MULTI-ROBOT LEARNING WITH BAT ALGORITHM WITH MUTATION (BAM)

CHANDRATHEVAN A/L SATHIAMURTHY
EA18182

BACHELOR OF ELECTRICAL ENGINEERING (ELECTRONICS) WITH HONOUR

UNIVERSITI MALAYSIA PAHANG

UNIVERSITI MALAYSIA PAHANG

DECLARATION OF THESIS AND COPYRIGHT

Author's Full Name : CHANDRATHEVAN A/L SATHIAMURTHY

Date of Birth : 21 MARCH 1997

Title : MULTI-ROBOT LEARNING WITH BAT ALGORITHM WITH MUTATION

Academic Session : 21/22

I declare that this thesis is classified as:

- ☐ CONFIDENTIAL (Contains confidential information under the Official Secret Act 1997)*
- ☐ RESTRICTED (Contains restricted information as specified by the organization where research was done)*
- ☑ OPEN ACCESS I agree that my thesis to be published as online open access (Full Text)

I acknowledge that Universiti Malaysia Pahang reserves the following rights:

1. The Thesis is the Property of Universiti Malaysia Pahang
2. The Library of Universiti Malaysia Pahang has the right to make copies of the thesis for the purpose of research only.
3. The Library has the right to make copies of the thesis for academic exchange.

Certified by:

(Student's Signature)	(Supervisor's Signature)
970321145997	DWI PEBRIANTI, PHD, ENG. TECH
New IC/Passport Number	Name of Supervisor
Date: 21 JANUARY 2022	Date: 13 FEBRUARY 2022

NOTE: * If the thesis is CONFIDENTIAL or RESTRICTED, please attach a thesis declaration letter.

THESIS DECLARATION LETTER (OPTIONAL)

Librarian,
Perpustakaan Universiti Malaysia Pahang,
Universiti Malaysia Pahang,
Lebuhraya Tun Razak,
26300, Gambang, Kuantan.

Dear Sir,

CLASSIFICATION OF THESIS AS RESTRICTED

Please be informed that the following thesis is classified as RESTRICTED for a period of three (3) years from the date of this letter. The reasons for this classification are as listed below.

Author's Name
Thesis Title

Reasons (i)

(ii)

(iii)

Thank you.

Yours faithfully,

____
(Supervisor's Signature)

Date: 13 FEBRUARY 2022

Stamp: DR. DWI PEBRIANTI, ENG. TECH.
SENIOR LECTURER
COLLEGE OF ENGINEERING
UNIVERSITI MALAYSIA PAHANG
LEBUHRAYA TUN RAZAK
26300 GAMBANG, KUANTAN, PAHANG
TEL : +609-549 2239 FAX : +09-549 2689

Note: This letter should be written by the supervisor, addressed to the Librarian, *Perpustakaan Universiti Malaysia Pahang* with its copy attached to the thesis.

SUPERVISOR'S DECLARATION

I hereby declare that I have checked this thesis and in my opinion, this thesis is adequate in terms of scope and quality for the award of the Bachelor of Electrical Engineering (Electronics) with Honours

(Supervisor's Signature)

Full Name : DWI PEBRIANTI, PHD, ENG. TECH

Position : SENIOR LECTURER

Date : 13 FEBRUARY 2022

(Co-supervisor's Signature)

Full Name :

Position :

Date :

STUDENT'S DECLARATION

I hereby declare that the work in this thesis is based on my original work except for quotations and citations which have been duly acknowledged. I also declare that it has not been previously or concurrently submitted for any other degree at Universiti Malaysia Pahang or any other institutions.

(Student's Signature)

Full Name : CHANDRATHEVAN A/L SATHIAMURTHY

ID Number : EA18182

Date : 1st January 2022

MULTI-ROBOT LEARNING WITH BAT ALGORITHM WITH MUTATION (BAM)

CHANDRATHEVAN A/L SATHIAMURTHY
EA18182

Thesis submitted in fulfillment of the requirements
for the award of the
Bachelor of Electrical Engineering (Electronics) with Honours

College of Engineering
UNIVERSITI MALAYSIA PAHANG

FEBRUARY 2022

ACKNOWLEDGEMENTS

First of all, I would like to thank my supervisor, Dr. Dwi Pebrianti for the guidance that she has given us throughout completing our project. I would not only say that her guidance has come to this level of success but her promptness in helping us out solving the confusions, encouragement and advice are the reasons which has led to a success in this project. With so many of learning materials, my supervisor has guided through by simplifying the understanding towards programming language used during our project. Other than that, we would like to convey our thanks to all of the staffs of Universiti Malaysia Pahang who played the role to complete up this project. Our huge thanks also should be reached to our Prof Mohd Ikhwan B Mohd Ridzuan and Prof Nur Huda binti Ramlan for guiding on how to complete at the best way in the PSM Presentation.

Next, I should convey my thanks to my family and friends, for their support especially for my friends from different faculties such as FTKEE and FKP, where they shared their experience, they have actually guided us by full patience in controlling my emotions while completing this project. Moreover, their suggestions were really helpful when they give the suggestions of which software is best to be used for development of robot model.

In addition, the postgraduate students from Universiti Malaysia Pahang should also deserve their gratitude for guiding me to keep things updated from time to time to ensure that I do not end up making huge mistakes in my project.

Finally, I would like to thank my faculty which is the Electrical and Electronics Engineering for not only providing me the experience but it has taught me how to expand my knowledge furthermore and implementing it in the future. These have really made me to get prepared for handling real projects when I enter into the real working environment by giving the advance feel of how it feels being an engineer in the future.

ABSTRAK

Robotik mudah alih adalah bidang penyelidikan yang aktif. Beberapa kaedah sedang dikaji untuk meningkatkan dan mengoptimumkan keupayaan kerja sistem pelbagai robotic. Sistem robotic atau kumpulan robot pelbagai ini mempunyai aplikasi yang luas dalam industry sebagai pembantu manusia untuk membawa barangan dan pelbagai pekerjaan yang boleh dilakukan. Pelbagai teknik seperti pengoptimuman kawanan, algoritma yang diilhamkan oleh kelawar biologi dilaksanakan untuk mencapai sasaran. Algoritma BAT dilaksanakan untuk mencapai sasaran. Algoritma BAT menggunakan teknik echolocation seperti kelawar untuk menjana populasi kelawar dan data rawak dihasilkan, robot kemudian melintasi dan jarak dikira yang dibandingkan dengan jarak dari halangan. Untuk lebih daripada satu robot, robot mempunyai statistik serta halangan dinamik. Oleh itu, kelajuan traversal dan kecekapan algoritma buruk mengurangkan sedikit.

ABSTRACT

The mobile robotics is an active area of research. Several methods are under study to increase and optimize the working capabilities of multi robotic systems. These multi robotic systems or robot swarms have vast applications in industry as a human assistant to carry goods and can-do variety of jobs. Multiple techniques like swarm optimization, cuckoo algorithm and other such algorithms are under study for multi robotic systems. In this research, a biological bat inspired algorithm is implemented to achieve the target. BAT algorithm is implemented to achieve the target. BAT algorithm uses echolocation technique like bats to generate bat population and random data is generated, the robot then traverses and the distance is calculated which is compared to the distance from the obstacle. The loop continues and robot keeps moving. For more than one robot, robots have statistic as well as dynamic obstacles. So, the traversal speed and efficiency of bad algorithm reduces slightly.

TABLE OF CONTENT

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

A	Loudness
P	Population size
g	g Generation
A_i	initial loudness produced
vi	initial velocity
xi	initial position
Γi	random number generated
f	frequency
X_{new}	New position
d	dimension
θ	theta(orientation)

LIST OF ABBREVIATIONS

Ai	Artificial intelligence
BA	Bat Algorithm
ACO	Ant Colony Optimization
GA	Genetic Algorithm
PSO	Particle Swarm Optimization
CS	Cuckoo-Search
VFF	velocity force field
ANFIS	Adaptive Network-based Fuzzy Interface System
NFL	No Free Launch
AMR	Autonomous Mobile Robot
MANET	Mobile As Hoc Network
GSO	Glow Worm Swarm Optimization
dPSO	distributed Particle Swarm Optimization
RPSO	Robotic Particle Swarm Optimization
FA	Firefly Algorithm
ABC	Artificial Bee Colony
EHO	Elephant Herding Optimization
KH	Krill Herd
DA	Dragon Fly Algorithm
GWO	Grey Wolf Optimizer
EWA	Earthworm Optimization Algorithm

CHAPTER 1

INTRODUCTION

1.1 Project Background

Swarm robotics refers to the utilisation of a large number of autonomous robots to complete a task. In a decentralized way, robot swarms coordinate the behaviours of a large number of very simple robots. In the development of collective artificial intelligence, swarm robotics is crucial (AI). Search and rescue, precision agriculture, supply chain management (SCM), and military surveillance are some of the current applications for robot swarms.

Swarm robots aims to mimic how social creatures, such as insects, use collaborative behaviours to complete complex tasks that are beyond the capabilities of any single individual. Swarm robotics researchers, for example, might look into how bees use pheromones to identify trails and record geographical areas. The researchers might then use the algorithms developed by the bee to mimic the same behaviour in robots.

Individual robots concentrate spatially in a certain section of the environment as a result of aggregation. Individuals in the swarm are able to get spatially near to one another, allowing for more interaction.

The swarm of robots is arranged in a specified shape using pattern formation. Chain formation is a specific instance in which robots build a line to establish multi-hop communication between two sites.

Self-assembly links the robots together to form structures. They can be physically or virtually connected via communication lines. Morphogenesis is a specific situation in which the swarm grows into a predetermined shape.

The swarm of robots can manipulate spatially distant objects using object clustering and assembly. For construction processes, item clustering and assembly are critical.

Swarm robotics is an important scientific tool in addition to being applicable to engineering applications. Using robot swarms, numerous models of natural swarm intelligence systems have been enhanced and evaluated. Robot swarms to evaluate a model of collective decision-making behaviour in cockroaches.

Swarm robots has also been used to examine the conditions under which some sophisticated social behaviours might emerge as a result of an evolutionary process, using controlled experiments. Robot swarms, for example, have been employed to investigate the evolution of communication and collective decision-making.

Research about designing even simple behaviours for robots that are effective and robust can be hard for human; it is regularly not difficult to execute a rudimentary controller that achieves the undertaking, however accomplishing ideal performance can be extremely challenging. Unsupervised robotic learning considers computerized plan of effective, robust controllers, which saves a lot of design time and effort. Unsupervised learning is additionally helpful for permitting robots to adapt to situations where the task/environment is obscure in advance or is continually evolving.

Modern optimization algorithms are frequently influenced by nature, and are frequently based on swarm intelligence. Because there are many various ways to get ideas, there are many different sorts of algorithms. However, for creating the key update formulae, all of these algorithms prefer to use some special characteristics. For example, genetic algorithms were inspired by Darwinian evolution characteristics of biological systems, and genetic operators such as crossover, mutation and selection of the fittest are used. Solutions in genetic algorithms are represented as chromosomes or binary/real strings. On the other hand, Particle Swarm Optimisation (PSO) was based on the swarming behaviour of birds and fish, and this multi-agent system may have emergent characteristics of swarm or group intelligence. Many variants of PSO and improvements exist in the literature, and many new metaheuristic algorithms have been developed.

As a result, the below given assumptions must be made:

- There is a communication mechanism among the robots;
- Due to the availability of different varieties of sensors, sensors can identify different variations of signals, every robot is updated and different types of sensors are integrated. The decision of which type of sensor to integrated is based on the different types of signals emitted by the obstacle and target.
- The target is responsible for emitting heat as well as light signals, that the robot can perceive.

1.2 Brief Introduction

Bat echolocation is the inspiration behind the bat algorithm (bat algorithm). Yang proposed the Bat Algorithm, which is built on the echolocation behaviour of flying mammal, bat. The only mammals that can fly are bats. Bats come in a variety of sizes and are found in many distinct kinds. Micro sized bats, for example, rely heavily on echolocation. They send out high frequency sound pulses and listen for echoes that ricochet off of nearby objects. Bats produce sound pulses at a constant, fixed, unvaried frequency of 25 to 150 kHz. Every burst of ultrasonic sound lasts just a few milliseconds to a few milliseconds. Micro sized bats also called, micro bats, produce 10–20 sound bursts each second on average. When they get near to their prey, the rate at which these sound pulses are emitted increases (to over 200 pulses per second). Bats create powerful sound pulses (in the range of 110 dB) when seeking for food, but as they move closer to their target, they grow quieter. Micro bat echolocation skills may be linked to the objective function to be improved, and optimization algorithms that replicate bat behavior can be developed to identify the best answer.

To address an optimization issue, the following assumptions are established to approximate the bat echolocation characteristics.

- All bats utilize echolocation to gauge distance, and in some mysterious manner, they "know" the difference between food/prey and backdrop obstacles.
- Bats seek for prey by flying at a random velocity v_i at location x_i with a constant frequency in the range f_{max} to f_{min}, variable wavelength as well as the loudness A_o.
- 3. Bats have the ability to automatically modify the wavelength (or frequency) of their produced pulses as well as the pulse emersion rate, r [0, 1], depending on the closeness of their target.
- 4. Their volume is considered to range from a high (positive) A_o to a low (constant) A_{min}.

This is a useful technology that effectively integrates the benefits of current optimization algorithms operating in a balanced way. The integration and utilization of the bat algorithm will result in lessening the troubles in addressing optimization issues that are coupled manually, the improvised Bat algorithm-based algorithms have been used to optimize energy-saving systems, solve constraint optimization tasks, optimize PID parameters, resolve the issue of antenna arrays, design effective data descriptions, and optimize artificial neural network models, among other things. The bat algorithm was utilized to obtain the best clustering information from training samples and to organize sports workouts. In addition, it has been utilized to drive a robot manipulator and in multi-level picture thresholding.

Researchers have extensively researched the Bat algorithm's broad application potential, and the working of Bat algorithm-based algorithms was successfully enhanced. A bunch of researchers has attempted to address the issue of early convergence. The discrete and parallel versions of the bat algorithm concentrate on certain sorts of issues, but the parameter less form of bat algorithm is more practical for handling real-world optimization problems. Many of the better bat algorithms were achieved by incorporating excessive theories, into the actual Bat Algorithm, for example, the cloud theory, differential evaluation, and chaotic sequences, into the original bat algorithm.

