parabhaker Reddy Ravula and Kranthi Kumar Rachavarapu

### 5.3.4 Butterflies : A new source of inspiration for futuristic aerial robotics By J Chakravarhi1, U Ashok2, B Pavan3

### 5.3.5 Butterfly Inspired Multi-robotic Swarm for Signal Source Localization by Chakravarthi Jada

### 5.3.6 www.google.com

### 5.3.7 www.wikipedia.com

#### 5.3.7.1 www.researchgate.com

## Appendix A

# Appendix A

Write your Appendix content here.

# Appendix B

# Appendix B

Write your Appendix content here.