Particle swarm optimization (PSO) is related to the bat algorithm, and it has been used in multi robot systems (MRS). When nearing optimality, the bat algorithm has better services to provide such as, it has enhanced converging abilities which are not present in other algorithms, and the process of exploring as well as exploiting may be matched very well. Furthermore, the Bat algorithm is more capable of solving limited optimization issues. The bat algorithm is now being utilized to optimize solutions based on an accurate model. However, in real applications, obtaining an exact mathematical model might be problematic. These issues are common in multi robot systems, and the implementation of the bat algorithm needs further study. Multiple robotic systems are attracting more attention, and their use is expanding in multiple applications such as in military-related as well as non-military related applications such as an unmanned aerial vehicle, unique environmental rescues peacekeeping, space exploration, formation, counter-terrorism, and other areas that need to be operated in group forms. Even though the process of targeted identification with the help of mobile multiple robot systems is a hot issue, researching the subject is still restricted. Initially, the robot does not have any knowledge of the placements of targets as well as barriers the target searching is being done in new surroundings, but it does know the number of targets and the search area bounds. Mobile robots must employ sensors to perceive and gather environmental information due to not enough previous knowledge, for instance, target searching by environmental models. As a result, the below given assumptions must be made:

- There is a communication mechanism among the robots;
- Due to the availability of different verities of sensors, sensors can identify different variations of signals, every robot is updated and different types of sensors are integrated. The decision of which type of sensor to integrate is based on the different types of signals emitted by the obstacle and target.
- The target is responsible for emitting heat as well as light signals, that the robot can perceive.

The Bat algorithm is a unique methodology in multi robot target searching research, and a Bat algorithm-based method may be utilized for target finding in several unfamiliar settings. Moreover, bat algorithm-based algorithms offer numerous benefits over other metaheuristic algorithms, including frequency tuning, quick solution speed, high accuracy, low parameter change, and global convergence. Furthermore, such algorithms allow for more flexible exploitation and exploration balance. However, there are a number of obstacles to overcome.

- In case, when the algorithm's pulse emission, as well as loudness parameters, are not properly adequately synchronized, the algorithm's convergence will be either too fast or too sluggish. The conventional bat algorithm has a poor convergence speed and accuracy in the latter stages, and it is easy to slip into not well-matured convergence as well as local optima, limiting the applicability of the bat algorithm in a severe manner.

- Because a single bat algorithm does not have a proper mechanism for mutation, maintaining variety in the process of searching as well as removing the limitations of a local extreme value, is challenging. Because of its limited global exploration capacity, the optimization outcomes achieved by a bat algorithm are unstable, and its search, which of local nature impact is weak at the upcoming stages or parts.

The adaptive robotic bat algorithm (AR bat algorithm) is presented in this study for multi robot target hunting in unexpected conditions. According to the authors, this is the first research to employ a bat algorithm-based multi robot cooperative approach for target hunting. The adaptive robotic bat algorithm is a multiple robotic intelligent systems compatible intelligent optimization approach that has been presented. All robots are considered bats in the proposed system, and one robot corresponds to one bat. The adaptive robotic bat algorithm is a networked technique that enables robots to interact and move like bats. Adaptive robotic bat algorithm achieves the robots' regulating mechanism, in which robots acquire ambient information via sensors, make new judgments based on updated knowledge, and draw new courses to fulfil the job of seeking objectives. The proposed solution takes into account the obstacle avoidance issue. The three key contributions made by the adaptive robotic bat algorithm are described below.

- The adaptive inertial weight method helps the adaptive robotic bat algorithm increase its diversity while also avoiding local optima.
- The very important effect that is, Doppler Effect is part of the suggested approach for improving the frequency formula; the Doppler Effect is adaptively corrected as the robot advances and helps robots avoid premature convergence.

- In the adaptive robotic bat algorithm, the multi swarm technique is utilized to improvise variation, widen the finding region of the robot, as well as speed up the search process.

1.3 Problem Statement

Path planning is an old problem in robotics, just like other robotics problems. Obstacle avoidance is the primary focus of research, with the goal of determining the most efficient path for the robot to travel from the source to the destination. There are many algorithms for this; some simply stop the robot at a safe distance from the obstruction to avoid a collision, while others allow the robot to move. This method of detecting the obstacle and the robot ensures that the robot's entire dimension does not collide with any of the obstacles and that it advances smoothly towards the objective location, going along the obstacle until it is in its path, and then resuming its travel towards the target. In the case of autonomous path planning, the robot assumes an environment in which known or static obstacles as well as unknown or moving obstacles are present. In particular types of environments, this aids in good path planning.

1.4 Objectives

- Using the Bat Algorithm with Mutation, enable robot teams to build cooperative behaviours and techniques independently (BAM).

- Develop a system on a robot that can execute waypoint navigation while taking into account local information, with a focus on allowing each robot to contain and manage a single BAM particle.

- To create an initial waypoint path via known barriers as well as a reactive planner that reacts to pop-up obstacles.

1.5 Project Scope

- This is a robot control system for obstacle avoidance that can be used in multi-robot learning.

- Develop a system on a robot that can execute waypoint navigation while taking into account local information, with a focus on allowing each robot to contain and manage a single BAM particle.

- To create an initial waypoint path via known barriers as well as reactive planner that reacts to pop-up obstacles.

1.6 Thesis Outline

The thesis starts with introduction of the problem statement and the scope of the project summarizing the features of the end result. Then in next section, we have historical background, biological foundations and literature review of the project. Third section is the methodology that, in depth elaborates the implementation of algorithm. Then comes the result of the simulation and finally conclusion with future recommendations.

1.7 BAT Algorithm

A metaheuristic algorithm is a bat algorithm. Although each pulse of a bat's echolocation lasts just a very few seconds or precisely, a few thousand seconds, its frequency is virtually not changing, that is, constant, generally between 25 and 150 kHz. Approximate or Idealized set of rules, such as all of the bats can discriminate among the food, can also be called as prey, and barriers, they utilize the technique of echolocation to assess the displacement or distance. Moreover, they can automatically alter the rate of frequency and pulse emission rate, which should be incorporated in the standard bat algorithm. Robots are becoming more and more prevalent in people's everyday lives.

CHAPTER 2

LITERATURE REVIEW

2.1 Path Planning in Mobile Robotics

Mobile robots are used for a wide range of tasks, from basic lawn mowing to military applications. In the world of automated industrialization, the utilization of mobile robots is increasing every single day. The popularity of research is increasing every day as a result of different factors. The major goal of the mobile robot is to discover the best route to the location of the target while avoiding any hurdles and obstacles coming in its way. Route planning for mobile robots begins with observation of the surroundings, followed by map development and path planning in accordance with the required environment. After the successful planning of path by the robot, next step is to avoid the obstacle. Efficient path planning was taken to another level by the factor of obstacle avoidance. It involves utilization of different algorithms to achieve the task and each algorithm exhibits different pattern, technique, or method through which the robot avoids and surpasses the obstacles. This characteristic or property helped a lot in various fields involving robot utilization. Now, further advancement is being made in this field by making the robot trained enough to avoid all the obstacles of different types, such as, static as well as mobile. Previously, a mathematical model was built to determine the best approach, but the results were disappointing. Following that, several Soft computing methods were created, but the main disadvantage is that they take longer to identify the best route. There are traditional procedures such as the Voronoi diagram, potential field method, and cell decomposition methodology, which will then hit into local minima. Soft computing methods comprise networks of neurons, fuzzy logic, as well as others. The metaheuristic algorithms Ant Colony Optimization written as ACO, Genetic algorithm written as GA, Particle Swarm Optimization written as PSO, Bat algorithm written as bat algorithm, and the algorithm Cuckoo-search written as CS, have been created to alleviate the shortcomings of conventional techniques, which take longer to compute the ideal route. Because of its simplicity, the potential field approach has attracted greater attention. Characterization of the potential field method is as follows:

"The robot exerts attractive forces while the barriers produce repulsive effects." The PFM's fundamental flaw is that they only happened in local minimum conditions. In light of these flaws, he created the velocity force field (VFF), whose primary goal is to avert impediments in a real-time ambiance. However, there are several traditional approaches such as the cell decomposition method, and the Voronoi diagram. The Voronoi method involves the formation of the diagram. The diagram is built using the area of separation between two points, and the resultant path obtained by the Voronoi method is undoubtedly not the shortest method. In the second method, that is, the cell decomposition method, the spaces are fractioned into multiple grids. The accuracy and the precision of this procedure are determined by the size of the grids; if the grids are smaller, the accuracy of the grid is greater. The two ways of route planning are local path planning as well as global path planning. It is called global route planning if all knowledge about the environment is available, and it is known as local path planning when no previous information is available. In the process involving the utilization of local route planning, the robot may determine the course by utilizing external embedded or connected devices such as infrared sensors, cameras, and other similar equipment to handle these issues. For route planning, fuzzy logic has been used. Various fuzzy logic approaches have been created. The accuracy of fuzzy logic route planning is determined by the rules provided by humans. The most significant network for robots that are involving mobility route deciding is the network made up of neurons. The process or technique of neural networks found its basis in the idea of harnessing neurons in the human brain to solve problems. A group of researchers presented a multi-layered network of the neuron for planning the path in which the actual changing and dynamic environment were taken into account. The inputs of the network of the neuron are the front hurdle, right hurdle, and left hurdle, and the output is the angle of steering. Objects are detected and path planning is performed using sensors. However, accuracy is solely dependent on the state of training. A group of researchers then published a comprehensive review study on neural networks and mobile robot route planning applications, concluding that the feed forward of the back propagation will provide more precision than any other neural network. A genetic-based algorithm has been proposed. Some academics have mathematically proved that the resultant route is the most efficient route by assuming that chromosomes contain variable length binary strings; nonetheless, the basic drawback of genetic algorithms is that they take up more computer resources. Researchers suggested a route planning strategy that found its basis from PSO also known as swarm or particle swarm optimization, in which the function of the fitness was generated by keeping in view the area between the robot and the barrier. The cuckoo search was established by a group of researchers that took into account the target function of each nest value in order to determine the ideal route by removing blockages. He was offered the optimal route, which he considered was better than both algorithms are PSO which is particle swarm optimization, and genetic algorithm which is GA. The cuckoo search is the best algorithm. A group of researchers developed route planning for the bat algorithm using static obstacles of varied forms and sizes. (Fister, I. 2013)

2.2 Algorithms

The proposed technique assesses a fitness function based on robot and obstacle distances using a mix of particle swarm optimization (PSO) as well as genetic algorithms (GA). Hybrid technology, rather than relying on a single algorithm, will deliver superior results. To tackle the issue of local minima, a group of researchers devised a hybrid solution that combines the potential field approach with the distance transform method. Some studies noticed route planning utilizing a neuro fuzzy system, with the ANFIS system being more accurate and resilient than individual fuzzy and neural networks, according to one study. The above-mentioned options take longer to perform the algorithm and cover a bigger search area. With the aforementioned concerns in mind, an unusual hybrid cuckoo search and bat algorithm was created and tested on numerous case studies. Genetic Algorithm (GA), Ant Colony Optimization (ACO), Particle Swarm Optimization (PSO), Bat Algorithm (bat algorithm), CSCuckoo Search Algorithm (CSCuckoo), Potential Field Method (PFM), Velocity Force Field (VFF), and Adaptive Network-based Fuzzy Inference System (ANFIS) (ANFIS). For mobile robot route planning, researchers developed a hybrid cuckoo search bat approach. In this study, the hybrid cuckoo bat technique was applied to select a suitable route for the mobile robot. The cuckoo search algorithm is based on the cuckoo's parasitic nature: cuckoos deposit their eggs in other people's nests, and the host birds lay the eggs. On the other hand, the Bat algorithm is based on bat echolocation. Using their echolocation, the bats will estimate the distance between the obstacles, and they will mysteriously know the difference between the food as well as the background or the surrounding obstructions.

Bats are sometimes referred to as winged mice (in UK, they are sometimes referred to as blind mice), despite the fact that they are not birds or rodents. These noteworthy creatures exemplify a natural ecosystem that survives through decentralized decision making and synchronized locomotion. They are often colonial in nature. The largest global metropolitan bat colony is located beneath Austin's Ann W. Richards Congress Avenue Bridge. These bats are migratory, passing through Austin in the warmer months and Mexican in the cold season. Surprisingly, throughout the warmer months, Austin has more bats than people. (Fister, I. 2013)

2.3 Biological Background

Even though not all bat species are blind, they rely heavily on echolocation to navigate to a ground. Furthermore, this technique assists them in locating prey and distinguishing amongst different species of insects. Indeed, bats aren't the first species that employ echolocation or acoustic signal technique; a variety of other extraordinary creatures do as well. For instance:

• Dolphins: In darker seas, they rely on sound generation and reception for navigation, communication, hunting, and protection against predators.

• Toothed whales have acquired an extraordinary sensory capacity that they employ to locate food and navigate underwater. They generate sound by air movement across their head's air gaps or nostrils.

• Shrew: The animal, sometimes known as the shrew mouse, is a tiny mammal. They employ echolocation technique only for the purpose of studying their environments, not for the purpose of locating food. (Yang, 2010 #34)

• Oilbirds: Like bats, they navigate by echolocation, but with a high-pitched clicking sound at 2kHz that is perceptible to people. Additionally, some non-sighted individuals learn to navigate their environments through detection and tracking via sound waves. Before introducing the bat algorithm, a brief introduction of bat biology and behavior is offered. Following that, the fundamental bat algorithm is presented in detail. Following that, a full study of the works that have addressed the bat algorithm is offered. Following this examination, the subsequent query is posed: 'With far too many accessible algorithms and strategies, which of those has been most appropriate to addressing my research problem'? To aid bats in locating and hunting prey in the dark, the majority of species have evolved a specialized navigation mechanism called echolocation. As if we were standing at the entrance of a cave and yelling a single word, echolocation might be imagined. And second later, we shall hear our own words reverberating back to us. That is, sound is created when air is rushed from our lungs. Some of these vibrations induce oscillations in the flowing air, which result in the generation of sound waves. Pure sine wave having the below mentioned fundamental characteristics: amplitude, sound speed, frequency, sound speed, wavelength, wavenumber, sound intensity, sound pressure, and direction are usually represented as sound waves having acoustic energy. A sound wave is simply a flowing sequence of changes in pressure of air. The changing pressure gradient of air pushes out adjacent air particles and afterwards pulls them quickly in. Such molecules subsequently create tension on neighboring molecules, transferring the energy and rhythm of the sound wave. This enables sound to travel over great distances via the air. The term "echolocation" was coined by biologist Donald Griffin, who studied how bats travelled. Additionally, his work aided in the study and evolution of techniques like sonar and radar, when a great deal of research was conducted by monitoring bats, along with their lifestyle, hunting, and echolocation, among other things. In studies, some of the more well-known works on bats and their infrasound are discussed. The initial algorithm for bats. Xin-She Yang invented the bat algorithm (BA) in 2010. These pieces were inspired by microbats and their remarkable echolocation. (Villalón, 2020), (Yoshoika, 2008). The author intended to emulate their natural behavior and succeeded in developing a robust method that could be used to practically any field of optimization. Algorithm illustrates BA's pseudo-code. The behavior of bats is represented by the fitness function associated with the task to be addressed. The original BA algorithm described in Algorithm 9 operates as follows. Originally developed to handle combinatorial optimization issues, the bat optimization algorithm or technique is a bio inspired swarm-based intelligence method. The technique is reliant on how bats use echolocation to find food. Microbats, in fact, employ echolocation, a sort of sonar or radar that uses variable pulse rates of transmission and amplitude of sound signal to identify prey, avoid collisions with obstacles, and find their nesting niches in the night. The very same approach is followed in robotics for path finding, obstacle avoidance and optimization, which is why the term bats should now be understood as referring to microbats except as otherwise mentioned. (Ahmadlou, 2019). The following is a summary of the idealization of microbat echo detection and tracking:

- Bats utilize acoustic signals to detect distance and differentiate amongst foodstuff, predators, and environmental obstacles.
- To find prey, each digital bat flies at a stochastic velocity vi at point (location) xi with a constant frequency fmin, fluctuating wavelength, and amplitude Ao. According on the closeness of the object, it modifies the wavelength (or frequency as both are dependent) of its generated pulses and adjusts the frequency of pulse emission r as it seeks for and discovers its victim.
- It is anticipated that the volume would fluctuate from Ao (which is originally high and positive value) to Amin (which is originally small and negative).Several assumptions or hypothesis are highly recommended in order to use the bat algorithm to optimize problems in a timely manner. (Sharma, I. 2017)**.**

Suppose that the resonance frequency develops across a limited range [f_{min} , f_{max}] in general. Because f and lambda are tied to each other through the fact that the product of lambda and f is a constant quantity, the wavelength is constrained as well. It is also useful to choose the greatest wave length that is similar to the length of the field of view for pragmatic purposes (the search space, for optimization problems). It can be supposed that $f_{min} = 0$ for the sake of simplification, hence f belongs to [0, f_{max}]. The pulse rate might merely be in the range r from [0, 1], with 0 denoting no strokes and 1 being the highest rate of pulse emission. (Roy, 2013), (Yao, 2019)

2.4 Pseudocode

Bat algorithm pseudocode:

Initial Parameters are required such as:

Population size: P

Loudness: A

Max Frequency: fmax

Objective function

Max no. of generations: Gmax

Pulse rate: r

Problem dimensions: d

Random number: Θ from [0 , 1]

After that:

1. g:=0
2. Initialize the population of bat as xi and vi where i= 1,…,N
3. Define the frequency of the pulses fi at position xi
4. Initialize the pulse rates loudness Ai and ri
5. While g < Gmax do
    a. For i = 1 to P do
        i. Find new solutions by adjustment of fi
        ii. Update vi and xi
        iii. If Θ > ri
            1. Current location is taken as best
            2. Local solution around current location is found
        iv. End if
        v. Generate another solution randomly
        vi. If Θ < Ai then
            1. Take new value as solution
            2. Increase the value of ri and decrease Ai likewise
        vii. End if
    b. End for
    c. g := g + 1
6. end while
7. Re number the bats and get updated best xi

8. Return xi

Essentially, the algorithm takes into account an original subset of the population (bats). Every bat does have a position xi and a velocity vi, indicating a possible answer to the optimization issue. Within the search process, the algorithm initialises these parameters with random numerical values.

The pulse pattern, pulse rate, and amplitude for each particular bat are then calculated. The swarm then develops in a distinct manner across generations or iterations, similar to time instances, until it reaches its maximum span of generations, max. The accompanying evolutionary formulae are used to calculate new frequency, position, and velocities for each generation g and each individual bat:

$$fi^g = fmin^g + \beta(fmax^g - fmin^g) \quad 2.1$$

$$vi^g = vi^{g-1} + \beta(xi^{g-1} - x^*)fi^g \quad 2.2$$

$$xi^g = xi^{g-1} + vi^g \quad 2.3$$

$$xnew = xold + \in A^g \quad 2.4$$

For simplification, the numbers Ao = 1 and Amin = 0 are often used, with the latter value implying that somehow a bat has discovered the target and has momentarily stopped generating any sound. The following are the development equations for volume and pulse rate:

$$Ai^{g+1} = \alpha Ai^g \quad 2.5$$

$$ri^{g+1} = ri^0 \, [1 - exp\,(-\gamma g)] \quad 2.6$$

In this equation, both α and γ are taken as constants. For any 0 <α< 1 and any γ > 0 we have

$$Ai^g \; approaching \; to \; zero \; and \; ri^g \; approacing \; to \; ri^0 \; as \; g \; approaching \; to \; \infty.$$

This algorithm is composed of the following components:

• Initialization and configuration steps (step 1-3): initialising the constants or hyper parameters of the algorithm, forming an initial population of microbats, evaluating it, and ultimately identifying the population's best possible solution xbest.

• Construct a new value (step 6) by repositioning the microbats in the nearby space in accordance with the physical laws governing echolocation of the bats.

• Local search stage (steps 7-9): utilising the Random Walking Direct Exploitation (RWDE) experiential, improve the best solution.

• assess the new solution (step 10):

• Conditionally save the optimum solution (steps 12-14): analogous to simulated annealing, this saves the new optimal answer under certain likelihood Ai.

• discover the best possible solution (step 15): determining the best option available at the time. Interestingly, those elements are stated throughout the methodology as either functions offered by the system when invoked in a single step or as a comment inside two quotation marks in the final step when the part covers multiple lines. This bat colony is seeded arbitrarily. The newly obtained solutions are generated in accordance with the equations.

2.5 Applications

Nowadays, BA and its variations are used in practice to solve a variety of optimization and classification issues, as well as a variety of engineering difficulties. BA has been used to the following kinds of optimization problems, as illustrated in this figure: continuous, constrained, and multi-objective. Additionally, it is utilized to solve problems of classification in clustering, feature selection and neural networks. Finally, BAs are employed in a wide variety of engineering disciplines.

The original BA was used to evaluate a variety of benchmark functions. The author compared his unique approach to the genetic algorithm and particle swarm optimization in this work. The findings of this investigation were interesting, since the author discovered several observations that may be resolved in the future:

• increase the pace of convergence

• apply BA to problems with bigger dimensions. In one investigation, the authors enhanced the convergence rate by combining the original bat algorithm with variant evolution techniques termed as (HBA). However, BA in terms of high dimensions remained a difficult problem. Using differential evolution techniques and a random forests machine learning method, the same scientists hybridized the BA algorithm in this work (HBARF). Yilmaz and Kucuksille presented three improvements to the improved bat algorithm (IBA) for continuous optimization:

• Inertia weight factor modification: By adding a linear decreasing inertia weight factor, the exploitation potential of BA was enhanced.

• Adaptive frequency modification: Each dimension of a solution was allocated a distinct frequency.

• Scout bee modification: To compensate for BA's lack of exploration, it was hybridized using an artificial bee colony algorithm.

Their IBA outperformed the original BA. Wang and Guo combined a BA with a harmony search (HS/BA) to create a hybrid. The enhancements included the addition of a pitch adjustment operation in harmony search that acts as a mutation operator throughout the bat update process, with the goal of increasing convergence and making the technique more applicable to a broader range of real-world applications. This technique was validated using 14 industry-standard benchmark functions. Experiments demonstrated that their strategy outperformed SGA, GA, ACO, BA, DE, BBO, ES, HS, and PSO. Additionally, they investigated several HS/BA factors during their investigation. In conclusion, the HS/BA technique intends to enhance the advantages of BA and HS algorithms in hopes of avoiding all bats population being stuck in sub-optimal regions of local optima;

• the HS/BA technique helps the microbats to understand from a broader variety of paragons, even as microbats are updated at every recursion, and also forms new chords while trying to search within the same relatively large solution space;

• this new technique can expedite the optimal convergence without compromising the ruggedness of the bat algorithm.

To address the sluggish convergence rate and low accuracy of BA, Xie et al. suggested a novel BA using a differential operator and L'evy flights. The authors increased the algorithm's convergence speed by employing a differential operator. On the other hand, the L'evy flight trajectory can protect the population's variety against premature convergence and keep the algorithm out of local minima. Experiments were carried out on 14 common benchmark functions as well as on a few nonlinear equations. The findings demonstrated that the suggested approach was both practical and successful, as well as possessing improved approximation skills in high-dimensional spaces. Researchers used the traditional BA in conjunction with the Doppler effect (DEBA). This new algorithm proved to be quite efficient. The authors said that future enhancements will focus on algorithm convergence and correct parameter selections.

2.6 Optimization with Several Objectives

Multi-objective optimization is a technique for optimising a problem using more than one criterion. Numerous real-world optimization problems do not estimate a solution based on a single criterion, but on a combination of criteria. Single objective optimization seeks a single solution, whereas multi-objective optimization seeks a collection of optimum options, colloquially referred to as the Pareto front. During multi-objective optimization, two objectives are pursued:

• to identify solutions that are as near to the Pareto front as feasible

• to identify solutions that are as diverse as possible inside the resulting non-dominated front. Numerous methods for addressing multi-objective problems have been developed in the past. AbYSS, IBEA, DEMO, GDE3, FastPGA, MOCHC, MOEA/D-DE, NSGA-II, PAES, SPEA2, and VEGA are a few of the more well-known algorithms. Additionally, there is a specific Java framework called jMetal for multi-objective optimization. Several multi-objective algorithms are implemented in this framework, as are numerous multi-objective benchmark problems. Chinese researchers developed a BA for multi-objective optimization in 2011. (MOBA). Due of its simplicity, it employs a weighted sum to consolidate all objectives into a single goal.

Numerous novel Bayesian algorithms have been devised and used to a variety of problem fields. The BA method has demonstrated success in numerical optimization, particularly on tiny dimensions. Similarly, this holds true for engineering optimizations and real-world situations. Multiple examinations will be required in the future to determine the theoretical foundations of BA. On the other hand, there are several novel hybridization techniques available to producers of new BAs. Allow us to highlight just the most critical activities that must be completed in the near future:

• a broader theoretical foundation for the BA algorithm; • fusion with additional meta-heuristics and machine learning approaches (e.g., highly randomised forests, decision trees, gradient boosting),

• use Bayesian analysis for combinatorial optimization (e.g., graph coloring, b-coloring),

• the BA is inefficient for optimising problems with several dimensions;

• improved control over exploration and exploitation;

• adaptation and self-adaptation of control parameters;

• dealing with subpopulations;

• implementing BA in real-world enterprises.

Indeed, it is predicted that BA algorithms' global search capabilities would increase, and so premature convergence should be avoided. Additionally, machine-learning techniques have developed into a very interesting area of research, both in SI and EC. Several of these techniques have been applied to the BA algorithm in its original form. Finally, randomization techniques are critical components of every stochastic algorithm. To demonstrate how each randomization approach affects the performance of the original BA algorithms, a comparative analysis of several randomization methods was conducted (e.g., uniform and Gaussian distributions, L'evy flight, chaos maps, and random sampling in turbulent fractal clouds).

Numerous real-world issues have been solved using the bat method. Its performance, however, is still constrained by the 'No Free Launch' (NFL) theorem. It claims that when two algorithms' performance is compared across all potential issues, they are comparable. Fortunately, when problem-specific knowledge is included into some algorithms, their performance can be enhanced. Typically, this information is included via heuristic operators, initialization methods, and fitness functions, among other things. By combining problem-specific heuristics and the BA algorithm, a hybrid algorithm is created. Indeed, by definition, the bat algorithm is hybridised. That is, this algorithm expressly includes a phase for doing a local search. In a manner similar to simulated annealing, population diversity is achieved by randomly creating new solutions. Additionally, this phase is a possibility for hybridization. Additionally, by relocating the virtual bats and measuring the fitness function, problem-specific knowledge may be added. Additionally, the initial bat population can be formed by the incorporation of solutions from existing algorithms or through the use of heuristics, local search, and so on. This thesis's practical work concentrated on hybridizations and expansions of the original bat method. In accordance with this, two hybridizations have been produced. Both relate to the bat algorithm's local search stage. While the former adjusts the solution using the 'DE/rand/1/bin' method, the later employs random forest regression. Additionally, tests with bat algorithms employing other distributions for generating random numbers were undertaken to demonstrate how the outcomes of the original bat algorithm are affected by these distributions.

2.7 Mobile Robots

A mobile robot is one that can move around in its environment (locomotion). Robotics and information engineering are commonly regarded to be subfields of mobile robotics.

A spying robot is an example of a mobile robot that can move around in its surroundings.

Mobile robots can roam around their environment and are not restricted to a single physical area. Mobile robots can be "autonomous" (AMR - autonomous mobile robot), meaning they can navigate an uncontrolled environment without the use of physical or electromechanical guidance systems. Mobile robots, on the other hand, can rely on guiding systems to travel a pre-defined navigation route in a somewhat controlled region. Industrial robots, on the other hand, are usually more or less fixed, consisting of a attached to a fixed surface is a jointed arm (multi-linked manipulator) and gripper assembly (or end effector). A linear actuator, servo motor, or stepper motor controls the joint-arm.

In commercial and industrial environments, mobile robots have become more widespread. For many years, hospitals have used autonomous mobile robots to carry supplies. Mobile robotic devices have been installed in warehouses to carry products efficiently from stocking shelves to order fulfilment zones. Mobile robots are also a major focus of contemporary research, with one or more labs dedicated to mobile robot research at almost every major institution. [6] Mobile robots can be found in a variety of settings, including industrial, military, and security.

A mobile robot's components include a controller, sensors, actuators, and a power system. A microprocessor, embedded microcontroller, or personal computer is commonly used as the controller (PC). The sensors utilised are determined by the robot's requirements. Dead reckoning, tactile and proximity sensing, triangulation ranging, collision avoidance, position localization, and other specific applications could be among the requirements. The motors that move the robot, which might be wheeled or legs, are known as actuators. Instead of using an AC power supply, we normally utilise a DC power supply (which is a battery) to power a mobile robot.

2.8 Importance of swarm robotics

2.8.1 Decision making

These characteristics enable a swarm of robots to make a shared decision on a given issue.

- Consensus allows the swarm's individual robots to agree on or converge on a single common choice from a set of options.

- Task allocation dynamically assigns emergent jobs to the swarm's individual robots. Its purpose is to improve the overall performance of the swarm system. If the robots have different capacities, the work can be distributed differently to boost the system's performance even more.

- Individual robot shortcomings are determined by collective fault detection within the swarm of robots. It allows for the detection of robots that differ from the swarm's expected behaviour, such as due to hardware issues.

- Collective perception mixes data sensed locally by the swarm's robots.

2.9 Navigation in swarm robot

Navigation system that is loosely based on MANET routing algorithms. The swarm's robots build a MANET among themselves using wireless communication. The fundamental idea is to acquire navigation information in this MANET through communication and use it to direct a seeking robot from hop to hop to its target, just how routing information is obtained in a MANET and used to send data packets to their destination. The swarm's robots keep a table with navigation information for all known target robots. An estimate of the navigation distance to T is included in the information on a target T, as well as an indication of the information's relative age. Each robot is rotated on a regular basis transmits its own content table to its neighbours, who update their tables based on the information they've got. Navigation information is disseminated across the swarm via wireless communication in this manner. Robots also use dometry information to update the distance calculations in their table depending on their own motions.

2.10 Proposed Solution

There are several possibilities for robot search applications. Different niches were utilized to locate various outdoor sources; the robot was treated as a particle in their research; nevertheless, the scenario where the robot only had one sensor was studied. In an unfamiliar area, robots were employed to identify and cover the landscape, but the technology is not yet ready for robots having sensor errors and uncertainties and many other common flaws. The job allocation issue of a multi robot in the retrieval area and study was investigated, with the necessity to balance fuel consumption and completion time reduction. Certain restrictions must be taken into account by the target seeking algorithm.

- An appropriate method is required by the search algorithm to prevent long-term knowledge exchange amongst various robots.

- Decentralized rather than centralized control is preferable.
- The computation of the search method should be straightforward.

In research on robot target seeking, the most common swarm intelligence algorithms, for example, glow worm swarm optimization abbreviated as GSO and PSO-based algorithms have been used. The communication range of GSO affects its performance. GSO generally has no benefit over other algorithms when it comes to robotic target hunting. The PSO algorithm is a smart strategy that has been extensively utilized in target seeking related to robotics. Moreover, it can solve the issue type related to search satisfactorily. To find targets with tiny mobile robots, a distributed PSO (dPSO) was developed. The robots are shown as particles in this program. The primary frame of dPSO permits gbest particles to be transported in a single scenario, namely when any of the robots show better results and much bigger values when compared to the gbest. The better the efficiency of the search procedure, yet the more the average time will become necessary to attain the objective, the lesser the maximum achievable velocity or speed of each robot. The use of the potential field function as the fitness function was innovative in this work; one clear drawback is that it is a centralized control method. These PSO-based approaches have a fundamental fault in that they don't adequately address the issue of rapidly falling into local optima. Multiple robotic systems based on particle swarm optimization, conduct distributed detection work while avoiding obstacles in two PSO-based algorithms for robotic target seeking (called "RPSO" and "RDPSO"). Researchers presented an enhanced RDPSO technique in which robots were seen as ions, with a repulsive mechanism applied to the RDPSO between analogous ions to maintain consistent diversity levels and expedite convergence. A group of researchers presented A-RPSO, a multi robot system for target finding that has the benefits of speed and scalability. The authors investigated obstacle avoidance using A-RPSO as a robot control system. A-RPSO, on the other hand, requires a significant degree of robot-to-robot communication. PSO-based algorithms provide a straightforward calculation approach, a small number of control parameters, and are easy to implement. Despite the fact that PSO-based algorithms are excellent at regulating the complexity of robotic target searching, several concerns remain, such as the algorithms' proclivity for premature convergence and local optima.

To overcome these challenges from three perspectives, the suggested adaptive robotic bat algorithm approach employs multi or multiple swarm strategy, adaptive inertial weight, and Doppler Effect. The suggested adaptive robotic bat algorithm approach for multi or multiple swarm robots outperforms A-RPSO and Rbat algorithms in terms of performance.

CHAPTER 3

METHODOLOGY

3.1 Introduction

Methods based on swarm intelligence have been shown to be particularly successful in tackling difficult optimization issues. Due to various promising results when trying to solve various types of real-world problems and issues related to optimization, such as parameterization, extraction of features, mentoring the neural network, as well as the knapsack problem, SI based methodologies have become quite famous across several engineering disciplines when compared with the typical methods and procedure of optimization. Although there are several SI-based techniques, the swarm optimization or more specifically the PSO as well as another optimization technique based on ant colony are the two most famous as well as the frequently employed so far. These were influenced by the birds' social behaviour and ants' path-marking procedure that found its basis in pheromones while they are searching, hunting and looking around for the food. Numerous SI-based methods and techniques have recently been created and suggested in the literature, based on the behaviour of swarming of various animals some of them are FA, ABC, MBO, EHO, KH, MVO, DA, GWO, EWA, CS, and, BA which are firefly algorithm, artificial bee colony, elephant herding optimization, krill herd, monarch butterfly optimization, multi-verse optimizer, dragon fly algorithm, grey wolf optimizer, cuckoo search, earthworm optimization algorithm, and lastly the bat algorithm respectively. The bat algorithm led to success when tested on a vast version of the questions, challenges, and problems in the real-world scenarios.

3.2 Differential Evolution

DE, commonly written as differential evolution is basically an evolutionary algorithm that means, an algorithm based on evolutionary concept, or the concept of evolution in simple words. An Evolutionary algorithm that creates novel solutions and resolutions with the help of the mixing of the individual parent along with few additional members of the similar community. This can only recognize the novel resolutions, answers as well as solutions which are superior to those of their forefathers or parents. Differential evolution has minimal criteria to alter, making installation as well as modification of this method simple. Because of these benefits, it has a very vast variety of actual, practical implementations, including multi-objective optimization, job-shop scheduling, portfolio timetabling, as well as artificial neural training. Various DE techniques have been presented, based on the goal vector chosen and the number of differentiation vectors employed. The article uses the Differential evolution /rands as well as /1/bin method.

3.3 BAT Algorithm

Every bat is specified in BA by its location xt I speed vt I speed fi, volume At I as well as emissions heart rate rt i. The novel approaches at step involving time t, xt I, and velocity profile vt I are determined by

$$fi = f_{min} + (f_{max}\ f_{min}) \qquad 3.1$$

Wherein $\beta \in [0, 1]$ and it is a random number of vectors taken from a homogenous distribution. Thus, x represents the present, most up-to-date worldwide best answer. When a resolution is chosen, a new one is created locally utilizing a stochastic process, as seen below:

$$x_{new} = x_{old} + \varepsilon AT \qquad 3.2$$

Where [1, 1] is a random scaling factor and At = At I is the average loudness of all the bats at a time step.

3.4 Flowchart of project

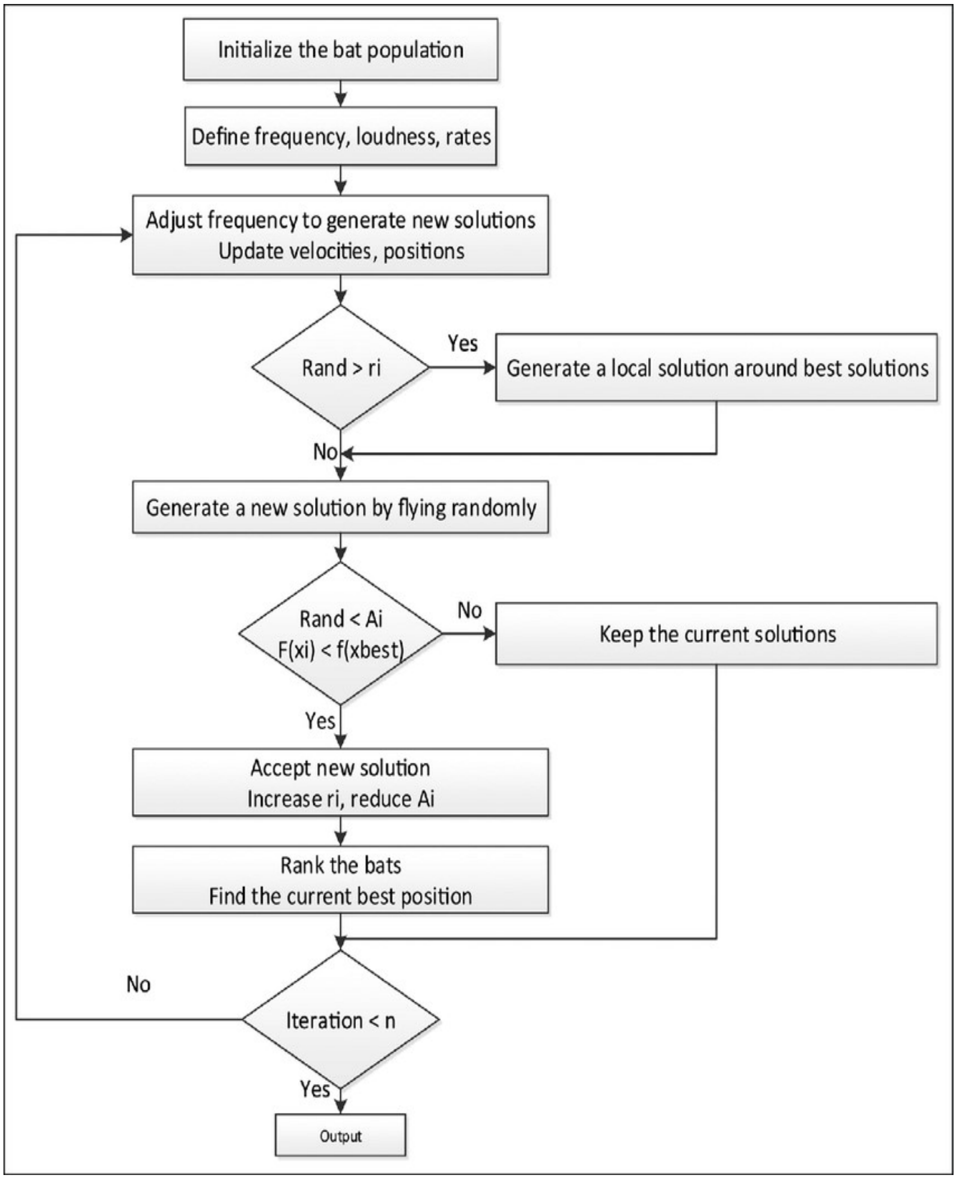

Figure 3.1: Flowchart of BAT Algorithm

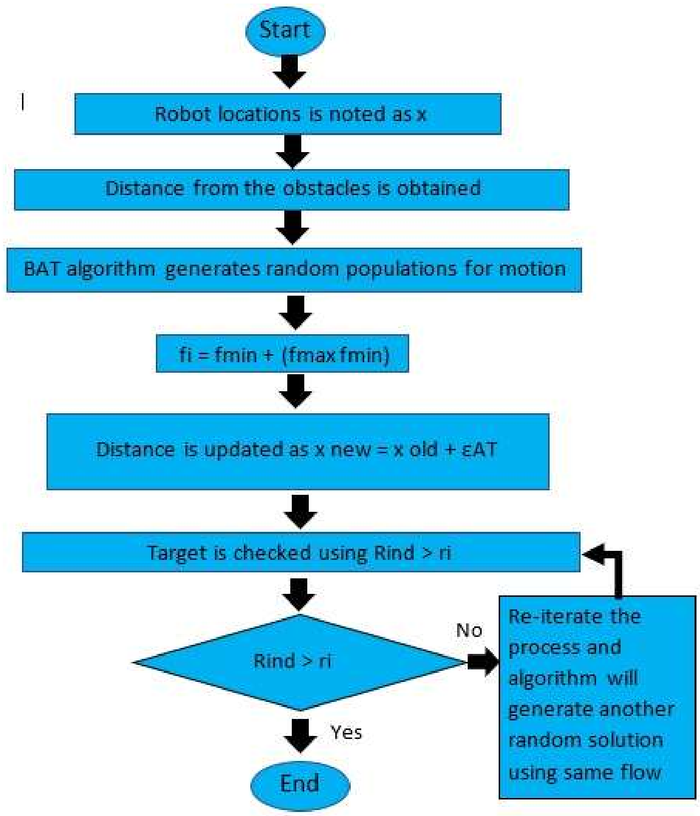

Figure 3.2: Flowchart of BAT Algorithm for 1 robot

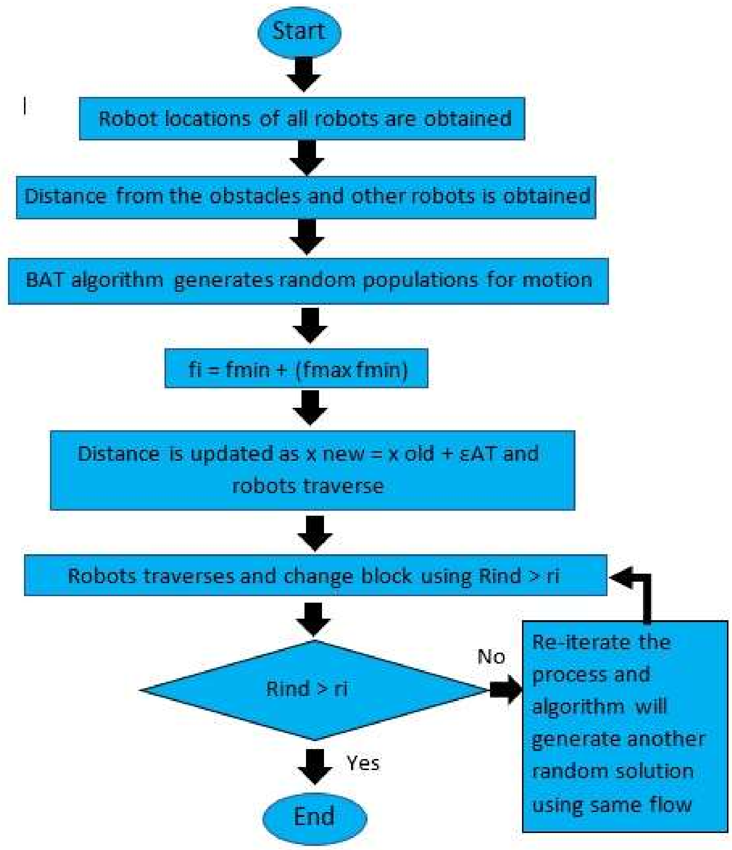

Figure 3.3 Flowchart of BAT Algorithm for multi robots

3.5 Proposed Methodology

Throughout many circumstances, Differential evolution is capable of searching worldwide and quickly locating the global ideal value. Nevertheless, substantially improving the answers in the mining phase is tough. Conventional bat algorithm, but on the other hand, is quite bad in exploring the answer. As a result, inside this study, an enhanced variant induced by a mutation in Differential evolution into the Bat algorithm, known as IBA, is employed to address the complex three-dimensional planning as well as managing problems related to the path. The distinction between IBA as well as BA would be that in IBA, the mutation operator is utilized to enhance the original BA by creating a new solution for every bat with such a probability of 1 r rather than the random walk is used in basic BA. As a result, IBA explores the environment both internationally and regionally, mostly based on the modification of the Differential evolution as well as the Bat algorithm respectively. This technique can effectively use differential evolution as well as the Bat algorithm.

The bat algorithm is presented. The bat model is enhanced by natural bats that use echolocation to evade natural adversaries and hunt prey. The update formula for the bat individual's position shown as, xxii and velocity written as, vvii at time node represented as, tt for a digital bat in the d-dimensional search area is provided here, in the current report.

The highest, as well as the lowest values for the initial setting for bat, vocalize frequency, are represented as f_{max} and f_{min} respectively. The present global optimal solution, denoted by xx in the preceding calculations, is the optimal fitness value of all individuals in the bat population. The single bat calculates its speed to the ideal solution based on its closeness to the world-wide optimal solution at tt1. The velocity of the individual bat in the following second is proportional to its proximity to the optimal solution accepted worldwide. The separate inertia of the bat algorithm is also put under consideration along with the previously observed acceleration process, values, and variations. The velocity in the next second is vvii, which would be modified by the prior moment's speed vvii tt1. Eq given above shows the way bat swarms move as their position, velocity and acceleration vary. The iteration procedure described above has been followed by the bat population in the search that is spread across the world, whereas the bat individuals near the ideal comprehensive solution employ the randomized strolling rule to construct a partially completely new solution, as indicated in described or stated equations. According to the calculation, whereas the bat algorithm approaches the best available solution indefinitely, the vocalization output decomposes continually until eventually stops. Furthermore, with each repetition, the vocalize rate of occurrence that is, frequency approaches asymptotically the original pulse rate rrii 0.

3.6 Mainframe of BA

The typical bat method provides features such as quick convergence time, high resilience, ease of implementation, and on and on. Nevertheless, it has been demonstrated that the typical BA has flaws in terms of convergence accuracy and rapidly slipping into the local optimum. In light of these concerns, we introduce three significant changes to the traditional bat method and integrate the features of the UAV route planning issue. The first step is to include the APF approach into the overall selection strategy. The second stage is to modify the weight of inertia of the speed iteration equation in the typical bat algorithm that initially or firstly proposes the best success rate approach to regulate the inertia weight. This suggested technique properly balances both international and domestic search processes. Third, the chaos technique is used to randomize the initial distribution of the bat population in order to speed up the searching process. These three enhancements are explained in detail. The operator based on mutation is derived from differential evolution, which brings the strategy and the function of mutation into the bat-based algorithm setup is the critical operator in IBA. Inside this approach, we incorporate the permutation into IBA, which alters answers with low fitness in order to increase population diversity and hence promote a better search experience. Because BA's search is exclusively based on non - stationary time series, a rapid resolution cannot be ensured all the time. For the very initial at, a major adjustment of the increasing and elevating mutation producer and manager is established to the algorithm based on bat, along with the available smaller improvements that are two in number, aimed at accelerating the process of converging. The first change is that we employ a fixed frequency f and a fixed loudness A instead of a variable frequency written as, fi and a variable loudness represented as, At I. In another algorithm which is IBA, every bat is characterized inside a single d-dimensional search area by its own location xt I speed vt I emissions rate of pulse rt I as well as fixed frequency f, loudness A. At the step involving the time t, the innovative approaches xt I as well as velocity profile vt I are supplied by vt I where, the x is written for the world's most appropriate solution in present time. In current paper, the f is adjusted to a value of 0.5. The second difference is to include a modified form in an approach or a try to boost demographic variety in terms of improving the efficiency of the functions such as searching and speeding up converging to optima. When one resolution is chosen from among the existing best approaches, workable formula for every bat is created locally using the stochastic process that is given in the above equations. Whenever the value ξ of is higher than that of the value of the pulse rate r, i.e., > r, where [0, 1] is any variable or random as well as real value or number selected from a uniform distribution; and when r, we employ the establish to produce the mutations in differential evolution to upgrade and install latest modification in the new solution to enhance population diversity and improve search efficiency by Equation previously mentioned. F is the alteration loading element and r1, r2, and r3 are uniformly distributed random integer integers ranging from 1 to NP. The foundation of the algorithm which is, IBA may be defined using the previous section analysis, as illustrated in the given Algorithm. NP represents the population size in Algorithm 1. The mutagenesis value of scaling is denoted by the letter F. Both vt I as well as xt I parameters represent both speed and position of the bats having the ith number at a given time period t respectively. Then, the upcoming progeny, NP, is represented by the xt u parameter and is a random integer that is evenly dispersed number ranging from 1 to NP, and the rand is the evenly dispersed number that is real and random in nature, ranging from 1 to NP. Then, the DE/rand/1/bin technique was employed and it depicted in the 1st algorithm. As it can be seen in Method 1, this algorithm simply has four regulating variables: NP, F, A, and Q.

3.7 Strategy for smooth path planning

In most circumstances, the course determined by meta-heuristic algorithms is difficult to fly precisely. Some strategies are applied to improve the generated route. In this case, a smoothing method based on B-Spline curves is utilized to dynamically smooth a viable route. Due to the reason that they require a limited number of variables to specify intricately curved routes, B-Spline curves are ideal for the evolutionary approach. B-spline curves are parametric curves, which means they are built using blending factors. When somehow the matching curve has n+1 command posts, the parameters of the B Spline arc can be stated, wherein u is indeed the random variable also called free parameter in this case, of the arc, Ni,p(u) are all the mixing factors of both the bend, as well as p is its level or degree. The mixing parameters are performed iteratively using a knot matrix U = u0,…, um, with the homogeneous non-periodic version being the most popular. Using the knot values specified above, the blending parameters Ni,p are calculated as follows: Ni,0 = 1 UI ui+1 0 elsewhere. B-Spline arcs are being used to establish a flight route even though they have the benefit of defining complex non-monotonic 3-dimensional curves using regulated regularity with only just a few design specifications. Following this procedure, the importance of establishing wi1wi wi wi+1 may be substituted with the path wi1 B.

This topic is split into two sections. The initial section goes into great depth on the MPSP. The (DBA) is presented in the above part. The problem of multi-point shortest route planning. Some nodes in imperfect connected graphs are chosen as required points in any MPSP. In a logical and distributive challenge, for instance, the items must be delivered to a certain location. The globe is an imperfect directed graph with a required points scale from 1 through n, wherein n represents the total places on the globe. We really would like to discover the quickest path from the starting location to the relevant places and back again. Every node in the network is reusable. The MPSP is comparable to, but not identical to, the TSP. The TSP's main goal is to discover the smallest Hamilton circle that encompasses all of the vertices in the whole graph. However, because most networks are not entirely graphs, we must identify the smallest circular structure or object in an imperfect linked graph that contains a beginning point in addition to this, certain essential points are also present, moreover, all the vertex in the graphical figure may be recycled or reutilized inside the circle inside the MPSP. The MPSP can be defined as something dealing with a much more complicated route planning challenge that includes node limitations. To decrease the problem's difficulty, a visual translation approach is utilized to turn the challenge into the TSP problem. Suppose G = (V, A) be an imperfect directed graphical figure; |V| = N denotes that V has N cardinality. The starting location is s V. M Vs could be described as the number of points that must be present in the aspects of the work. |M| equals n. The goal is to discover the shortest route in G that begins at s, traverses all of the locations in M, and is therefore not restricted to M, as well as returns to s. Every location in G could be visited over and over again.

3.8 Problem Resolution

The process of solving the problem includes Enhancements to the discrete bat algorithm, path coding as well as assessment, dealing with the variation, and eventually will result in the formation of the Bat algorithm. Given below section details the main step of designing a bat algorithm. Inside the DBA, fully written as, discontinuous bat algorithm, almost every bat adopts the identical 2-opt strategy to produce progeny. Whenever each bat would be far enough out from the greatest solution, 2-opt may well not cause it to develop; whenever a bat is near to the optimal method, 2-opt may demolish it. As a result, the discontinuous bat algorithm is vulnerable to localized fluctuations. As a result, we present an enhanced discontinuous bat algorithm shortly written as, IDBA in an attempt to increase the opportunity to move from out localized extremes. IDBA created a variable neighborhood generator action with three potential neighborhood contractors: new, 2-opt, and mutation. In the sequence of their difficulty, we place these

3 objectives into the controller set N = Nk(k = 1, 2, 3) as N1 = fresh or new, N2 = 2opt, and N3 = modification. For any bats, a straightforward evolutionary method has been devised. Assume k = 1 first; the controller Nk is employed. If Nk(x) is greater than x, the operation is repeated; otherwise, k = k + 1, and so on until k = 3. Throughout this manner, a bat may create progeny using several operators. If one controller can enhance the bat, it would be employed frequently till the capacity of the operator to evolve is lost. Whenever one activator misses, another one is selected to attempt to produce a superior progeny. Because the operators are arranged by their difficulty, the easiest one would be chosen first, aiding in the rate of convergence. We identify the two parameters to help with the definition of enhanced discontinuous bat algorithm:

- I s the current iteration number.
- mbat is the number of bats.
- A t I represent the volume of the ith bat in the tth iteration.
- r t I represent the heart rates or the rate of the pulse of the ith bat inside the tth iteration.
- A0 represents the initial price of loudness.
- r0 represents the initial value of heart rates.

The IDBA or enhanced discontinuous bat algorithm's stages are as follows:

We enhanced the evolving activator in the current algorithm over the Discrete Bat algorithm. The phases of neighborhood improvement in the Discreet Bat algorithm rely solely on the continual repetition of 2-opt. Because it is difficult to obtain a good price with a single objective optimization procedure, we employ variable neighborhood operators and favor the new facility owing to its relatively lower complexity. If no change occurs, we employ the more complex 2-opt, followed by the mutation operation. Transforming the neighborhood operation can alter the dispersion of the next solution, allowing you to identify the best option. The optimal path can then be smoothed for viable flying. According to the experimental data, the Integrated Bat algorithm is capable of finding a significantly shorter three-dimensional route. As a result, it can find the best three-dimensional path in difficult battle situations. It is worth noting that, an enhanced version of CS, DE/CS, is utilized to tackle the 3-dimensional routing problem. The quickest route achieved by DE/CS is 165.51 that is slightly poorer than the quickest route produced by IBA in this study. DE/CS and IBA provide extremely comparable ultimate ideal 3-D pathways. This means that CS, BA, and their variations are viable algorithms such as those of meta heuristic ones for addressing multi-dimensional issues related to the planning of the path of the object. The performance of the integrated bat algorithm exceptionally well on the task under study because of the inclusion of a operator of the mutations. Even though the mutation encourages minor alter the localized mobility in evolution-based computation, it results in abrupt alterations within the location of the candidate solutions in algorithms that find their basis in SI system. The initial actual and a Bat algorithm suffers from global optimal stagnation resultantly, the findings of this work show that the method used may overcome this limitation by rapidly altering the placement of the synthetic bats inside the search area. In BA, sudden moves of search agents result in a broad investigation of the solution space and, as a result, avoidance of optima of a specific location. The main and prime problem faced when dealing with 3-dimensional planning of the path is that it has a large variety of the localized optimization, making it difficult to address using optimization techniques. The suggested IBA's improved investigation as well as localized minima minimization help this algorithm overcome these barriers and beat the old Bat algorithm.

3.9 Trajectory Analysis:

The usefulness as well as efficiency of the bat algorithm was validated in current work by drawing a comparison and evaluating the paths of the robotics or the robotic path in the simulated outcomes of the prior presented theoretical bat algorithm. Six bots were distributed across the environment's headspace. To evaluate the paths of the algorithm more accurately, the robots interrupted the goal seeking simulation between iterations numbering from the 20th iteration to the 50th iteration. Many robots face difficulties in their route during the robot search and begin employing the collision detection as well as avoiding mechanism to avoid the obstacles; nonetheless, the robot continuously attempts to choose a "best position" location having better rating of the fitness as another location possible If somehow bot seems to be in the former stance, this is already "local best," but that will advance first from the recently resigned position to the perfect in the present situation upwards until it achieves a novel, better and fresh stance. The robotic fluctuates near the obstruction, and even if the variety of the machine is low, this condition will reoccur. This has a serious influence on the robot's exploration efficiency. The adjustable weight of inertia of the bat algorithm not just to improvise the variety of the bat algorithm, but it also successfully investigates the notion of avoiding slipping into a local optimal solution. If indeed the robot's investigation advances gradually inside the robotic objective discovery, the adaptive component of Formula will drop. If somehow the machines have a propensity to approach adjacent to one other, the accumulation value of Formula or more specifically, the accumulation factor of the equation will grow. The impact of weights and biases rises as the accumulation component or developmental factor or evolution factor increases, allowing the robots to keep a reasonable distance amongst themselves to guarantee variety and prevent slipping towards local optimal solution as much as practicable. The robots in BAT algorithm are spliced mainly into three sub swarms, one of them explores again for objective independently however interacts with the others, guaranteeing that every sub-swarm appears to shift forward towards the ideal sub-swarm. The target's location in an unfamiliar area is uncertain. The multiple swarm technique increases the variety of such bat algorithm and speeds up search time. The experimental findings indicates that the moving paths of something like the bat algorithm are cleaner than those of other algorithms such as, RBA. The adjustable inertia weights are used to enhance the A-RPSO as well as BAT algorithm. Furthermore, the ideal location of the traveling, mobile robot may accomplish dynamic adjustment through the Doppler Effect, rendering the robot's movement path finer than those of A-RPSO. Whenever the technique is initiated in the experimentation phase, three distinct robotic systems based on sub swarm technique are established. As already shown, the robots initially hunt for the objective independently, and then every sub-swarm of robots selects the optimal location in its sub-swarm. At the very same moment, every robotic sub-swarm collaborates with the others. The ideal location of every sub-swarm would be influenced by the optimal, suitable location of something like the optimum sub-swarm, as well as the robots would decide to migrate towards the location with the highest fitness value in order to obtain access to the ideal sub-swarm. In cycle number 78, the left robot sub-swarm reaches the objective while another robotic sub-swarm is still a set distance away. Nevertheless, as already demonstrated, A-RPSO as well as RBA cannot use the multi swarm technique. As a result, all robots advance to the world's optimal location at the same time, substantially lowering variety as well as making it easy to slip into the localized optimum; this affects the robot's searching efficiency as well as dependability. The A-RPSO, RBA, as well as bat algorithms had elapsed periods of 22.457264, 24.315032, and lastly 20.263236s, correspondingly. In summary, the bat algorithm employs a responsive inertia weight as well as a multi swarm strategic planning to enhance the uniqueness of automaton hunt as well as prevent the robot from having fallen into the locally optimal point, thereby broadening the browse the robotic spectrum, after realizing a much efficient quest, moreover, increasing the effectiveness of the robot objective scanning as well as searching. Furthermore, compensating for the Doppler Effect prevents convergence before time that is the convergence that occurs before maturation. In comparison to the rest of the two algorithms, the bat algorithm's iteration number was greater, and the circumstance where the robot's beginning location was closer to each other to further was analysed and superiority of the bat method was calculated. The robots' starting positions are all in the same region that is, the upper left corner. When the variety of robots was compared, it was seen to be diminished. Furthermore, since the robot's starting mean velocity number is tiny, the robot progresses slowly at the start of the search. As shown, RBA suffers from the major issue of prematurely, wherein the robots are unable to leave the locally optimized point until the 200th iteration. The A-RPSO technique performs better in terms of keeping the robots from becoming too near to each other, which allows the robot to go closer to the target; nevertheless, since all of the robots must move to a global optimum location, the benefits of overcoming the diversity issue are minimal. The Bat method provides a superior solution to this issue; instead of employing a single robotic group during goal searching, it splits all robots into several sub swarms. The sub-swarm robots can explore separately, therefore the multi-swarm technique enhances the robot's variety and search speed. Furthermore, adaptive variations in inertia weight contribute to the preservation of robot variety. The Doppler Effect aids in preventing the algorithm's quick or very slow convergence, hence avoiding premature convergence. Furthermore, the robot's trajectory was smoother. The robot does not locate the target in the RBA technique even after the completion of the 200th iteration cycle, however in the A-PSO, the robot can identify the objective following the 112th iteration cycle in the A-RPSO. To detect the object rapidly, the robot may conduct 97 rounds of the bat algorithm. The A-RPSO, RBA, and bat algorithms have elapsed times of 28.476364, 55.634182, as well as 22.344642 seconds, respectively. When compared to alternative algorithmic methods that are, RBA and A-RPSO, it was observed that the bat method can find targets quicker, more efficiently, and with superior performance. As seen, the robots are in a spreader pattern rather than focused. During iterations number 20 and 30, the robot halted in the simulations of the objective hunt to better assess the trajectories. For the same beginning position, the RBA robot finds the target after 74 iterations, the A-RPSO robot finds the target after 53 iterations, and the bat algorithm method finds the target after just 38 iterations (for the same starting position). Algorithms with elapsed periods of 12.272164, 18.329346, and 7.843822 seconds are the A-RPSO, RBA and bat algorithms, respectively. Because of the intelligent adjustment of inertia weight, the diversity of the A-RPSO system has expanded, and the search speed has improved. All of the robots in the RBA make an effort to reach the "global optimum" location; however, the robot at the best position moves slowly, which causes the search speed to be slowed significantly. This results in a lower overall efficacy of RBA compared to A-RPSO. It is proposed that the bat algorithm use an adaptive inertia weight strategy to improve robot variety by increasing the aggregation factor while simultaneously reducing the evolutionary factor to retain robot diversity. At the same time, the multi swarm approach has the potential to extend the search scope, reduce the amount of information sent between robots, increase computing speed, and encourage robot variation, all of which contribute to increasing the robot's search speed overall. In most circumstances, the three algorithms can accomplish the target-search job and are effective target search algorithms. In comparison to previous algorithms, the bat algorithm technique employs a multi-swarm technique as well as the Doppler Effect. When combating an unfamiliar ambiance having an unknown targeted object or location, the robot can identify target's direction of motion via sensor data. The resulting division of a subset into sub swarms may minimize processing time as well as the increase efficiency. In terms of temporal complexity, the bat algorithm technique is preferable. The iteration cycle frequency and time taken by the bat algorithm technique for the same beginning location are less than those of the other two approaches. Furthermore, the bat algorithm strategy gets the robot to the target faster than the other two approaches, as well as the bat algorithm technique seems to be more effective. Adaptive inertia weights are used in both the A-RPSO as well as the bat algorithm methods to improve the algorithm, which not only retains robot diversity but also optimizes the robot's next location, smoothing out the motion trajectory. Furthermore, by including the Doppler Effect into the proposed technique, the bat algorithm adaptively adjusts the frequency to raise the optimal position of the moving robot, and the Doppler Effect may improve the bat algorithm's performance. The A-RPSO as well as the bat algorithm techniques have smoother motion trajectories than the RBA, while the bat algorithm's robot has a smoother motion trajectory than the A-RPSO. Not only is the Bat algorithm capable of accurately looking for objects, but it is also faster. But even though the bat algorithm is better at identifying preventing obstacles, the robot may fall into local optimum if it meets a major impediment. As a consequence, the obstacle avoidance technique has to be improved.

CHAPTER 4

RESULTS AND DISCUSSION

4.1 Introduction

To evaluate the suggested strategy's effectiveness, a series of contextual simulations was processed using the MATLAB 2021a system software creating a virtual environment for robots and an Intel-core i7 16 GB quad core microprocessor chip. The optimization method's purpose is to increase the chance of cohesiveness as the agent approaches the destination whilst decreasing the likelihood of colliding with barriers and neighbouring agents. Overall weights of the different variables or parameters used in this computational research are listed in Table 1.

Parameter	Value
N	6
δ	4
A_{Min}	0
Γ	[0 to 1]
β	[0 to 1]
A	0.05
Γ_i^s $i\epsilon$ [1,2,….N]	1
T	[35,35]unit
A_0	1
f_i	[0 to 1]
ζ	[-1 to 1]
y	0.95

Table 1.1: Parameters of the BAT Algorithm

Fig. 4.1 illustrates one of the simulated outputs. The setting in this research is composed of stationary obstacles, depicted by the walls in the environment, like a maze, and rigid limits represented by outer boundry. The robot starts from its zero-position mentioned with a cross. The next target is to achieve the check point. Yhe robot starts heading towards the check point. The objective is shown by a coloured cross, depicting the end point, start point is also specified and the total area of the robot environment is [25 25] unit2. To create a swarm like multi robot topology, all the units are merged into an unstructured configuration beginning from their random placements. The team generated a polygonal form at t = 5 s. At t =7 s and t = 9s, the agents develop separate forms (polyhedral and elliptical) in response to the stationary obstacles.

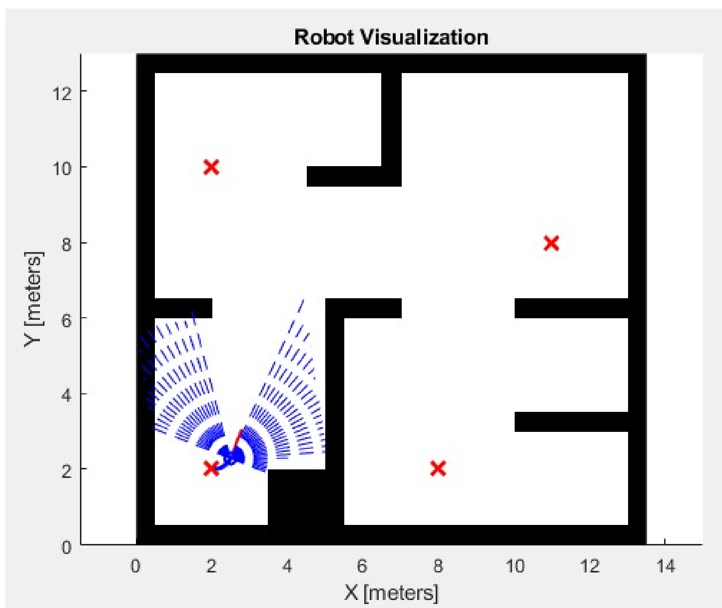

Figure 4.1: Path planning of single robot (Initial stage)

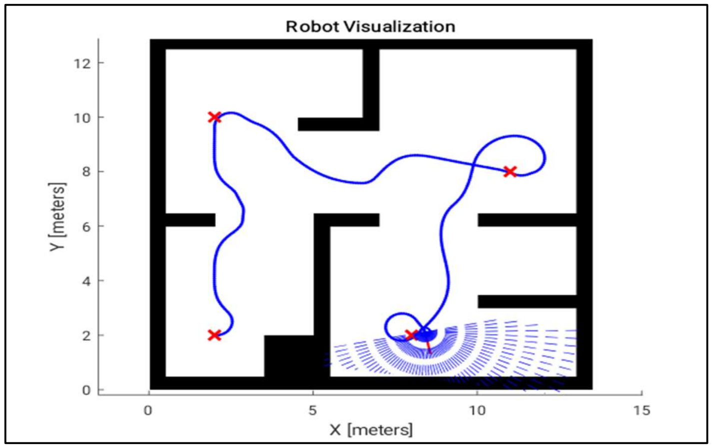

Figure 4.2: Path planning of single robot (Final stage)

Finally, after avoiding all impediments, the basic semi elliptical formation was regained very soon, and the complete system approached the targeting reticule mostly with specified configuration at t = 10s to 15s. In the starting phase, the robots are scattered without any segregation of colour. No groups are formed and initial positions of the swarm are quite random. Simulation results evidently show that the various new formations were formed inside the cluster throughout the transit more toward the target in order to maintain grouping and cohesiveness and to avoid obstacles.

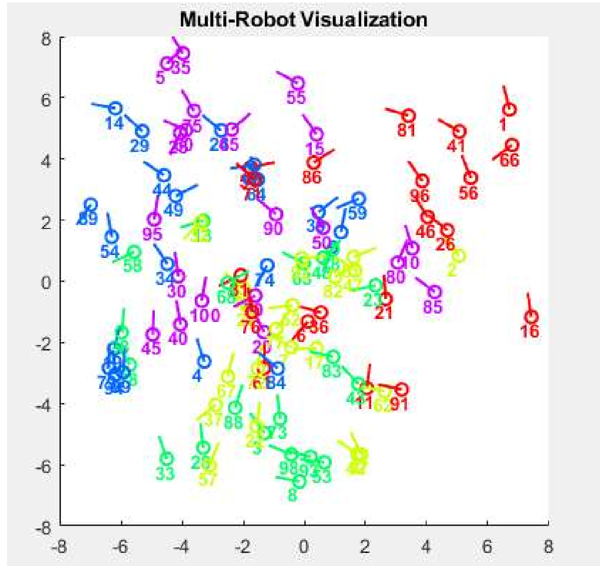

Figure 4.3: Swarm Optimization (Initial Stage)

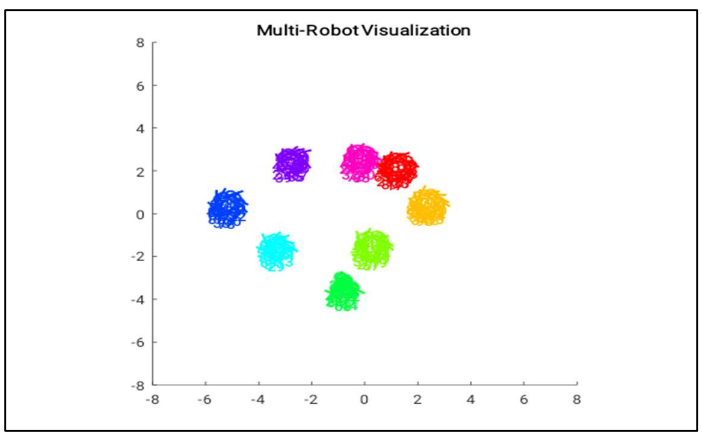

Figure4.4: Swarm Optimization (Final Stage)

Additionally, the starting and ending shapes are indistinguishable, indicating the absence of an obstacle avoidance controller in an obstacle free situation. Likewise, without any of the influence of a formed controller, the swarm can self-organize, therefore this management system is referred to as self-organising direction of control. To compare the suggested strategy to the pre referenced circle shaped based robotic swarm gathering approach (Sharma et al. 2017) in context of virtual environmental difficulty, it is estimated that two strategies operate efficiently in a relatively less complex situation.

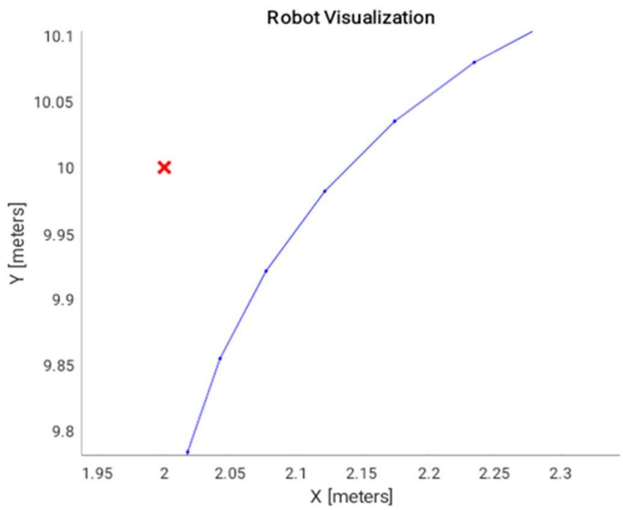

Figure4.5: Deviation (Error) of the path from first checkpoint

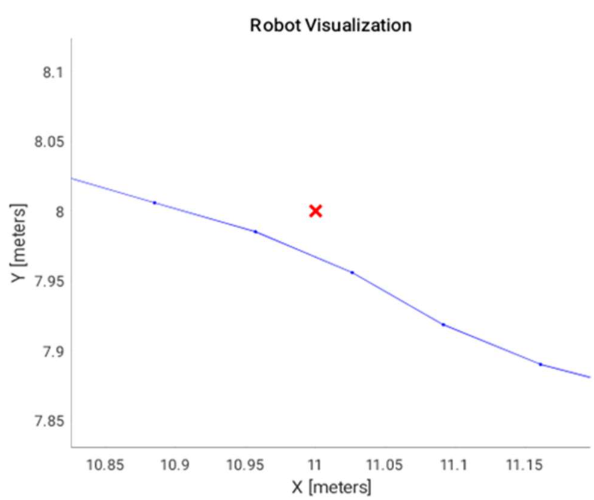

Figure4.6: Deviation (Error) of the path from second checkpoint

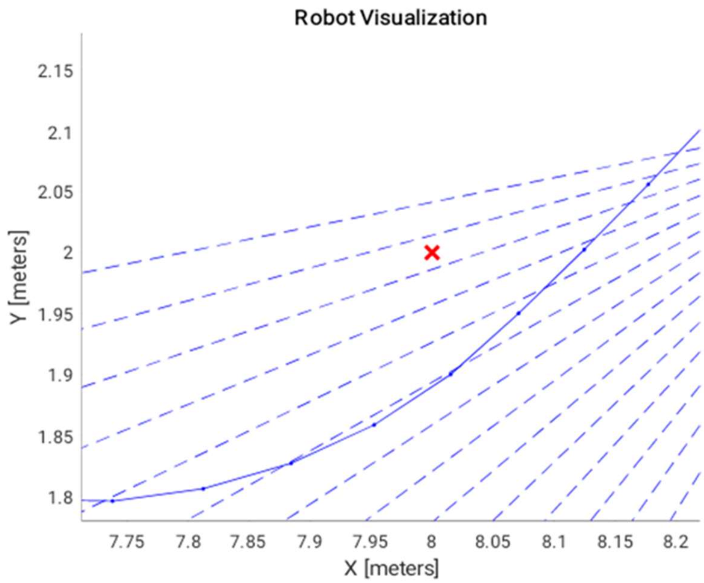

Figure 4.7: Deviation (Error) of the path from goal

Nevertheless, when the complexity of the environment increases (i.e., as the frequency of stationary barriers increases), our suggested BAT method outperforms the circle pattern gathering method due to the self-organising coordinated control characteristic. The proposed BAT based strategy is compared to other adaptive algorithms (Roy, 2016), (Ye. At al 2020) which have already been employed in the past to accomplish comparable goals. Coherence of the system when different gain settings are used. As before mentioned, the bat algorithm examines the controllers' gains (represented by ka, kv, kr, and kOA) in depending on the external world at each stage of the mobility while minimising the function i for the ith agent. A comparative study was undertaken to determine the mean difference of all the agents' coefficients during route planning, and standard deviation. Because when Bacterial foraging is used, the cluster always maintains a rigid configuration, however when PSO is used, no configuration is detected throughout route planning duration. When contrasting the stringent configuration (BG) as well as no configuration (PSO) scenarios, it is demonstrated that ka and kr are significantly greater than kOA for strict structure, meanwhile for no configuration, the opposite is witnessed. Additionally, the fluctuations in assessed quantities for no organization are far greater than those for rigorous configuration, which is most likely the primary explanation for non rigid configuration condition.

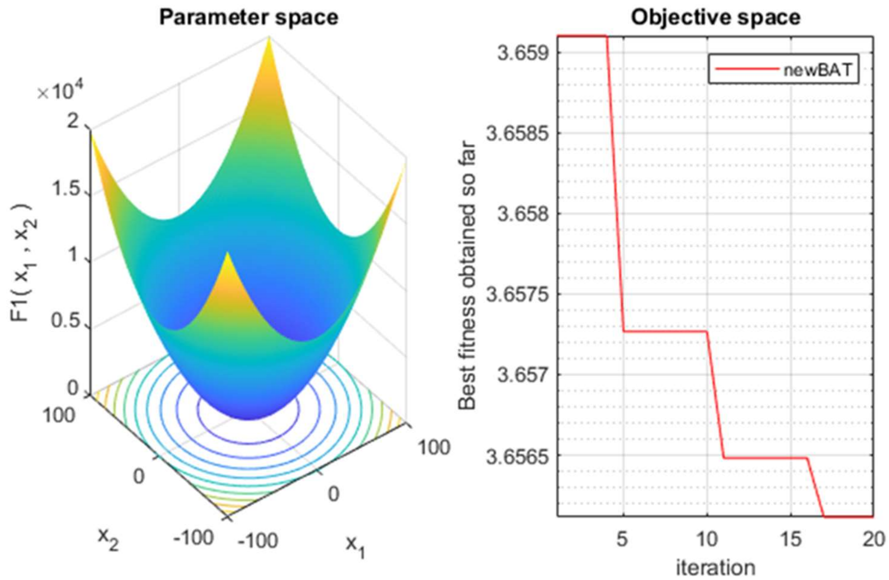

Figure 4.8: Optimal solution representation in BAT algorithm

The proposed methodology based on bat algorithm exhibits fewer discrepancy in the evaluated parameters, and the parameters are well within the definite boundaries of the above two situations, indicating the swarm's practically feasible group cohesion and intelligent trajectory planning and navigation system. For one robot, the robot follows the check points strictly and reaches the target position. The steady state error in the end point is very small, which is only visible if the visualizer is zoomed in. As we can see that there is minor variation in the first check point and path i.e. 2 and 2.05. Similarly for second check point, it is 8 and 8.01, likewise for end goal it is (8,2) but the position obtained is (8, 1.9). In next section three robots or agents constitute the swarm or team. Three bots are used in this experiment, and they are supposed to maintain a pre-determined target position (represented by crosses in the figure). At t=0.1 s, all the agents start from some random pose (indicated in figure below), probably close to a wall representing obstacle.

While remaining constant, they are getting closer at t=5s. Just then, the channel is blocked by a small tunnel (as seen in Figure below, where blockage is evident due to walls of the maze), and as a result, the agents alter their present inter agent arrangement in order to escape the obstacle and re-orient themselves or change the pose.

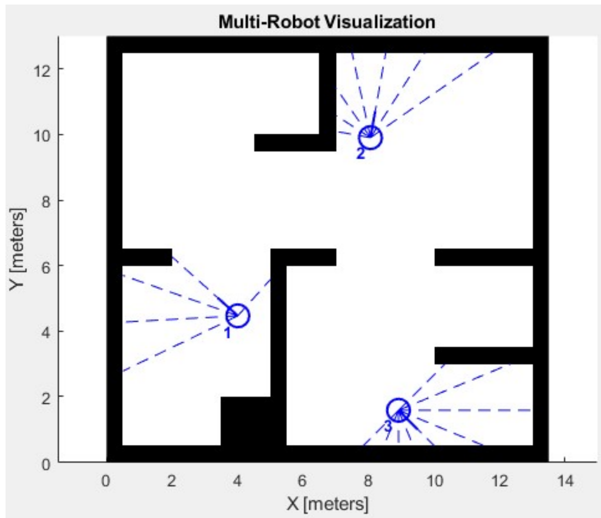

Figure 4.9: Simulation for three robots (Initial Stage)

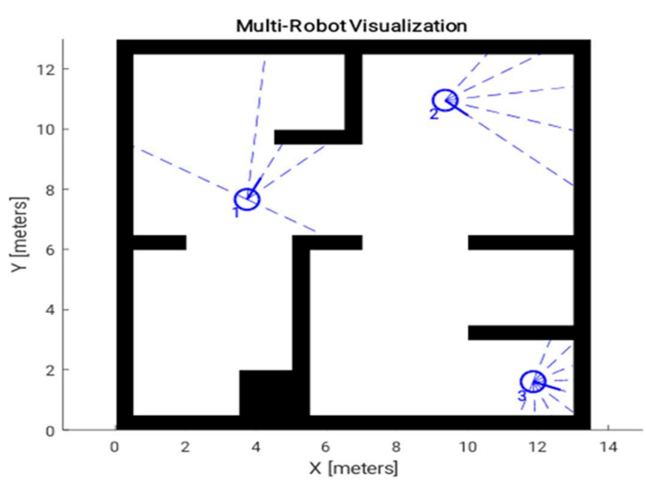

Figure 4.10: Simulation for three robots (after few seconds)

The agents traverse through the whole maze, avoiding collision with each other and all the obstacles. The starting positions of the robots can be changed and will give the same results. Thus, changing initial position does not affect the path planning and obstacle avoiding capabilities of the algorithm.

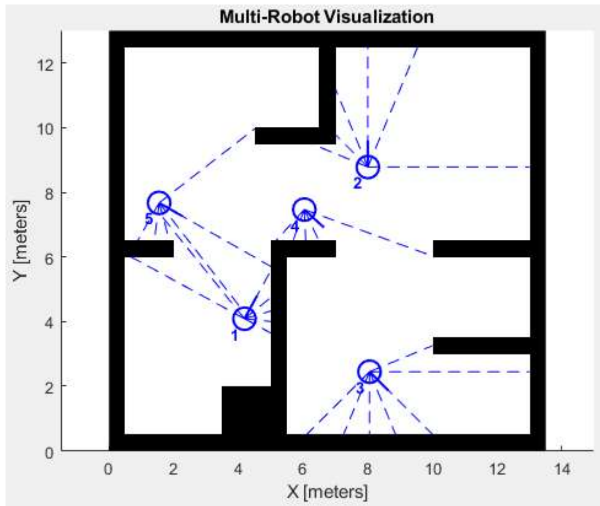

Figure 4.11: Simulation for five robots (Initial Stage)

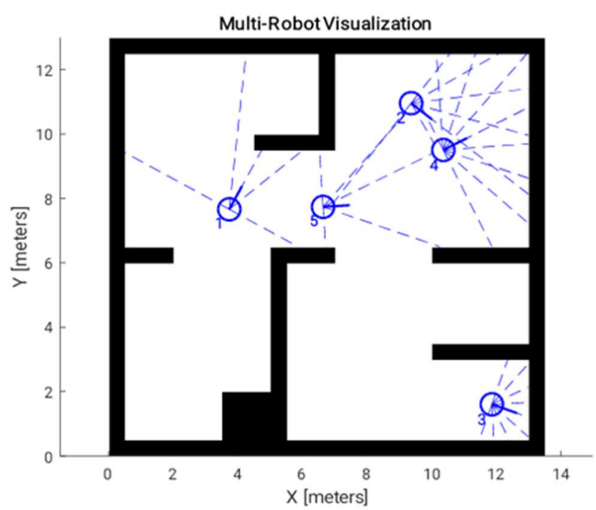

Figure 4.12: Simulation for five robots (after few secs)

The algorithm works efficiently as compared to the before mentioned algorithms but a bit slower than the single robot system implemented using same technique due to dynamic obstacles in form of other robots present in the environment and traffic, thus minimizing the free space for each robot. For a group of five robots, the dynamic obstacles are increased because each moving robot is an obstacle for the other that is to be avoided. Hence the speed and mobility of the system is reduced.

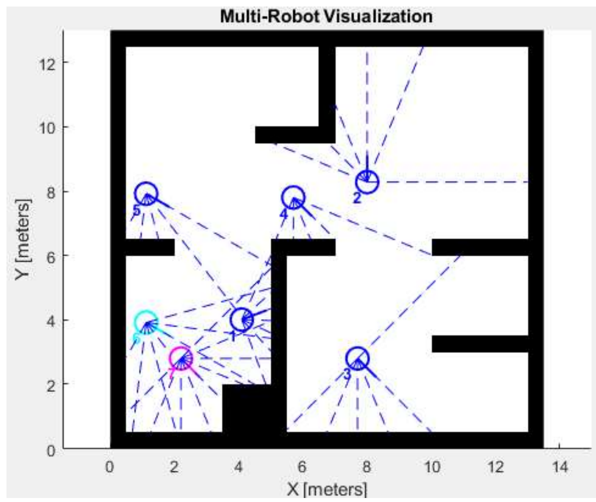

Figure 4.13: Simulations for seven robots (Initial Stage)

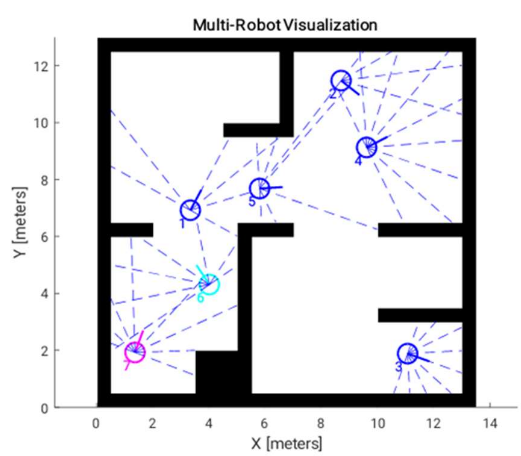

Figure 4.14: Simulation of seven robots (after few secs)

Similarly, in seven robots, the number of dynamic obstacles is further increased and free space is reduced, posing constraints on each robot, hence, mobility and traversal speed of robots is reduced. So, is the case with ten robots.

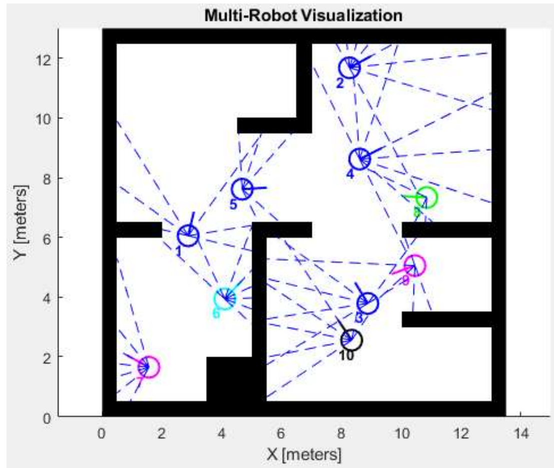

Figure 4.15: Simulation for ten robots (Initial Stage)

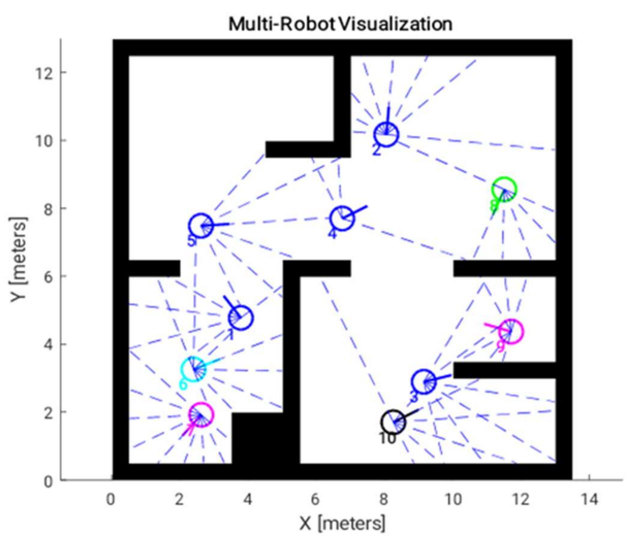

Figure 4.16: Simulation for ten robots (after few secs)

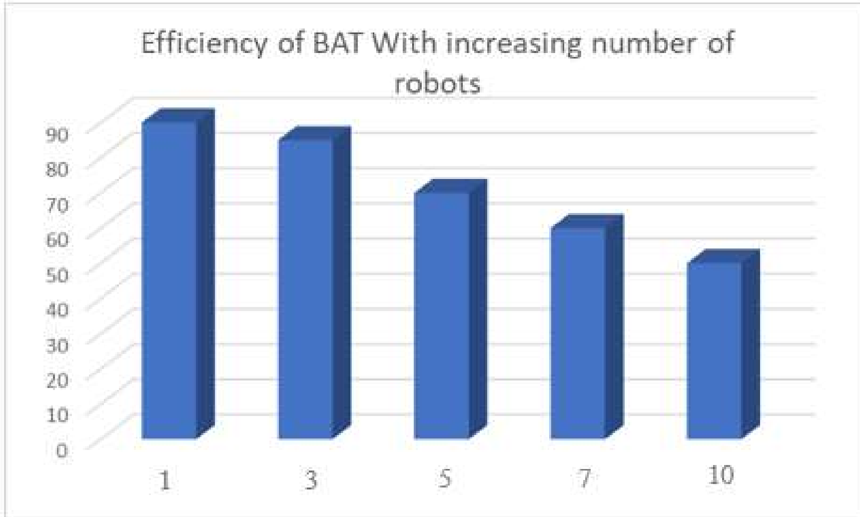

Figure 4.17: Comparison of efficiency

CHAPTER 5

CONCLUSION

5.1 Introduction

Swarm optimization and multi robot obstacle avoidance are important research areas due to their vast applications in environments that require multiple humans to cooperate and work together especially in dangerous situations. These robot teams can be made to integrate and coordinate with each other to ensure human safety and optimal task execution.

Multiple topologies have been studied to perform the task. One of them, which is under consideration here is a bio inspired algorithm which utilizes the echolocation method used by bats for navigation and is named as BAT algorithm. In this algorithm, a cluster of microbats is used to generate random solutions and the best of them is chosen for the robot to move. This technique has outperformed many traditional methods and algorithms available. The simulation is carried out using a swarm of robot, one robot, three, five and five, seven and ten robots in a maze-like environment having static and non- static obstacles.

The BAT algorithm, when used by introducing slight mutations in it can serve in building corporative behaviour and techniques separately. It does so by enhancing the search ability of the system. Moreover, it developed a system on a robot that can execute waypoint navigation while taking into account local information, with a focus on allowing each robot to contain and manage a single BAM particle. In current study initial waypoint path was created via known barriers as well as a reactive planner that reacts to pop-up obstacles. TThehe steady state error is very small and robot closely achieves the target. The execution and path planning slows down a bit for higher numbers due to limited free space and more dynamic hurdles in the way. Further studies are being conducted to find more nature inspired methods to enhance the technology.

5.2 Recommendations

The BAT algorithm performs really well as compared to other competitor algorithms. Such as PSO and other algorithms of same nature and is widely used but this algorithm can be further improved by its combination with other algorithms that may or may not use echolocation. This is because, our algorithm faces challenges for multi-robotic environment because of the increased dynamic obstacles. If combined with a faster technique, it can perform well in dynamic environments as well, sharply reducing the steady state error and improving the transient response of the system.

REFERENCES

Yang, X.-S. (2010). A new metaheuristic bat-inspired algorithm. Nature inspired cooperative strategies for optimization (NICSO 2010), Springer: 65-74.

Fister, I. (2013). A comprehensive review of bat algorithms and their hybridization, Univerza v Mariboru, Fakulteta za elektrotehniko, računalništvo in informatiko.

Villalón, C. L. C., Stützle, T., & Dorigo, M. (2020, October). Grey wolf, firefly and bat algorithms: Three widespread algorithms that do not contain any novelty. In *International Conference on Swarm Intelligence* (pp. 121-133). Springer, Cham.

Ahmadlou, M., Karimi, M., Alizadeh, S., Shirzadi, A., Parvinnejhad, D., Shahabi, H., & Panahi, M. (2019). Flood susceptibility assessment using integration of adaptive network-based fuzzy inference system (ANFIS) and biogeography-based optimization (BBO) and BAT algorithms (BA). *Geocarto International*, *34*(11), 1252-1272.

Ye, F. L., Lee, C. Y., Lee, Z. J., Huang, J. Q., & Tu, J. F. (2020). Incorporating particle swarm optimization into improved bacterial foraging optimization algorithm applied to classify imbalanced data. *Symmetry*, *12*(2), 229.

Yigit, B., Alapan, Y., & Sitti, M. (2020). Cohesive self-organization of mobile microrobotic swarms. *Soft Matter*, *16*(8), 1996-2004.

Yoshioka, C., & Namerikawa, T. (2008). Formation control of nonholonomic multi-vehicle systems based on virtual structure. *IFAC Proceedings Volumes*, *41*(2), 5149-5154.

Yao, Weijia, Huimin Lu, Zhiwen Zeng, Junhao Xiao, and Zhiqiang Zheng. "Distributed static and dynamic circumnavigation control with arbitrary spacings for a heterogeneous multi-robot system." *Journal of Intelligent & Robotic Systems* 94, no. 3 (2019): 883-905.

Yamada, Y., Ito, K., Tsuji, T., Otani, K., Kobayashi, R., Watanabe, Y., & Hiryu, S. (2019). Ultrasound navigation based on minimally designed vehicle inspired by the bio-sonar strategy of bats. *Advanced Robotics*, *33*(3-4), 169-182.

Wu, H., An, B., & Li, B. (2020). Distributed consensus control protocols for heterogeneous multi-agent systems with time-varying topologies. *IEEE Access*, *8*, 152772-152779.

Wang, Xiangyu, Shihua Li, Xinghuo Yu, and Jun Yang. "Distributed active anti-disturbance consensus for leader-follower higher-order multi-agent systems with mismatched disturbances." *IEEE Transactions on Automatic Control* 62, no. 11 (2016): 5795-5801.

Sharma, Gokarna, Costas Busch, Supratik Mukhopadhyay, and Charles Malveaux. "Tight analysis of a collisionless robot gathering algorithm." *ACM Transactions on Autonomous and Adaptive Systems (TAAS)* 12, no. 1 (2017): 1-20.

Roy, D.: Some Studies on Structural Controllability and Optimum Link Weight Assignment of Complex Networks. PhD diss., (2013)

Roy, D., Maitra, M.: Studies on strong structural controllability and optimum link weight assignment of complex networks. In 2013 Annual IEEE India Conference (INDICON). IEEE, pp. 1–6 (2013)

Roy, D., Maitra, M.: Studies on strong structural controllability and optimum link weight assignment of complex networks. In 2013 Annual IEEE India Conference (INDICON). IEEE, pp. 1–6 (2013)

Roy, D., Maitra, M., Bhattacharya, S.: Study of formation control and obstacle avoidance of swarm robots using evolutionary algorithms. In: 2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC). IEEE, pp. 003154-003159 (2016)

Roy, D., Chowdhury, A., Maitra, M., Bhattacharya, S.: Robust Path Planning of Swarm Robots using PSO assisted Bacterial Foraging. In: The Second Workshop "Evaluating General-Purpose AI" (EGPAI2017) in conjunction with IJCAI (2017)

APPENDIX A
APPENDIX 1

Code:

```
% Define Vehicle
Rad1 = 0.1;                       % Wheel radius [m]
Len1 = 0.5;                       % Wheelbase [m]
dd = DifferentialDrive(Rad1,Len1);

% Sample time and time array
sample_Time = 0.1;             % Sample time [s]
tVec = 0:sample_Time:45;       % Time array

% Initial conditions
initialPosition = [2;2;0];           % Initial pose (x y theta)
position= zeros(3,numel(tVec));    % Pose matrix
position(:,1) = initialPosition;

% Load map
close all
load exampleMap

% Create lidar sensor
lidarSens = LidarSensor;
%Lidar Sensor parametes are specified
lidarSens.sensorOffset = [0,0];
lidarSens.scanAngles = linspace(-pi/2,pi/2,51);

lidarSens.maxRange = 5;


N=10; % Number of Bats

Function_name='F1'; % Name of the test function

Max_iter=200; % Maximum number of iterations

% Load details of the selected benchmark function
[lb,ub,dim,fobj]=Get_Functions_details(Function_name);
[bestfit,BestPositions,fmax,BAT_Cg_curve]=newBAT(N,Max_iter,lb,ub,dim,
fobj);
figure('position',[500 500 660 290])
%Draw search space
subplot(1,2,1);
func_plot(Function_name);
title('Parameter space')
xlabel('x_1');
ylabel('x_2');
zlabel([Function_name,'( x_1 , x_2 )'])

%Draw objective space
subplot(1,2,2);
semilogy(BAT_Cg_curve,'Color','r')
title('Objective space')
xlabel('iteration');
```

```
ylabel('Best fitness obtained so far');
axis tight
grid on
box on
legend('newBAT')

display(['The best solution obtained by BAT is : ',
num2str(BestPositions)]);
display(['The best optimal value of the objective funciton found by
BAT is : ', num2str(bestfit)]);




% Create visualizer
visual = Visualizer2D;
visual.hasWaypoints = true;
%Visualized map parameters are specified
visual.mapName = 'map';
attachLidarSensor(visual,lidarSens);

%% Path planning and following

% Create checkpoints for robot
checkpoints = [initialPosition(1:2)';
             2 10;
             11 8;
             8 2];

% BAT Controller
controller = controllerPurePursuit;
controller.Waypoints = checkpoints;
controller.LookaheadDistance = 0.5;
controller.DesiredLinearVelocity = 0.75;
controller.MaxAngularVelocity = 1.5;

% Vector Field Histogram (VFH) for obstacle avoidance
vfh = controllerVFH;
vfh.DistanceLimits = [0.05 3];
vfh.NumAngularSectors = 36;
vfh.HistogramThresholds = [5 10];
vfh.RobotRadius = Len1;
vfh.SafetyDistance = Len1;
vfh.MinTurningRadius = 0.25;

%% Simulation loop
r = rateControl(1/sample_Time);
for id_x = 2:numel(tVec)

    % Get the sensor readings
    currentPose = position(:,id_x-1);
    totranges = lidarSens(currentPose);

    % Run the path following and obstacle avoidance algorithms
    [vReference,wReference,lookAheadPoint] = controller(currentPose);
    targetDir = atan2(lookAheadPoint(2)-
currentPose(2),lookAheadPoint(1)-currentPose(1)) - currentPose(3);
    steerDir = vfh(totranges,lidarSens.scanAngles,targetDir);
    if ~isnan(steerDir) && abs(steerDir-targetDir) > 0.1
        wReference = 0.5*steerDir;
```

```
    end

    % Control the robot
    velocityB = [vReference;0;wReference];                     % Body
velocities [vx;vy;w]
    veloci = bodyToWorld(velocityB,currentPose);  % Convert from body
to world

    % Perform forward discrete integration step
    position(:,id_x) = currentPose + veloci*sample_Time;

    % Update visualization
    visual(position(:,id_x),checkpoints,totranges)
    waitfor(r);
end
```

APPENDIX B
APPENDIX 2

Simulink Model of Robot: