DESIGN AND DEVELOPMENT OF BAT ALGORITHM STRATEGIES FOR DYNAMIC ENVIRONMENT IN APPLICATION SPECIFIC CONTEXT

A

Thesis

Submitted to

For the award of

DOCTOR OF PHILOSOPHY (Ph.D)

in

COMPUTER SCIENCE AND ENGINEERING

By

Shabnam Sharma

41400206

Supervised By | **Co-Supervised By**

Dr. Sahil Verma | **Dr. Kiran Jyoti**

LOVELY FACULTY OF TECHNOLOGY AND SCIENCES
LOVELY PROFESSIONAL UNIVERSITY
PUNJAB
2019

DECLARATION

I hereby declare that thesis entitled "Design and Development of Bat Algorithm Strategies for Dynamic Environment in Application Specific Context" submitted by me for Degree of Doctor of Philosophy in Computer Science and Engineering is the result of my original and independent research work carried out under the guidance of Supervisor Dr. Sahil Verma, Associate Professor, School of Computer Science and Engineering, Lovely Professional University, Jalandhar, and Co-Supervisor Dr. Kiran Jyoti, Assistant Professor and Head of Information Technology department, Guru Nanak Dev Engineering College, Ludhiana. This work has not been submitted for the award of any degree or fellowship of any other University or Institution.

Shabnam Sharma

School of Computer Science and Engineering,

Lovely Professional University,

Phagwara, Punjab-144411,

India

Date:

CERTIFICATE

This is to certify that the thesis entitled "Design and Development of Bat Algorithm Strategies for Dynamic Environment in Application Specific Context" submitted by Shabnam Sharma for the award of the degree of Doctor of Philosophy in Computer Science and Engineering, Lovely Professional University, is entirely based on the work carried out by her under my supervision and guidance. The work reported, embodies the original work of the candidate and has not been submitted to any other university or institution for the award of any degree or fellowship, according to the best of my knowledge.

Dr. Sahil Verma
Associate Professor
School of Computer Science and Engineering
Lovely Professional University
Phagwara, Punjab-144411
India
Date:

Dr. Kiran Jyoti
Assistant Professor &
Head- Department of Information
Technology department
Guru Nanak Dev Engineering College
Ludhiana, Punjab-141006
India
Date:

ABSTRACT

Research in ‘Optimization’ ranges from the design and analysis of algorithms to their software implementations. A substantial area of optimization is the formulation of algorithms which are representative of real-world applications. Optimization problems arise in all areas of science and engineering. Most realistic optimization algorithms deal with uncertainty of algorithms’ parameters and data. One of the challenge of optimization algorithms is how to achieve optimized results on large scale using optimization algorithms.

Different optimization techniques/algorithms are adopted to obtain optimized results in different application areas. Swarm Intelligence (SI) based optimization is popular metaheuristic technique which is developed and inspired by the collective behavior of swarms. SI has attracted significant attention of researchers in past decade. Given that Ant Colony Optimization, Particle Swarm Optimization, Genetic Algorithms, Firefly Algorithm, Bat Algorithm and many more, belong to same family and share the common characteristics of ‘swarm’. But, this thesis focuses on Bat Algorithm which is inspired by echolocation of a colony of bats. Due to astonishing echolocation behavior of bats, frequency tuning, automatic parameter updation and automatic zooming, Bat Algorithm is preferred over other Swarm Intelligence algorithms. The aim of the research is to introduce novel variants of Bat Algorithm by incorporating the biological behavior of bats.

Bat Algorithm relies on the assumption of calculating distance between bat and prey, in a ‘magical way’. In order to obtain optimal solution ‘timely’, while satisfying underlying constraints, it becomes important to determine the range between prey and bat. It is also important to track the movement of prey (target). The movement strategy adopted by bats depend upon prey’s movement. Few preys tend to move at constant speed and direction, which is predictable to bats. The first objective of the thesis is to develop such a variant of Bat Algorithm, which can determine range between prey and bat. The algorithm proposed in this research work will help in

tracking targets moving at constant or predictable speed. The proposed algorithm is based on Constant Bearing pursuit strategy, which will help in achieving more optimized solution. This type of algorithm is suitable for Cloud Computing environment, where motive is to select optimal virtual machine and selection of virtual machines is done more likely from same zonal area. So, movement of target (i.e. Virtual Machine) is within same zonal area, but virtual machine may differ. So, in this case, movement of target may be negligible (in case of selection of same optimal virtual machine) or may be predictable (in case of selection of optimal virtual machine from same zonal area, but different virtual machine).

Furthermore, Bat Algorithm is modified by incorporating different strategies which are adopted by bats for targeting erratically moving targets. In the presence of multiple prey (targets), selection of optimal prey becomes crucial. Selection of feasible solutions depend upon the range between bat and prey, and also depends upon movement of prey. As per biological features of bats, they adopt different pursuit strategies for capturing static prey or moving at predictable speed or moving at unpredictable speed. The second objective of the thesis is to develop such a variant of Bat Algorithm, which can track targets moving erratically. This algorithm primarily focuses on tracking of such preys, which are moving erratically, i.e. at unpredictable speed. The proposed algorithm is based on Constant Absolute Target Detection pursuit strategy, which will help in achieving more optimized solution. This type of algorithm is suitable for routing through sensor nodes in WSN, where motive is to select optimal sensor node either for acting as sender or recipient or may be as intermediate (forwarding) node. So, movement of target (i.e. sensor node) is very frequent in most of the cases. So, deployment of such algorithm becomes necessity to track such unpredictable targets (sensor nodes).

Another variant is developed which has incorporated different pursuit strategies. Most of the nature inspired optimization techniques relies on fact of obtaining optimal solution with the collaborative work of swarm population. Bat Algorithm is such a nature inspired optimization technique, where one bat of a swarm jams/blocks sound produced by another bat of same swarm or it may steal the information encoded in the sound produced and the received echo or by entering in ‘silent’ mode. In the existing literature survey, presence of other target seekers (bats), seeking for optimal solution (prey) in the same search space is not considered. Existing literature and this behavior of bats motivate to develop another variant of Bat Algorithm. Bats which are the part of same colony, do not work collaboratively to obtain the solution. Bats prefer to search their optimal solution individually while utilizing the information available in search space and conserving their energy levels. The third objective of the thesis is to explore different pursuit strategies which are adopted by bats while capturing their targets. The proposed algorithm will help in those scenarios, where energy conservation is one of aspects while obtaining optimal solution. It can be implemented in solving Traveling Salesman Problem, where motive is to take advantage from traveling plan of another traveler without preparing its own plan, as one traveler will be following the other traveler.

Moreover, the movement strategies of bats are also studied in this research work. Bats adopt three different types of pursuit strategies: Following, Converging and Diverging. It has been noticed during result evaluation that 65% times, bats adopt ‘following’ pursuit strategy. Most of the times ‘follower bats’ are able to capture their targets (preys) as compared to ‘leader bats’.

The tools, methodologies, and approach used for different variants of Bat Algorithm are detailed in the thesis. To evaluate the performance of all (three) proposed variants of Bat Algorithm, best, worst, mean, median and standard deviation are considered as parameters. The results are evaluated for varying bat population, i.e. [25,50,75,100] over varying number of iterations, i.e. [250,500,750,1000]. The performance of three proposed/developed variants of Bat Algorithm are verified through rigorous tests over thirteen optimization benchmark test functions. Further, performance of proposed variants of BA are assessed by solving three real-world problems. The results validate better performance of proposed algorithms for solving single-objective optimization problems.

ACKNOWLEDGEMENT

It is a pleasure for me to thank all those who have helped me to accomplish this Ph.D. thesis. Firstly, I wish to express my deepest gratitude to Dr. Sahil Verma and Dr. Kiran Jyoti for guiding me throughout this research work. Both of my supervisors have been a continuous source of knowledge, inspiration, motivation and encouragement during the entire course of this research work.

A special thanks to the management of Lovely Professional University for supporting me in the best possible manner and facilitating me in balancing my work and my research. The doctoral programme of LPU has made it possible for me to pursue my dream of research and upgrade my knowledge.

I thank my friends, especially Monica Sood and Aditya Prakash, for being there for me and motivating me at every step to achieve this goal. Without their support during the numerous ups and downs and their constant encouragement, completion of this journey would have been impossible.

Special thanks to Examiners of end term reports and reviewers of journals who vetted my submissions and gave valuable comments to improve the work further.

I wish to express my profound gratitude to my parents and all other members of my family. Their love, support and unshakable faith in me provides umpteen strength to succeed in all the goals of life. I am also thankful to my brother Abhinav Sharma who has provided the full support to me during the entire period of my research work. I take this opportunity to express my gratitude to all my teachers who have shaped me and have contributed immensely to my knowledge and skills development since childhood.

Finally I would like to thank each and every person who has directly and indirectly helped and motivated me in this herculean task.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVATIONS

ABC	Artificial Bee Colony
ACO	Ant Colony Optimization
ASCHEA	Adaptive Segregation Constraint Handling Evolutionary Algorithm
BA	Bat Algorithm
BGRA	Bacterial Gene Recombination Algorithm
BVM	Balanced Virtual Machine
CATD	Constant Absolute Target Detection
CATD-BA	Constant Absolute Target Detection Bat Algorithm
CB	Constant Bearing
CS	Cuckoo Search
DE	Differential Evolution
EA	Evolutionary Algorithm
EGHS	Effective Global Harmony Search
FA	Firefly Algorithm
FBI-BA	Flight Behavior Inspired Bat Algorithm
FCPSO	Fully Constrained Particle Swarm Optimization
GA	Genetic Algorithm
MBA	Mine Blast Algorithm
MBBA	Multi Objective Binary Bat Algorithm
MOGOA	Multi-Objective Grasshopper Optimization Algorithm
MPSO	Master-Slave Particle Swarm Optimization
NSES	Novel Selection Evolutionary Strategy
OOVM	Overloaded Optimal Virtual Machine
PCPSO	Partially Constrained PSO
PSO	Particle Swarm Optimization
RD-BA	Range Determiner Bat Algorithm
RIO	Roach Infestation Optimization
SI	Swarm Intelligence

SiC-PSO	Simple Constrained Particle Swarm Optimizer
SINNS	Swarm Intelligent Neural Network System
SMABBPSO	Bare bones particle swarm optimization with scale matrix adaptation
SMPSO	Speed-Constrained Multi Objective PSO
SSO	Social Spider Optimization
UPSO	Unified Particle Swarm Optimization
VRP	Vehicle Routing Problem
WSN	Wireless Sensor Network

CHAPTER 1
INTRODUCTION

In this chapter, various optimization problems are briefly discussed, which is the foundation of this research work. In first section, basic idea of optimization is presented, followed by second section, which emphasis on single optimization problems and solutions to problems. Third section describes constrained optimization problems. In fourth section, multi objective optimization problems are explored and suggested solutions to problems. In fifth section, Swarm Intelligence techniques are described. In sixth and seventh section, behavior of real bats are explored and presented. In eighth section, Bat Algorithm (BA) is described, followed by existing variants of BA. Next section presents the different way of deriving new variants. At last, research aim and objectives of thesis are described.

1.1 Optimization: Overview

Optimization is considered to be the subset of mathematics; which include review of techniques, procedures, methods, algorithms to obtain optimum result to a given problem [6]. Optimization is process of obtaining best solution of any problem either by using minimization or maximization function, while specifying underline constraints [165] [189] [164] [147]. The author of [191] has mentioned in their research work that optimization process involves defining of objective and fitness function. These underlying fitness functions should satisfy respective parameters of interest and related constraints in order to provide solution to the problem. In the current era, optimization is applicable in almost every aspect of life, business, management or engineering designs, in order to reduce cost, time and resources, while improving performance, better results and increased profit [189]. The principle objectives of providing optimized solution are Design variables i.e. a numerical input that will change during the process of optimization; Objective function i.e. describes main motive of the function i.e. either to be minimize or maximize, depending upon nature of problem; Constraints i.e. conditions that must be satisfied while solving the problem and Standard Formulation i.e. representation of problem in mathematical notation, as depicted in Figure 1.1.

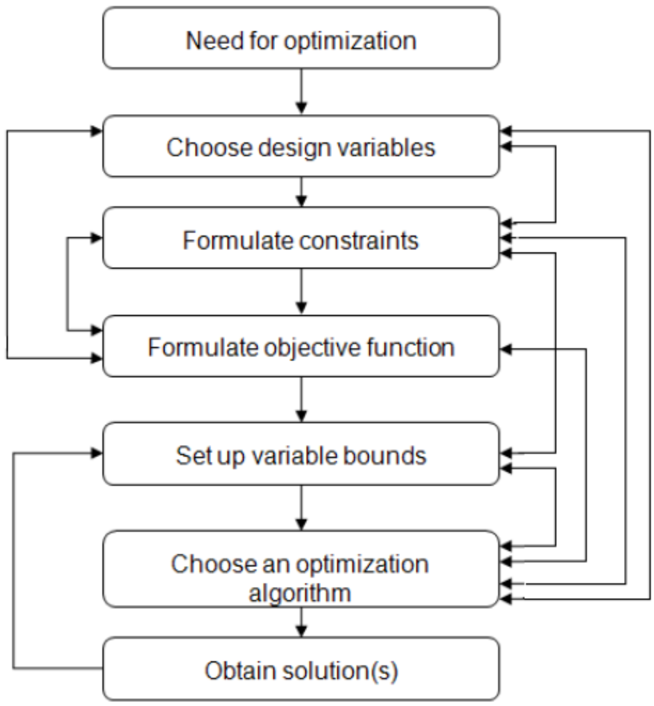


Figure 1.1: Steps involved in obtaining optimal solution

Optimization problems exist in all fields. To solve optimization problems related to engineering disciple which mostly includes designing of hardware components and circuits, planning and scheduling of production, quality controlling, providing maintenance and repairing of hardware equipment's [6], many optimization methods were proposed in past and proven to be beneficial for solving specific set of problems.

Definition of Optimization is represented in mathematical form, in equation 1.1:

$$Y = F\ (A1;A2;:::;An) \qquad \text{-(1.1)}$$

Here Y represents optimal solution and F represents fitness function which has been applied to obtain desired solution. This fitness function may be cost function, distance function, or any other objective function depending upon type of problem. The parameters A_1 to A_n represents variables associated with any problem and their values are required to be adjusted for obtaining optimal solution [6]. These parameters can fall under either category of decision variables, independent variables or control parameters. The authors of [6] and [189] have suggested the use of term 'optimum' is more appropriate and technical than word 'improved'. The term 'optimum' refers to quantitative measurement either in terms of 'minimizing' or 'maximizing' the objective function. Various optimization techniques/algorithms are proposed by researchers, in the past. Different algorithms inspired from swarm intelligence are mentioned in Table 1.1.

Few authors have categorized Optimization problem in two categories: Continuous and Discrete Optimization Problems [113]. A problem is said to be discrete optimization problem, when it has finite number of solutions to a given problem. In case of continuous optimization problem, solutions obtained for problem at hand, could be infinite. This research wok primarily focuses on providing solutions to continuous optimization problems. Further, Continuous Optimization Problems are categorized into two categories, i.e. single-objective optimization and multi-objective optimization [165]. It is obvious that motive of single objective optimization problem is to obtain outcome with respect to single-objective only, whereas multi objective optimization problem has to maintain balance among multiple objectives, to obtain the optimized result to problem at hand [84] [43] [78] [190]. Considering existence of multiple solutions to a multi-objective optimization problem [144], researchers have contributed the concept of Pareto Optimum Solutions [151] [17].

To describe single-objective optimization problems, further categories can be formed as either constrained or unconstrained. As name suggests, constrained optimization problems do consist of constraints and fulfillment of these constraints is a must, whereas unconstrained optimization problems need not to satisfy any constraints while providing solutions to problems and these type of problems seems to be less complicated as compared to constrained ones [165] [12]. Over the period of past four decades, many techniques have been put forward and implemented to unravel different kinds of problems [107] [165]. According to [151] [49] [107], to obtain optimal solutions, various linear and non-linear mathematical or programming methods have been adopted. The author of [107] has focused on usage of numerical optimization techniques to obtain optimal solutions, but implementation of such techniques in real world is a cumbersome task and unpredictable. As mentioned in [108] [198] [16], authors have suggested that numerical optimization techniques are incapable of solving any problem, where optimization is required, due to absence of gradient information and computational limitations. Due to drawbacks and limitations of numerical optimization techniques, as mentioned by [108], another aspect is to opt for either heuristic or metaheuristic approaches for solving optimization problems [108] [110] [49] [93].

In earlier days, optimization is achieved by using different 'traditional' or 'numerical' optimization techniques. Optimized Solution was obtained using Stochastic Programming, Hill Climbing, Constraint Programming, Goal based Programming, by assigning Weights to objectives, by applying Sequential Optimization techniques, Gradient based techniques and Linear Programming. Nowadays, nature of problems is much more complex and count of problems is increasing day by day. To provide solutions to different set of problems, numerous methodologies/techniques have been developed and adopted so far. Evolutionary Computation, Genetic Algorithm, Ant Colony Optimization, Harmony Search, Particle Swarm Optimization, Bat Algorithm, Firefly Algorithm, Cuckoo Search Optimization, to name the few, are gaining popularity due to their applicability in most of the application areas.

Despite of computationally extensive and without any guarantee of obtaining optimal solution, metaheuristic approaches are still preferred by many researchers for providing solutions to problems. Though metaheuristic approaches offer many benefits like ease of development and applicable to variety of problems [3]. Even the convergence rate of metaheuristic approaches is better than other optimization approaches [15].

Evolutionary optimization techniques are combination of a set of semantics and constraints, along with uncertain population. Evolutionary algorithms (EA) have capability to simulate natural characteristics of physical systems or biological systems

[3] [151] [133] [210] [16] [190]. Here physical system refers to Simulated Annealing algorithms and biological system refers to those algorithms which are inspired from either human biological system like neural network or animal behavior inspired like ant colony optimization and many more. Evolutionary algorithms offer many advantages over other algorithms. Priori the use of Evolutionary algorithms, previous knowledge of problem is inessential and is applicable to vast range of areas. In order to optimize the solution to any problem, one needs to describe objective function either explicitly or implicitly [83] [145]. An evolutionary algorithm relies on hit and trial method, keeps on updating solutions and also instruct search entities of population to maintain trade-off between exploitation of obtained best solutions so far and also keeps on exploring new solutions in order to obtain global optimum solution [83] [69] [56] [135]. Author of [1] has categorized evolutionary algorithms in sub categories. These sub categories include genetic algorithm (GA) which is proposed in 1975 and developed by Holland. In 1966, Fogel has introduced evolutionary programming. In 1992, Koza has proposed genetic programming. In 1995, Stom and Price has implemented differential evolution. Among these popular EA, few algorithmic techniques have attracted various researchers to contribute in this field.

Swarm intelligence based algorithms is one such category, which has astonished and inspired researchers across globe and also generate optimal solutions for problems related to almost every aspect of life. These algorithms operate as per collective behavior of swarm entities/population and share a complex interaction between individuals and their neighborhood. Here individual refers to ant, bat, honey-bee, bacteria, bird or fish [3] [109] [53] [8] [177]. One of the reasons that swarm gained popularity is due to their self- organization and decentralized nature. Entire population of swarm has to follow same rule, cooperate with each other, interact with each other to achieve common objective of either foraging or socializing [109] [8] [191]. Another remarkable feature includes presence of memory component, presence of multiple individual entities, continuous solution improvement mechanism and adaptable to environmental changes [53] [68]. In 1995, Kennedy along with Eberhart has devised PSO which depends on how flock of birds behave socially. Development of PSO is shadow of ACO which is developed in 1999 by Dorigo. This algorithm is based on idea, how ants follow each other while seeking a path to reach food source. Artificial Immune System is based on idea that how human body reacts when some non-body cells interact with body cells. This concept motivates Hofmeyr and Forrest and lead to development of an artificial immune system algorithm in 2000. In 2002, Passino has come up with an idea of imitating social foraging behavior of Escherichia coli while searching for nutrients using bacterial foraging optimization algorithm. ABC is considered to be among efficient algorithms that was developed by Karboga and Basturk in 2007. In 2008, Havens has started investigating how cockroaches behave socially and which leads to the development of roach infestation optimization (RIO) algorithm. Later in 2010, Xin She Yang got inspired from the way by which bats echolocate their targets and which lead to the formation of Bat Algorithm. Bats produce sound and listen to echo received to find their targeted prey. Considering applicability of Bat Algorithm, Xin She Yang has proposed another meta-heuristic approach, namely, cuckoo search [194], soon after firefly algorithm was proposed [195] that is related to flashing behavior of fireflies. To strengthen swarm intelligence techniques, researchers have opted for hybridization of one algorithm with another, incorporating biological features or inculcating conventional approaches [191] [15]. Here, focus is on Swarm Intelligence based Optimization techniques, which is sub-set of Bio-Inspired Optimization techniques, which in-turn is a sub-set of Nature-Inspired Optimization techniques.

$$\textit{SI-Based} \subset \textit{Bio-Inspired} \subset \textit{Nature Inspired}$$

whereas Physics based and Chemistry based algorithms are from the sub-set of Nature Inspired, but not from Bio-Inspired Algorithms.

$$\frac{\textit{Physics Based Algorithms}}{\textit{Chemistry Based Algorithms}} \begin{cases} \notin \textit{Bio-Inspired Algorithms} \\ \in \textit{Nature Inspired Algorithms} \end{cases}$$

Few algorithms from category of Physics & Chemistry Based Algorithms are Memetic Algorithm, Harmony Search, Shuffled Frog Leaping, Simulated Annealing, Big bang-big Crunch, Charged System Search, River Formation Dynamics, Stochastic

Diffusion Search, Spiral Optimization, Water Cycle Optimization Technique, Galaxy based Search technique, Black Hole Optimization, Gravitational Search, to name the few [75].

1.2 Single Objective Optimization Problems

Selection of appropriate optimization technique depends upon type of problem. The broad categories of problems are Single-Objective, Constrained and Multi-Objective optimization problems, as depicted in Figure 1.2. Description of such problems and appropriate optimization algorithm required for solving these types of problems are listed in subsequent sections.

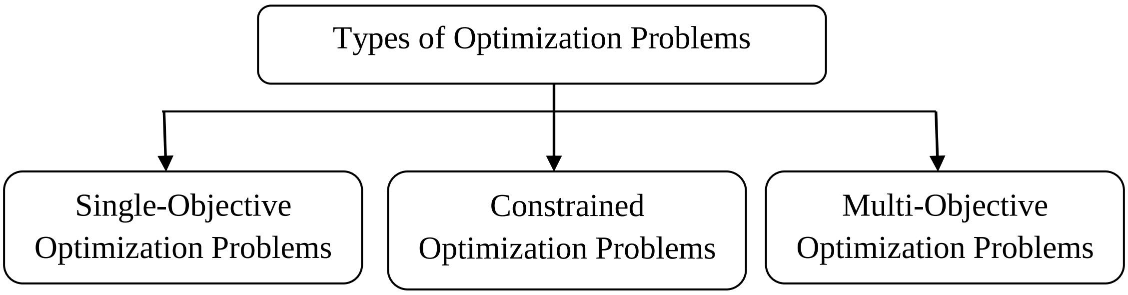


Figure 1.2: Categorization of Optimization Problems

1.2.1 Outline

A single-objective optimization problem is defined as combination of an objective function, which relies on ‘n’ numbers of variables, which are tied to minimum and maximum threshold value. These minimum and maximum threshold values are sometimes also referred as lower and upper bound of variables. The motive behind usage of any optimization method is to obtain result of problem, considering underlying parameters (variables) which generates optimum solution for function F(x). According to [189], solutions are categorized into two categories:

1. Local optima: A result obtained is represented as local optima if no solution exists in neighborhood, which is improved version of selected one. If objective function

is a 'minimization' function, then there should be no solution lesser than selected best solution.

2. Global optima: A result obtained is represented as global optima if no solution exists in neighborhood, in all magnitudes, which is better than selected solution. If objective function is a 'minimization' function, then there should be no solution, present in entire search space, which is lesser than selected best solution.

The author of [164] has mentioned about three techniques of solving single-objective optimization problems. These techniques are: Numerical Methods, Enumerative Techniques and Random Guided Techniques. Numerical Method makes use of local solution and sufficient condition for finding solution to a given problem [164]. Direct and In-direct searching approaches, belong to Numerical Methods. The problem with such techniques is that, these can be used to solve unimodal problems, but fail to solve real-life applications. Another set of techniques are enumerative techniques, which assess all solutions available in neighborhood to obtain optimal solution. To evaluate each and every solution, problem at hand is divided into smaller parts, having lower complexity and then optimal solution is obtained [164]. Third category of techniques for solving single-optimization techniques involve Random Guided Technique, which is an extension of enumerative techniques and includes additional information about search space to generate more optimal solution. These types of techniques can be further classified as single and multi-point search techniques. SI techniques are considered to be the part of evolutionary techniques and rely on multi point search techniques, which offers good exploration along with different set of parameters, depending on problem at hand [164]. In case of multimodal and discontinuous category of problems, swarm intelligence techniques help in obtaining approximate optimal solution, in a larger search space.

1.2.2 Approaches for handling Single Objective optimization problems

Generally, there are two approaches which are deployed to solve single-objective optimization problems, namely, gradient based methods and direct methods [189]. Direct approaches are dependent only on the value of fitness function in order to control search process and does not utilize derived statistics related to fitness function [189]. On contrary, gradient-based methods make complete utilization of first derivative or second derivative to refine search process. SI based techniques are nowadays gaining popularity for solving optimization problems, comes under category of direct methods. Numerous techniques of SI have been implemented in the past for providing results to either single or multi-objective optimization methods. In 2005, Yang has introduced artificial bee algorithm (ABA) that simulates behavior of honeybees. Artificial honey bee finds food by process of exploration in search space; returns to hive with honey collected; performs waggle dance to convey route information to other bees; other artificial bees calculate distance as well as direction from waggle dance; and follow same path to reach same food source. The author has done performance evaluation of this algorithm with respect to genetic algorithm (GA) and proved that ABA performs better than GA due to parallelism factor. Later, in 2007, Yang has done hybridization of particle swarm optimization (PSO) and genetic algorithm to provide solutions to single objective optimization problems. In the hybridized algorithm, flying strategy of particles and diversity among particles are improved using GA. This hybrid strategy works in two different phases. In first phase, implementation of PSO is done and in later phase, GA is used, which avoids untimely conjunction. The performance of hybrid algorithm is validated for solving single-objective optimization problems over Sphere, Rosenbrock and Rastrigin mathematical functions. This hybrid algorithm is proven to be beneficial over PSO and GA.

In [48], author has proposed an artificial bee colony (ABC) algorithm which is enthused from the nectar searching process carried out by artificial bees. Here, bees belong to same colony are divided into three sets; first set of bees search food source, another set of bees (known as onlooker bees) selects best food source and third set of bees (known as scout bees) search for food in search space randomly. In work [48], author has carried out comparison between ABC algorithm, GA, PSO and hybridized PSO-EA. Experimental results have shown that ABC algorithm have capacity to escape from being stuck in local optimal solution. ABC algorithm can also be used to provide solutions to multimodal and multivariate problems. In [176], researchers have developed roach infestation optimization algorithm (RIO), inspired from behavior of cockroaches. The authors have considered collective as well as individual behavior of cockroaches. This algorithm is based on three interesting behaviors of cockroaches, which include movement in darkest location, enjoying and socializing with other cockroaches and searching for food very frequently. According to [176], results has proven that algorithm has obtained global optima and can perform far better than PSO. The author of [195] got astonished from lightening feature of firefly has proposed firefly algorithm (FA). FA works on three elementary assumptions. The first is that gender of all fireflies is same, attractiveness depends upon brightness of light and fitness function relies on 'brightness' parameter. The results of FA are evaluated on benchmark functions, related to single objective optimization techniques and outperforms in terms of success rate and efficiency. In [194], author has developed another swarm intelligence-based algorithm, namely cuckoo search (CS) algorithm. This algorithm is grounded on parasitic behavior of cuckoos. This algorithm is hybridized with Levy flight behavior of fruit flies. This system is based on three rules; every cuckoo can lay an egg at a unit interval of time, randomly in any nest; nest having top superiority of eggs will be forwarded to next level; nests must be quantified. The CS technique has been proved against PSO and GA on the basis of single objective optimization benchmark functions and results proved that CS is better in terms of multimodal objective functions [194].

In [58], author has proposed an extension of ABC algorithm, which is Rosenbrock ABC algorithm, which is combination of Rosenbrock's rotational scheme to be implemented during exploitation, along with basic concepts of ABC algorithm used during exploration. Rosenbrock function is a derivative-free local search method, having adaptive search mechanism. ABC is such a SI technique which is stimulated from behavior of bee colony, which is involved in process of nectar searching. The author of RABC algorithm has stated in research work that proposed algorithm offered good performance level, when tested on convergence factor, accuracy factor, efficiency factor and robustness factor on different 41 optimization benchmark functions. In [10], author has proposed another nature inspired optimization technique, namely, Krill Herd which is stirred from krill's herding behavior. This algorithm relies on movement of each individual, their foraging behavior and randomly induced diffusion to determine next position of krill. This algorithm is tested over different 8 algorithm to prove its performance over 20 single objective optimization benchmark functions. Another author has proposed hybrid version of ant colony optimization and firefly algorithm for resolving single-objective optimization techniques. The reason behind hybridization is that ant colony optimization technique is good at exploitation and will work as global searcher. Whereas firefly algorithm is good at exploration, so it will perform tasks of local searcher. The author has evaluated performance of algorithm over different 15 benchmark functions, which are purely meant for single objective optimization techniques. The author has named this hybrid algorithm as ACO-FA. In [150], another variant of bee colony algorithm is proposed and named it as directed bee colony algorithm. This algorithm is inspired from collective decision-making process which is carried out by bees while finalizing nest location. Generally, population of bees is kept constant. Environmental conditions, information sharing procedure and the way bees perform tasks, constitutes the formation of another algorithm. The author has evaluated performance of algorithm over 9 unimodal and multimodal based functions. The experimental results have proven that it is superior in terms of accurateness and robustness over other nature inspired meta-heuristic techniques.

After inspiring from mating strategy of birds, author of [7] has proposed an algorithm in 2014, namely, bird mating optimization technique. The base of this algorithm is resultant of birds' mating behavior. Based on mating behavior, birds residing in same neighborhood are referred as five different groups. These groups are referred as: monogamous, parthenogenetic, polygynous, promiscuous and polyandrous. This algorithm is tested over 3 different kind of optimization benchmark functions, which are suitable for testing single objective-based optimization problems. Three different kind of optimization categories include unimodal, multimodal and low-dimensional multimodal. The results have suggested that algorithm is good at performing local as well as global search, in comparison to other algorithms. Bare bones particle swarm optimization with scale matrix adaptation (SMABBPSO) algorithm is improved version of original bare bones particle swarm optimization. This algorithm has improved problem of premature convergence. The author of [126] has suggested that every component present in search space makes use of multi-variate distribution, along with adaptation of scaling matrix. To imbibe accumulated learning feature in every component and to help these components in getting stuck in local optima, above-mentioned strategy helps in doing so. This algorithm is evaluated on 15 single objective optimization bench marking functions and results has shown significant improvement over its previous variant. Based on social network evolution model, author has proposed an algorithm in [187], namely, social network-based swarm optimization algorithm which is meant for solving single-objective problems. This algorithm aimed at improving performance of swarm and it is based on dynamic topology, scalable population, and vast neighborhood. To achieve finest results, associates of swarm are categorized into two categories, based on their fitness value. The algorithm is then compared with other seven algorithms, to prove its performance. In [188], author has proposed Bat Algorithm based on echolocation feature of bats. In subsequent chapters, detailed study is carried out, as Bat Algorithm forms the base for this research work and motivates to explore biological features of bats to develop more variants.

1.3 Constrained Optimization problem

1.3.1 Outline

Constrained Optimization problem is defined as combination of objective function, equality and in-equality constraints and lower bound & upper bound associated with variables. The work done by researchers in [114] [26] [57] have stated that constrained optimization problems are able to deal with interferences among multi variable and multi constraint characteristics. Solving these types of problems are hard in comparison to unconstrained problems [6]. To solve any constrained problem, it is crucial to fulfill underlying constraints [6] [68] [88]. Many researchers have used different methodologies to convert constrained problems to unconstrained problems. In order to satisfy constraints, authors of [6] [146] have suggested to include these constraints into objective. Another aspect of constrained problem is to maintain stability among feasible and non-feasible solutions, during entire search process [24]

[135]. Available solutions are categorized into two categories: feasible and non-feasible, depending upon their satisfaction to equality & in-equality constraints and lower & upper bound of variables. Those solutions which satisfy constraints and lie in range of variables, are said to be feasible solutions. Those solutions which fails to satisfy even a single constraint, are said to be non-feasible solutions [189] [24].

Conventional way of solving problem suggests to ignore the presence of non-feasible solutions and optimal solution can be obtained considering only feasible solutions [24]. From this, importance of constraints can be concluded i.e. to obtain optimal solution, constraints play a major role, while solving constrained optimization problems [119].

1.3.2 Approaches for handling Constraints

To obtain feasible solutions, proper implementation of constraint handling techniques is very necessary. In [53], it has been mentioned that constraint handling techniques play major role while finding feasible solutions. As per literature survey, various techniques are available to handle given constraints effectively and to obtain optimal solution [45] [162] [179] [191]. These constraint handling techniques are given below:

1. Usage of operators to prefer feasible solutions over non-feasible solutions.
2. Inclusion of penalty functions to transform constrained problem into unconstrained problem.
3. Use of multiple objective optimization concept such as Pareto ranking scheme.
4. Identification of factors which differentiate between feasible solutions and non-feasible solutions.
5. Selection of techniques which treats objective function and constraints in a different manner.
6. Hybridization of evolutionary techniques with numerical optimization techniques.

1.3.3 Approaches for solving Constrained Optimization problems

In last twenty years, various researchers have contributed by unravelling the concepts associated with constrained problems. In this section, approaches for offering solutions to such problems are divided into four main categories: swarm intelligence techniques, multi-objective techniques, evolutionary techniques and hybridized techniques. Many swarm intelligence methods have been used to solve constrained problems. Among all, PSO is considered to be favorable for many applications. In [100], author has modified existing PSO and named it as unified particle swarm optimization (UPSO) technique, which has integrated penalty function approach for handling constraints. This technique holds capability of good exploration and exploitation, without requiring extra function evaluations, while maintaining feasibility of obtained solutions. In [26], author has introduced another variant of PSO, namely, master-slave particle swarm optimization (MSPSO). In this algorithm, master swarm particles and slave swarm particles en route for better solutions and keeps on updating their information. Meanwhile, these master and slave particles also share their information with other particles. This approach helps in achieving better global solution and avoids being trapped in local optimal solution.

The author of [145] has formed two groups of swarms and devised a mechanism of communication between these two swarms. This algorithm is named as co-evolutionary particle swarm optimization (CPSO). The responsibility of both groups is to develop decision result and adapt according to penalty aspects for obtaining better solution. In [108], author has carried out research-based analysis with respect to simple constrained particle swarm optimizer (SiC-PSO), which is united with constraint handling mechanism. The algorithm is considered to be faster, consistent and effective, when local best solution and global best solution are used to update velocity. The author of [129] has proposed a fully constrained particle swarm optimization (FCPSO) and three types of partially constrained PSO (PCPSO) to manage water resources. These algorithms are improved versions of PSO, which lead to elimination of non-feasible solutions from search space. In comparison to standard PSO, these algorithms are considered to be computationally effective and does not get effective by initial swarm particles and population of swarm. In recent years, artificial bee colony (ABC) algorithm has gained popularity and which motivates other researchers as well to contribute in improvement of algorithm. The author of [68] has introduced penalty function with ABC algorithm to provide solutions to various engineering problems. Prior to this, the work done in [47] has opted Deb's rule for selection of mechanism which can be used to satisfy constraints. The author has proposed improved version to handle large scale constrained optimization problems, by introducing method for handling constraints [46]. Remaining techniques which are covered under umbrella of swarm intelligence includes Bat Algorithm, which relies on adjustment of loudness and pulse emission rate to achieve optimal solution [197]. Another algorithm bacterial gene recombination algorithm (BGRA) is stimulated from process carried out to resist effect of virus in bacteria [177]. Another algorithm, namely social spider optimization (SSO-C) algorithm, is related to supportive behavior among spiders, who belong to same colony [53].

Apart from above mentioned swarm intelligence techniques, various researchers have applied evolutionary algorithms to provide solution to constrained optimization problems. The author of [162] has hybridized an evolutionary algorithm with genetic algorithm which uses chromosome and homomorphous mapping between an n-dimensional cube and search space. The authors have claimed that hybridized algorithm is an alternative approach for solving nonlinear programming problems. Differential evolution (DE) is a widespread technique which comes under umbrella of evolutionary algorithms and mostly used to solve to constrained optimization problems. In [208], author has modified DE and came up with an idea of archived DE (ADE). In ADE, archive of all best solutions, which are obtained from last evolution, are maintained and used for computation of novel solutions. This algorithm is also combined with penalty functions and for calculation of fitness value of all available solutions. In [209], author has proposed an extension of DE which includes self-adaptive strategy. The motive behind combining self-adaptive strategy is to identify control parameters used along with various constraint handling techniques. Dynamic constraint handling mechanisms are used to improve available results and to fulfill objective function. The author of [185] has carried out survey on variants of constrained DE and proposed an improved mutation dynamic DE. This improved algorithm has incorporated rank-based mutation operator to improve convergence rate of DE and dynamic diversity among feasible and non-feasible solutions, available in search space using multiple trail vectors generation method. EA which are famously used for solving constrained problems, also include algorithm developed by [161], namely, adaptive segregation constraint handling evolutionary algorithm (ASCHEA). The purpose of algorithm is to maintain both feasible and non-feasible solutions. This algorithm also focused on improvement of constrained simplex method in order to improve convergence speed, as suggested by [182]. A genetic algorithm inspired penalty algorithm, which is holding the property of self-adaptive, is introduced to acquire information hidden in non-feasible solutions [24]. Another approach, effective global harmony search (EGHS), is proposed by [110], which is related to musical performance and applied to pressure vessel design problems, in order to obtain feasible solutions. Researchers have developed teaching-learning based optimization technique which is enthused from behavior of teacher-learner relationship and how teacher influences learner [156]. Another novel selection evolutionary strategy (NSES) is integrated with self-adaptive selection technique [114]. Mine blast algorithm (MBA) is such an algorithm which is inspired from explosion of bombs in order to clear mines field [16].

Many researchers have opted for hybridization of two or more techniques to provide solution to constrained optimization problems. The motive behind hybridization is to obtain new solutions which are better than existing solutions. As mentioned in [27] and [5], hybridization of genetic algorithm with another technique is done to enhance capabilities of genetic algorithm, while providing solutions to constrained optimization problems. The researcher of [27] has integrated co-evolution with self-adaptive penalty factors to improve fitness function of GA. Here, co-evolution is used to create two groups of populations, to determine type of penalty function and also to optimize solution. The process is considerably easy to device and is applicable in all such situations where parallelization plays a major role in improving general performance of function. The author of [5] has experimented with GA by injecting GA with a varied range of variables using a stochastic ranking method along with shifting and shrinking mechanism (SSM). The algorithm moves and shrinks search space which leads to fast convergence and in turn generates global optimal solution.

In [146], author has proposed an algorithm, which is combination of PSO and simulated annealing. SA is used to escape untimely convergence of algorithm and feasibility-based rule of PSO is used instead of penalty function, while applying constraint handling technique. The author of [205] has paired Nelder-Mead simplex method (NMSM) with PSO as NMSM offers benefit of exploration and PSO is mostly used for exploitation. PSO has also been combined with DE [69], in order to improve convergence rate. DE offers good search capability and usually used with PSO to get rid from stagnation. Deb has suggested the use of feasibility-based ruleset for comparison of solutions used in this method. [180] has proposed an approach, which is the combination of M & I and stochastic ranking scheme. This algorithm is used to balance dominant nature of penalty function along with objective function. The researcher has enhanced capability of algorithm by adding stochastic ranking mechanism to reflect the effect of search bias in constrained optimization problems [180]. In [179], author has investigated society and civilization-based algorithm which models interactions among intra society members and inter society members. This algorithm is combination of genetic algorithm, machine learning and Pareto scheme used for ranking. In [151], author has proposed cultural algorithm, which is a part of evolutionary computation, along with combination of differential evolution. This proposed technique makes use of field understanding to further expand performance. There are few more instances which includes dynamic stochastic selection used for multi member differential evolution proposed in [135], hybrid evolutionary algorithm and constraint handling approach proposed in [202] and differential evolution along with level comparison in [119]. There are few other approaches which are used for optimizing constrained optimization problems, which includes non-constraint handling approach. For instance, author of [29] has introduced niched-Pareto GA, in which new way of handling constraint is used for achieving multiple objectives. This method uses multi objective optimization approach, without using any penalty function to maintain diverse population. Later, the author of [55] has reviewed various non-penalty functions which offers self-adaptive mutation of multi population evolution. Multi population/member evolution strategy includes diverse function to separate non-feasible solutions, feasibility-based mechanism used to obtain feasible solutions and a recombination operator used during intensification phase. In [198], author has investigated GA that uses multi objective optimization technique, along with Pareto ranking scheme in order to deal with non-feasible solutions which violate constraints. In [201], researcher has proposed hybridized version of multi objective optimization and DE. This hybridized version has utilized non-feasible individual replacement technique to improve performance, by driving population towards feasible individuals. The comparison between feasible and non-feasible solutions is carried out using multi-objective optimization technique.

1.4 Multi-Objective optimization problem

1.4.1 Outline

Multi objective optimization problem is described as such a problem which includes satisfaction of more than one objective. Most of the times, there is a conflict in process of satisfying many objectives simultaneously. Existence of such result which fulfills all the objectives of a given multi objective problem is difficult. So, motive is to obtain such a solution which offers trade-off between the objectives to satisfy. The need is to select such a solution for any multi objective problem among other feasible solutions, which should satisfy acceptance criteria and should not dominate existence of other feasible solutions of search space. The collection of such solutions is said to be a collection of Pareto Optimality solutions. In 1881, Francis Ysidro Edgeworth has firstly introduced this concept and Vilfredo Pareto has modified it later in 1896 [28]. The two concepts associated with Pareto optimality includes: Pareto optimum and dominated & non-dominated points. There are so many ways to provide solutions to multi objective optimization problems. These ways are broadly categorized into two major categories: Pareto and non-Pareto techniques. Former comprises of pure Pareto, multi objective genetic, niched Pareto genetic, non- dominated sorting genetic, Pareto archived evolution and strength Pareto evolutionary algorithm. Non-Pareto based techniques are further categorized into various techniques, i.e. Vector evaluated genetic algorithm, e-constrained based, weighted sum approach and target vector approach.

1.4.2 Approaches for handling Multi-Objective optimization problems

Nowadays, Bat Algorithm and Particle swarm optimization are widely used for unravelling multi-objective problems [133]. These authors have carried out an extensive review, which revolves around approximately thirty different research works based on multi objective PSO (MOPSO). In [93], authors have claimed that they are the ones, who have modified PSO, meant for single optimization problems and for resolving multi-objective problems. They have used theory of p-vector, which is used to alter list of solutions, which generally keeps track of non-dominating solutions, in order to comply with rules of Pareto based techniques. The MPSO algorithm is applied for unravelling two multi-objective problems that are taken from literature survey by authors of [93]. The author of [105] has applied PSO for identification of Pareto optimum set and produced appropriate shape of Pareto front. The authors have hybridized multi-swarm based PSO with vector evaluated genetic algorithm and also integrated weighted sum approach for providing solutions to a multi objective problem [105]. They have evaluated performance of vector evaluated PSO technique (VEPSO) based on non-trivial multi objective optimization-based benchmarking functions and proven to be beneficial, as results of VEPSO were able to generate good set of Pareto optimum solutions. In [30], author has proposed MOPSO which uses Pareto dominance for obtaining optimal solution. Pareto dominance is applied to find out flight direction of a particle, whereas non-dominated vectors are kept for guiding flight behavior adopted by other particles. The results shown that performance of mentioned technique, namely, MOPSO is far better in contrast to PAES and NSGA-II, when compared on different categories of multi objective optimization problems, considering various mathematical benchmark functions [30]. In [132], author has applied Pareto dominance while developing MOPSO. This algorithm is based on three major factors. These factors comprise of crowding factor, mutation operators and e-dominance. Crowding factor is used as another form for discrimination criteria. Mutation operator is used for bifurcating population present in swarm. E-dominance factor is used to state upper bound of results present in concluding set. It has been noticed that this technique is capable to approximate Pareto front in a better way, when compared to other exiting techniques used for same purpose. The author of [199] has used the concept of PSO for providing solutions to a turning process involved in manufacturing industry. Neural network system is combined with PSO, which leads to formation of swarm intelligent neural network system (SINNS). SINNS is formed for defining objective functions, parameters associated and techniques to initialize these associated parameters. In [13], another algorithm is proposed, which includes criteria opted for defining velocity construction and then integrated with PSO for solving multi-objective problems. This proposed algorithm is named as speed-constrained multi objective PSO (SMPSO) and have evaluated performance on mathematical benchmark functions. The author of [121] has used global best and local best MOPSO for solving environmental and economic dispatch problems. In [85], concept of MOPSO is used for providing solutions to vehicle routing problem (VRP), by integrating improved dynamic lexicographic ordering. In [190], author has proposed an algorithm, namely, multi-objective Bat Algorithm for solving welded beam design problems. In [117], author has applied MOBA for solving problem of floor planning in VLSI. The floor planning problem revolves around minimization of wire length and minimization of dead space. The work done in [118] developed multi objective binary Bat Algorithm (MBBA), which relies on usage of Pareto dominance factor to discover Pareto solutions and to elect flight leader. Author has also suggested the usage of mutation operator for improving local search abilities of algorithm. This algorithm proven to be beneficial and outperforms non-dominated sorting genetic algorithm II. The author of [18] has come up with an idea of achieving optimization using grasshopper-based technique (MOGOA) and applied proposed technique for solving constrained and un-constrained multi objective optimization techniques. The proposed algorithm consists of three major algorithms, namely, Multi-Objective Particle Swarm Optimization, Multi-Objective Ant Lion Optimizer and Non-dominated Sorting Genetic Algorithm version 2 (NSGA-II). Out of these techniques, MOGOA proves to be beneficial.

1.5 Swarm Intelligence

1.5.1 Overview

Swarm Intelligence describes cumulative behavior of agents/entities, which holds capability of self-organizing, functions in decentralized manner and are distributed in entire search space. These agents/entities can be natural or man-made. In the year of 1989, a researcher, namely, Beni has coined 'Swarm intelligence' term. From that time, Swarm intelligence has led to development of several nature-inspired techniques, either used for obtaining the solution of the problem or for optimization of obtained solution. Swarm Intelligence techniques majorly focuses on metaheuristic approaches. The meaning of 'meta' is to look beyond and meaning of 'heuristic' is to find out solution by hit and trail method. In the nut shell, swarm intelligence inspired metaheuristic approaches are said to be 'high-level' approaches which can be used for exploring search space by deploying different techniques [32]. The metaheuristic approach is a type of heuristic approach which focuses on finding global optima from given search space either in an intelligent way or less [130] and termed as Stochastic approach of optimization. Stochastic optimization suggests selection of higher order approximate solution of global optima is of more importance than lower order of local optima, obtained using deterministic approach or conventional approach. In order to achieve higher order of fitness value, this approach always focuses on improvement of feasible solutions. First of all, it selects any feasible solution, say x_{solution}, randomly and search termination criteria. Then, it keeps on exploring neighboring solutions, till the time, a new solution is not obtained which is more computationally fit than the existing one. Mathematically, it can be represented as ($Select\ x_{neighbor} \in x_{\text{solution}}$). New solution is selected (IF ($x_{neighbor}$) < (x_{solution}) then update x_{solution} with new value of x_{solution}) and global optima obtained at last is $x_{opt} = x_{\text{solution}}$. Metaheuristic approaches very often rely on local exploration techniques and used in such applications where solution/search space is not explored in a systematic way or exhaustive manner. Any heuristic approach is characterized by the way of exploration carried out in solution/search space.

Examples of metaheuristic includes Particle Swarm Optimization [39], applied to solve problem of designing of Antenna's [138] and Electro-Magnetics [96]. Another metaheuristic approach includes Ant Colony Optimization and proven to be beneficial in many application areas, as suggested in [50] [151]. Artificial Bee Colony optimization technique has proven its applicability in solving numerical optimization problems and exhibit good performance [47] [48]. It has also been implemented in solving large scale optimization problems [76], combinatorial problems [76] [59] [152]. In recent times, new metaheuristic approaches are developed by researchers and added under umbrella of Swarm Intelligence. These swarm intelligence techniques include Cuckoo Search optimization [194], Firefly optimization technique [195], Wolf-Search optimization [154] and Bat Algorithm [188]. These newly developed optimization techniques comprise of search methods, which are broad in terms of breadth and depth. Moreover, these techniques rely largely on behavior exhibited by animals\insects belong to that swarm. The performance of such metaheuristic approaches is superior in comparison to many conventional techniques, used in past, for obtaining optimized solutions. Genetic Algorithm and Particle Swarm Optimization fall under such category [44] [92].

Two fundamental phases of any metaheuristic approach-based optimization technique include exploration and exploitation. Exploration can be accomplished using randomization to explore neighbourhood in order to obtain more optimized solutions. This randomization can be acquired by inclusion of random walks. Exploration offers diverse solutions, which indirectly helps any optimization technique to obtain global optimal solution and avoid getting struck in local optimal solution. On contrary, exploitation phase allows any metaheuristic approach to obtain new solutions present in the neighborhood, by traversing in the search space (locally) and to find improved solution than already selected optimal solution [59] [168]. The explanation of both phases: exploitation and exploration are given in next section, with respect to metaheuristic approaches.

Genetic Algorithm is such a metaheuristic approach which imitates behavior of natural selection, while obtaining solution to a given problem [91] [44]. It is one sub-category of evolutionary techniques, which is inspired from natural evolution process. It offers different operations like selection, mutation, crossover and many other natural evolution processes. As per terminology of Genetic Algorithm, all feasible solutions are said to be chromosomes and solutions are categorized as parent solutions and child solutions. Parent Solutions are used to generate child solutions either by performing mutation or crossover, but one at a time, to obtain more optimal child solution. Selection process is an umbrella activity and is carried out to shortlist solutions, whose features can be transferred to next generation of solutions. This leads to exploitation phase in Genetic Algorithm [111]. To provide solution to a stochastic problem, another type of metaheuristic approach is proposed by Kirkpatrick, Gelett, namely Simulated Annealing (SA). The advancements are introduced by [167] and further improved by [183]. To increase durability of metal, a metallurgic process is carried out. During this process, metal is heated and cooled under controlled temperature and environmental conditions. The author inspired from this, has proposed SA based optimization technique. In SA, temperature is the parameter which is used for evaluating fitness value of SA, during exploration and exploitation phase [42].

Particle Swarm Optimization, proposed in [92] is a stochastic optimization technique which uses population to obtain optimal solution. PSO is inspired from birds' flocking behavior, where each bird is acted as a particle. Similar to other evolutionary techniques, particles have ability to fly to obtain optimal solution. These birds fly using certain velocity, to obtain local and global solution. Selection of global solution relies on multiple local solutions obtained during different traversals in search space. In every iteration, a local solution is obtained, which is having minimum fitness value. This method is repeated for finite number of iterations and then best among all locally obtained solutions is obtained. The applicability of PSO has already been proven, when implemented in un-supervised robotic learning [95] tile manufacturing cum optimization [178], electromagnetics [60], wireless sensor networks [155] and many more. Another metaheuristic technique, namely Harmony Search, is proposed in 2001 and popular for solving various optimization problems, like water distribution-based problems [127], vehicle routing problem [107], time-table scheduling problem [122], numerical optimization problems [46] and many more. Harmony Search is inspired from harmonic sounds which sounds pleasant to human ear. Another metaheuristic algorithm is proposed and named as flower pollination algorithm [193]. The motive of this optimization technique is to obtain a similar optimal solution, as produced by musician during a perfect harmony. The way musicians adjust three major factors: adjustment of pitch, randomization and harmony memory, to produce sound, which is pleasant to human ears, in the same manner, this technique works [195].

1.5.2 Characteristics of SI

Different optimization technique possesses have different behavior and different characteristics, even though they all are inspired from nature. But few features are common in their behavior. Below given are characteristics of all such algorithms, basic steps, process of obtaining optimal solutions and dynamics of algorithms.

- All swarm intelligence-based optimization algorithms rely on population/agents. Here population/agent refers to bats, ants, particles, cuckoos, fireflies, and bees for Bat Algorithm, Ant Colony Optimization, Particle Swarm Optimization, Cuckoo Search, Firefly Algorithm, Artificial Bee Colony algorithm, respectively. Here each agent relates to a solution, present in search space. Among all solutions present in search space, there exists a global best solution, which is having either maximum or minimum fitness value. Maximization or Minimization of fitness function depends upon problem at hand. The solutions present are highly diverse in nature, which makes implementation of any optimization algorithm a necessity.
- Improvement in population is achieved by using different operators like random operators or mutation or using certain variables or formulae or equations, depending upon problem at hand. These types of evolutions are iterative in nature, which will lead to the generation of new solutions with different characteristics. The system deployed for optimizing solution will start converging, once solutions obtained so far, seems to be similar.
- Most of the algorithms work in two phases: exploration and exploitation, to obtain local and global solutions. If it is a local search, chances of trapped in local optima is much more. But, if the used optimization technique only focuses on obtaining global search, then very soon it will converge, which reduces chance of obtaining good optimal solution. Different strategies are adopted by different optimization techniques. Few have introduced randomization, whereas few have introduced

different strategies to maintain equilibrium between exploitation and exploration [31].

- The solution having maximum or minimum fitness value is reflected to best among all. This best solution is kept in population, to obtain another best solution in subsequent iterations. This type of selection mechanism drives the force among all best solutions to converge towards best solution, in an organized manner.

1.5.3 Selection of Optimization Techniques

As there exists many traditional optimization techniques and various swarm optimization techniques. It is an obvious question that comes to mind, before selecting the optimization technique for solving any problem, is that which type of optimization technique should be preferred over the other? Is there really a necessity of swarm intelligence-based optimization techniques? What was the drawback of traditional optimization techniques, which led to development of swarm intelligence-based optimization techniques?

The answer to all such questions is that, there is no major drawback of traditional optimization techniques. Research carried out by many researchers have proved that traditional optimization techniques are really good at providing solutions to various problems. But there are certain shortcomings of traditional optimization techniques, which are mentioned below:

- Traditional optimization techniques are mostly suited for obtaining local solutions. These techniques do not offer guarantee of obtaining global solutions. Moreover, the final solution depends upon initial seed value. Few exceptions from basket of traditional optimization problem includes linear programming and convex technique.
- Traditional optimization techniques are more specific to problems and that’s why sometimes referred as problem dependent optimization techniques. These techniques usually make use of derivatives related to local objective. It is not possible to solve highly multimodal and nonlinear problems. Another problem is that, such techniques faces a lot of problems with discontinuity, specifically when gradients are needed.

- Such optimization techniques are highly deterministic. Due to this, their exploitation capability is good, but lacks in exploration, which in turns lead to less diversification of solutions.

On the contrary, Swarm Intelligence-based optimization techniques offer advantages over traditional optimization techniques. To overcome disadvantages of traditional optimization problems, inclusion of population-based optimization techniques, along with non-deterministic and stochastic capabilities, is done, to improve exploration capabilities. The focus of SI techniques is to improve result obtained as global solution. The features and advantages of SI techniques are mentioned in next section.

- Most of the SI techniques are good at obtaining global optimization solutions. These techniques are usually gradient free and do not depend on derivatives. These types of techniques are highly suitable for solving non-linear and discontinuity problems.
- Here optimization techniques are treated as black box, which do not require knowledge of specific field and thus can be used to solve any type of problems.
- To enhance exploration capabilities, stochastic features are used along with swarm intelligence techniques. Inclusion of stochastic features, helps in avoiding being trapped at local solutions. There is no chance of remembering initial solution, thus reduces chance of recalling initial guess and does not hold knowledge of problem at hand.

Table 1.1 summarizes various swarm intelligence-based optimization techniques, which are explored in this research work. Along with above-mentioned advantages of swarm intelligence-based optimization techniques, they do hold some drawbacks.

Table 1.1: Summary of Swarm Intelligence Techniques

Optimization Technique	**Swarm based**	**Bio inspired but not Swarm based**	**Physics based**

Artificial Bee Colony	√		
Ant Colony Optimization	√		
Bacterial Foraging	√		
Bat Algorithm	√		
Bee Hive	√		
Black Hole			√
Butterfly inspired	√		
Cat based Swarm	√		
Charged system search			√
Cuckoo Search	√		
Dolphin Echolocation		√	
Eagle Strategy	√		
Electromagnetism technique			√
Firefly Algorithm	√		
Fish Swarm	√		
Flower Pollination Algorithm		√	
Galaxy based search			√
Genetic Algorithm	√		
Glow worm technique	√		
Gravitational based search			√
Harmony based search			√

Intelligent water Drop			√
Invasive weed optimization		√	
Krill Herd	√		
Lightening Attachment procedure	√		
Marriage in Artificial Honey Bees		√	
Monkey Search	√		
Paddy Field Algorithm		√	
Particle Swarm Optimization	√		
Queen Bee Evolution		√	
Shuffled Frog Leap		√	
Simulated Annealing			√
Squirrel Search	√		
Termite Colony		√	
Thermal exchange		√	
Weighted super position attraction		√	

One of the major drawbacks is that computational power of such optimization techniques is much more than traditional techniques, because number of iterations required to acquire optimal solution is higher. Due to stochastic behavior of such techniques, it is not possible to obtain same solution again. It must be executed multiple times, to ensure consistency and to obtain meaningful statistical data.

1.6 Bats' family

From many decades, existence and livelihood of microchiroptera and megachiroptera bats have attracted humans [120] [70] [125] [41]. Bats are considered to be one of the extraordinary and diverse species, from mammalian family. More than 900 species of bats, exist all across the globe and is almost constitutes one-fourth of mammalian family [123] [86] [70] [41].

Table 1.2: Sub-categories of Megachiroptera and Microchiroptera bats

Order	Scientific Names	Common Names
Megachiroptera	Pteropodidae	Mega Bats
	Rousettus aegyptiacus	Egyptian Fruit Bat
	Pteropus giganteus	Indian Flying Fox
Microchiroptera	Megadermatidae	False Vampire Bats
	Craseonycteridae	Kitti's Hog-Nosed Bat
	Rhinopomatidae	Mouse-Tailed Bats
	Hipposideridae	Old World Leaf-Nosed Bats
	Rhinolophidae	Horseshoe Bats
	Miniopteridae	Long Winged Bat
	Noctilionidae	Fisherman Bats
	Mystacinidae	New Zealand Short-Tailed Bats
	Thyropteridae	Disc-Winged Bats
	Phyllostomidae	New World Leaf-Nosed Bats

	Molossidae	Free-Tailed Bats
	Emballonuridae	Sac-Winged Bats
	Natalidae	Funnel-Eared Bats
	Vespertilionidae	Vesper Bats

Among these 900 species, every species holds unique behavior and livelihood, which makes them unique among all species from mammalian family [120] [128]. The bats' species are categorized in two categories, as mentioned above: Microchiroptera and Megachiroptera depending upon size. The smallest species among all is named as Microchiroptera, for example: bumblebee bat. The heaviest species among all is named as Megachiroptera, for example: Indian flying fox. The bumblebee bats can span wings up to 13cm and have weight of only 1.5g, whereas Indian flying fox bats can span wings up to 1.7m and have weight more than 2kg [86] [70] [41]. Table 1.2 specifies the species of bats exist under Megachiroptera and Microchiroptera category along with their scientific and common names.

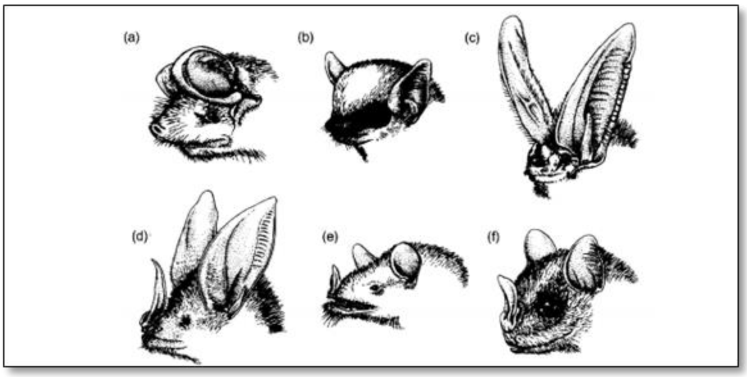


Figure 1.3: Sketches of designated Microchiroptera bats (a) Eumops underwoodi (b) Pipistrelle (c) Corynorhinus mexicanus (d) Mimon bennetti (e) Choeronycteris mexicana (f) Chiroderma improvisum [70]

Bats generally live together in large colonies, which includes approximately 1000 bats in common habitat and share roost [139] [163]. Bats living in same habitat is referred as a colony. Generally, bats belong to same colony occupy such locations, which are abandoned or caves or prefer locations with more darkness. Such roofs of a building are preferred which ends with a top limit of size between 0.73 to 0.99 inch width wise and 15.9 to 23.9 inches height wise [120] [128], depicted in figure 1.4.

Figure 1.5 depicts pictorial representation of a colony of bats' roosting. Generally, bats fly in dark locations. They start flying when environment starts turning dark. As per study carried out by [70], most of the bats from this species are those who prefer eating insects over other food, so called as insectivorous. Apart from such species, another category of bats do exist, who rely on fruits, small vertebrates, nectar and blood.

(a) Mouse-eared bat (b) Vampire bat

(c) Vesper bat (d) Pipistrelles

Figure 1.4: Glance at few species of mammalian bats

There exist four information transfer methods, which are adopted by colony members and these methods are mentioned below [120] and [86]: Colony of bats is depicted in Figure 1.5.

1. Intentional signaling includes calls related to mating process, alarm calls to inform peers in case of danger, territorial calls and foraging calls.
2. Local enhancement includes such calls which are produced unintentionally while guiding other bat of colony to a definite location of habitat.
3. Social assistance, which increases the chances of obtaining good food, by integrating cluster hunting behavior.
4. Imitative wisdom includes such bats, which acquire different hunting methods from other bats.

Figure 1.5: A colony of bats

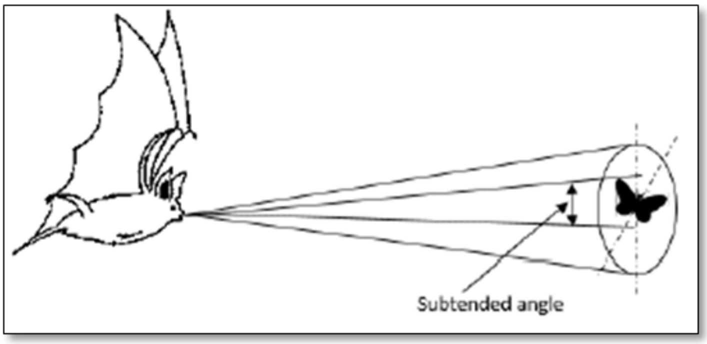


Figure 1.6: Formation of angle with respect to the emitted sound of bat

1.7 Biological behavior of bats

Many zoologists and researchers have studied biological sonar or echolocation of bats [80]. Apart from bats, various groups of animals do exist, who possess echolocation behavior, for example: shrews and tenrecs [120] [125]. Lazzaro Spallanzani has carried out study of such behavior of bats in 1794 [120] [87]. 'Echolocation' term is coined by Donald Griffin in 1944. The meaning of term 'echolocation' is to represent capability of bats to produce sound, which return echoes as well. The frequency of such signals is beyond range of human hearing. This echolocation is basically used for navigation purposes in dark locations or during night [120] [70]. Echolocation comprises of ultrasonic pulses, which can be frequency-modulated (FM) or constant-frequency (CF) or both [86] [101] [87]. Which type of signal will be produced by bats, depends on type of information bats' want to obtain about the environment. FM signals are best for determining target distance. CF signals are better for long range detection and for detecting target motion. To prevent pulse-echo overlap, bats shorten duration of FM signal produced. Pulse emission by one bat, in the presence of another, leads to interference of sound waves and echo produced by both bats. To avoid this interference, bats comes out from vocalizing phase and enters silent phase. Shifting from vocalizing phase to silent mode depends on relative spots of bats. Without actively echolocating, silent bat can acquire information about location of other bats and objects and can also avoid collisions, by listening passively. Some bats make use of tongue to produce sounds and emit short pulses using either mouth or nostrils [86] [87] [41] as depicted in Figure 1.6. When produced sound strikes with any object on path, it reflects back as echo [136]. In [141], author has stated in his research work that received echo follows Doppler shift, which means that frequency of received echo is much more than frequency of sound produced by bat. Bats do have capability to identify object and can also determine range between itself and object. The bats are able to acquire this information, by calculating time of reflection of modulated echoes [86] [136] [141] [41]. As per study of [86] [19] [136], while capturing prey, bat undergoes three phases. These phases include: searching phase, approachable phase and termination phase. During initial phase, bat starts capturing target, bat emits sound at very low frequency range, generally 10Hz [86]. In next phase, i.e. approach phase, bats detect location of target and attempt to get nearer to target. During approach phase, bat produces sound at such a frequency, so that overlapping of sounds can be avoided [86] [80] -as depicted in Figure 1.7. Shorter pulses can be produced by reducing time gap between sounds produced and received echo [86]. Even at this point of time, rate at which pulse is emitted keeps on increasing, as bat is moving closer to prey [86]. In [80], author has mentioned that pulse emission rate increases as bat needs to show more signs in order to trace exact location of prey. This is because angular position of prey changes so erratically, due to nearer distance between bat and target. During terminal phase, bat emits pulse at much higher frequency, sometimes even more than 200Hz and rate of pulse emission gradually increases with respect to fraction of milliseconds. This immediate increase in pulse emission rate happens just before capturing of prey. Bats have capability to avoid overlapping of sound produced and echo received during approach, target and terminal phase of prey capturing.

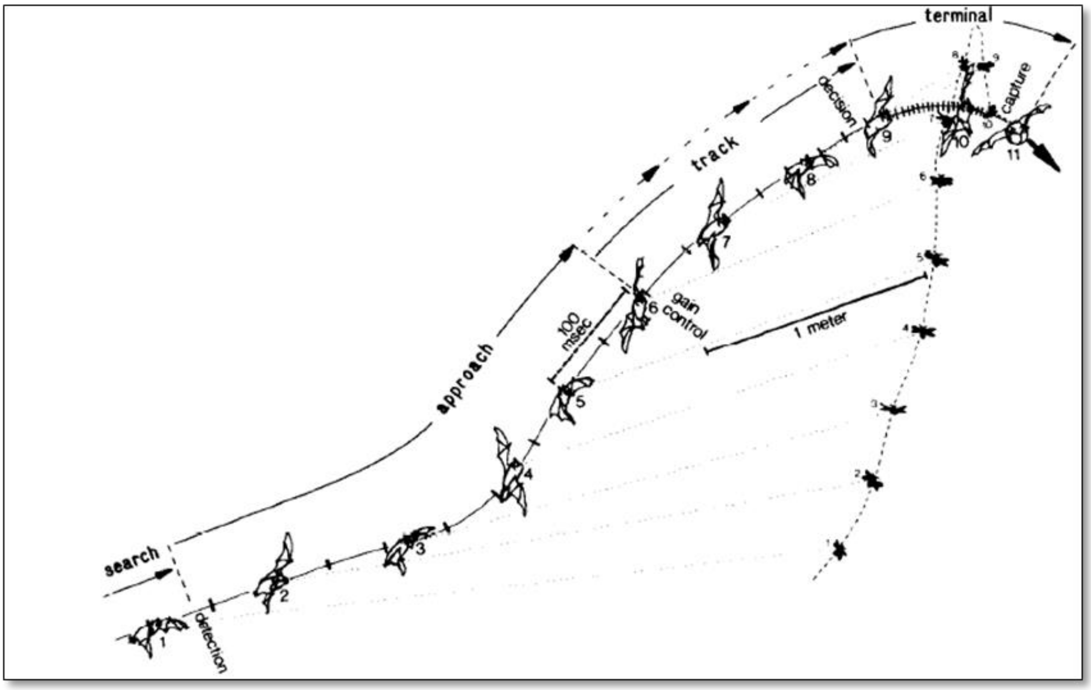


Figure 1.7: Phases of capturing prey [86]

Bat search space depends upon search cone angle, made from bat's mouth. Larger the angle, greater the search space and hence increases exploration. Bats are not only able to identify different targets with the help of echolocation, but also identify diverse parts of multifaceted object even located at far location. Bats are able to calculate jitter in echo delays before deciding target of interest. Three dissimilar pursuit behaviors are described of bats. Bats can compute relative position of bat with respect to conspecifics, along with details of angle between two bats. How bat distinguish a signal, even in presence of background noise? Source Initiation point, Strength of target, Communication Losses, Noise Level, Directional Index and Directional Thresholds are major elements for separating signal from noise [22]. Based on differences in echo spectra, shape and type of target can be discriminated. Moreover, information about object size, shape and surface properties are coded in temporal and spectral elements of echo structure. In [25], researcher has observed behavior of bats, which they adopt in order to avoid collision among themselves, while capturing prey. They adopt different pulse rate, in terms of frequency range, type of sound produced or sweep time course. Moreover, every bat follows a unique time structure, so that only bat produced sound should be able to interpret received echo [25].

From last so many decades, echolocation is considered to be one of the best characteristics that bat holds to detect position of target, to capture target and to select best target, in presence of multiple target's depending upon energy level of different targets [101] [80]. Bats belong to same colony, do share food, which is collected using echolocation [116] [63]. Bats favor each other, especially who belongs to same colony. For example, vampire bats share blood meals among themselves, so that energy level of all members of a colony should be at same level [116]. Research carried out by [63] has discovered that reciprocal altruism behavior exists among members of colony and fitness value of recipient bat is elevated with respect to non-recipient bat. This reciprocal altruism can takes place during common nursing or partnership formation in primates and backing behavior in cetaceans [63].

1.8 Bat Algorithm

Bat Algorithm (BA) is amongst those SI techniques which is proposed by [188]. This algorithm is enthused from echolocation behavior of bats, in order to identify target. Bat creates 3-dimensional picture of neighborhood by emitting pulse and calculates distance on the reception of echo. For calculation of distance between itself and prey to target, bats relies on time delay factor and varied intensity of sound. Received echo actually helps bat in determining not only location of prey, but also helps in determining type of prey, speed at which prey is moving and orientation of target/prey. For better functionality of algorithm author of this research has laid down certain ideal rules, which should be followed while using BA for solving any kind of problem. These ideal rules are mentioned below [188] [19]:

1. Bats will utilize echolocation feature in order to determine the range between themselves and prey.
2. Bats will differentiate between type of food source/prey and obstacles, using echolocation.
3. Bats initialize their parameters like velocity, position, frequency, upper bound of frequency, lower bound of frequency, pulse emission rate and loudness with random values.
4. Bats will adjust associated parameters while targeting prey, considering proximity of prey.
5. Lower bound of loudness factor is represented as A_{min} and upper bound is represented as A_0.
6. The concept of Ray tracing is not applied in estimation of time interval for determining range between prey and bat.
7. Upper bound of frequency can be selected depending upon type of problem at hand and which suits domain of underlying problem.
8. Another assumption laid down by the researcher is that distance between bat and prey is calculated in ‘magical’ way.
9. To represent lower and upper bound of pulse emission rate, 0 and 1 are used; where 0 represents absence of emitted pulse and 1 represents presence of emitted pulse.

Bat Algorithm is represented in form of pseudocode and represented in Algorithm 3.1. Here, [188] has updated velocity v_i, frequency f_i and position x_i of bats' in d-dimensional search space. In pseudo code, x_t^i represents new solution obtained at position 'i' at time 't', v_t^i represents new solution obtained having velocity at time 't', x_* represents global optima obtained so far, after evaluating all feasible results. The flow diagram of Bat Algorithm is depicted in Figure 1.8 and step by step procedure is explained below:

Step 1: Set the number of bats required, along with their parameters, including frequency, position, velocity, pulse emission rate and loudness.

Step 2: Assign initial values to above mentioned parameters, along with minimum frequency and maximum frequency.

Step 3: Repeat Step 3 to Step 10, till maximum number of iterations are not reached.

Step 4: Compute new solutions by varying the parameters.

Step 5: In case any randomly generated pulse rate is higher than pulse generated by existing bats, then execute Step 6, else execute Step 7.

Algorithm 1.1: Bat Algorithm

Initialize position x_i, velocity v_i, frequency f_i, pulse emission rate r and loudness A

while count < Max_Iterations

Compute the fitness value of each solution

Select the "minimum" fitness value as the best solution

Explore new solutions around the selected best solution, by adjusting f_i, v_i and x_i.

$f_i=f_{min}+(f_{max}-f_{min})*\ rand$

$v_i(t)=v_i(t-1)+(x_i(t)-x_*)*f_i$

$x_i(t)=x_i(t-1)+v_i(t)$

Narrow down the search space; explore the search space in the nearby areas of best selected solution

if (rand>r_i)

Generate the local solution around the selected best solution

end if

Generate a new solution by flying randomly if (rand <A_i && $f(x_i)<f(x_*)$)

Accept new solutions and increase r_i and reduce A_i.

end if

Rank the bats and find the current best solution.

end while

Figure 1.8: Pseudocode of Bat Algorithm

Step 6: Select best solution among all and try to compute another local solution around the same to avoid trap in local optimal solution.

Step 7: Compute any other solution, using random search.

Step 8: In case, randomly generated loudness is less than the loudness of any other bat and frequency is also lesser than the frequency of best bat, then follow Step 9.

Step 9: Accept that solution and keep on increasing the value of pulse emission rate and decreasing the value of loudness.

Step 10: Among all the bats, select the best bat and process the results.

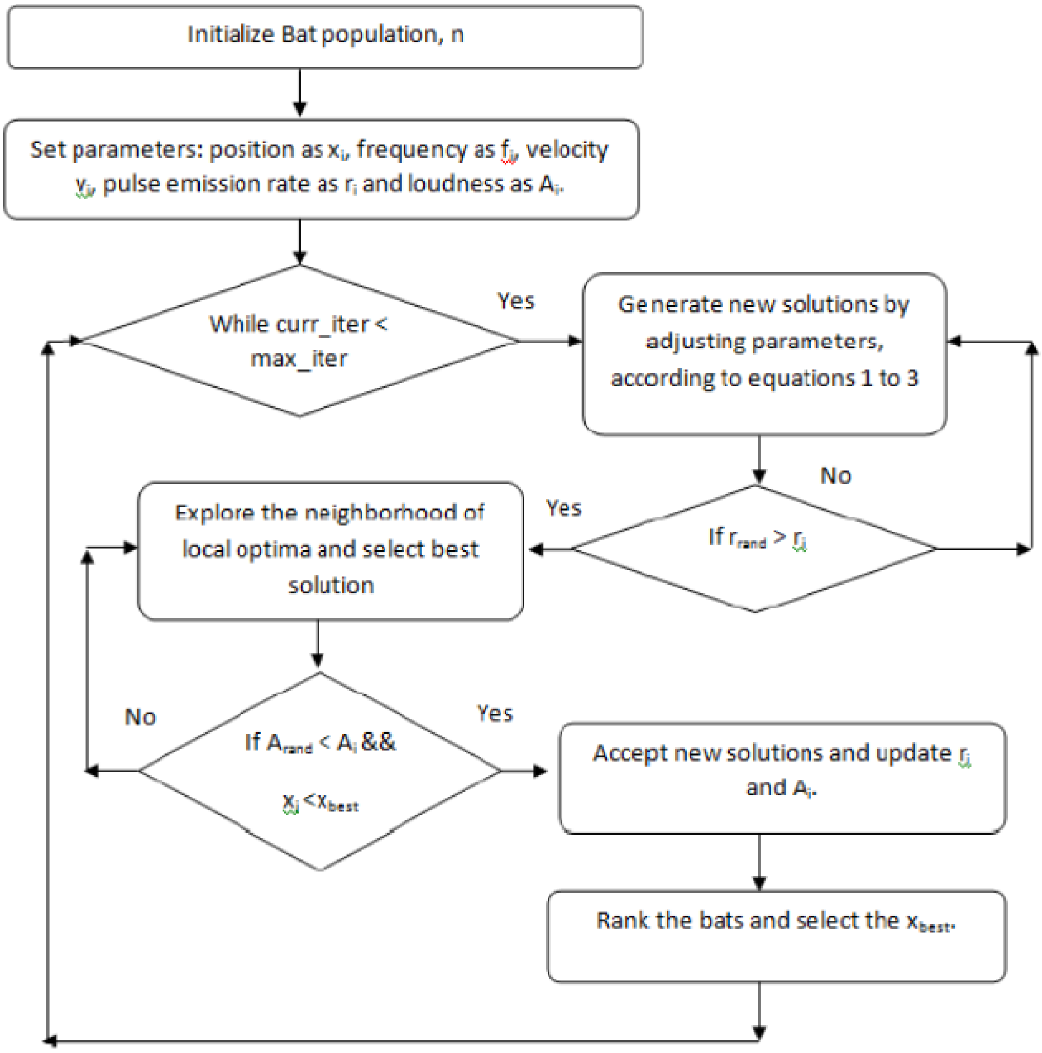


Figure 1.9: Flowchart of Bat Algorithm

The value of frequency f_i gains important as it helps in controlling the pace and movement of bats [188]. Whereas, values of f_{max} and f_{min} have been pre-defined during early phase of algorithm. The value assigned to f_{min} is 0 and value assigned to f_{max} is 100. The value assigned to frequency of every bat lies in the range of f_{min} and f_{max}. On contrary, values of these parameters also relies on kind of problem and domain size of the problem.As per author of this algorithm, bat can explore new position (solution) by walking randomly in search space and makes use of current best solution before moving to next best solution, as per the equation 1.

$$xnew = xold + eAt \qquad -(1)$$

where, e= [-1;1] is a random number and At =$<At^i>$

Here A_t represents mean loudness value of all bats present in search space at time 't'. Generally, whenever bat approaches its prey, loudness factor (A_i) keeps on decreasing but pulse emission rate r_i keeps on increases.

At initial step, every bat present in search space, have been assigned random value of pulse emission rate and loudness. Along with increment in iteration number, newly obtained solutions become better, and pulse emission rate and loudness will also get updated using equation (2) and (3). For instance, if we consider value of A_0 =1 and assume that bat will move towards prey and will stop producing any sound, once A_{min} become 0. On contrary, if r_0 is initialized with value 0 and considering r_{max} = 1, bat will keep on increasing its pulse emission rate once it reaches near to prey. From this, equation (2) and (3) are derived:

$$A^{t+1}_i = \alpha A^t_i \qquad \textit{-(2)}$$

$$r^{t+1}_i = r^0_i\,[1-\exp(-\beta t)] \qquad \textit{-(3)}$$

where $\alpha = \beta = 0{:}9$

Bat Algorithm has been implemented and tested over various benchmark functions. In all such cases, population of bats (n) lies in the range of 25 and 50. Further, author of this algorithm has equated performance of BA with GA and PSO, considering number of function evaluations by fixing tolerance level. As per outcomes of evaluation, BA is proven to be more precise and effective when equated with GA and PSO algorithms.

1.9 Broad division of Bat Algorithm's variants

X.S.Yang got motivated from echolocation feature of bats, back in 2010 and implemented concept of echolocation for optimizing solutions. Thereafter, various researchers have contributed in this field, by developing various variants and offered optimized solution to many engineering problems. In all subsequent versions/variants, researchers aimed and tried to improve Standard Bat Algorithm in one or the other aspect, either by hybridizing with other existing meta-heuristic techniques or by using different methods for initializing BA parameters, updating existing parameters or by developing discrete versions of Standard Bat Algorithm. All these advancements are introduced in standard BA, to improve performance and to make it suitable for solving different types of problems.

1.9.1 Improved version

After introduction of BA in 2010, research works have contributed towards performance improvement of standard BA and increasing scope of implementation. Initially, the author of [190] has applied Standard BA to solve non-linear problems. The proposed algorithm attained better results in comparison to existing meta-heuristic techniques. In [89], author has proposed a modified version of standard BA and entitled it as evolved Bat Algorithm (EBA). The authors have carried out process of analysis and redefined behavior adopted by bats of same colony. The proposed algorithm enhances accuracy level of obtaining optimal solution and also reduces computational time, while solving numerical optimization problems. Moreover, author of [190] has extended his own original work of BA for solving multi-objective optimization problems. The author has implemented multi-objective Bat Algorithm (MOBA) and tested it over welded beam design engineering problem. Afterwards, author of [9] has applied BA to solve constrained optimization problem. When BA is compared with other meta-heuristic techniques for obtaining optimal solution, it is proven to be far better than existing techniques, which are applied to solve similar problems. In [90], author has integrated concept of chaotic maps and Levy flight to produce more optimized solutions, in comparison to solutions generated using Standard BA. This research work focuses on improving search operation and to maintain equilibrium between exploration and exploitation process. In [89], author has stated that mentioned technique is reliable in estimating parameters for re-construction of dynamic biological system. Moreover, in [90] concept of Levy flight and chaotic dynamics for estimating parameter values in non-linear dynamic systems has been incorporated. Experimental results of proposed algorithm proven to be superior over other approaches [90]. In [124], author has used sonar of bats to develop an algorithm and named it as Bat Sonar Algorithm. In another research carried out by Yang, emphasis is given on intensification and diversification of Bat Algorithm, initialization of parameters and improvement of convergence performance. Many fields are yet to be explored which includes time delay estimation between emission of pulse and echo received by bat. Directional echolocation and Angular information fields can also be explored to propose a new variant of Bat Algorithm. In basic Bat Algorithm, position of bat updates, along with frequency and loudness, but position of prey is assumed to be constant. But, in real world, position of prey also gets updated with time. In [64], author has used concept of conditioned reflex to maintain foraging history. In [140], relationship between number of bats and operation to be formed, is found. The performance is increased and decreased along with increase in count of bats for 2-bit OR and E-bit XOR/4-bit XOR operation respectively. In Standard Bat Algorithm, self-adaptive compensation of Doppler Effect is not considered. So, work mentioned in [186], focuses on self-adaptive search strategy. To avoid being trapped in local solution, randomization approach of initialization of parameters is used. Updating loudness, pulse emission rate and compensation rate will affect performance of algorithm. This algorithm has yet to prove its efficiency in large scale optimization problems. In [112], diversification of population and optimization performance of Bat Algorithm is enhanced using real and imaginary part separately of complex valued encoding. In previous variants of Bat Algorithms, ability to learn from previous experiences is missing. In work of [115], author has used concept of double sub-population Levy flight to introduce this ability to Standard Bat Algorithm. Greedy algorithm is used here. Greedy algorithm fails to find global optimal solution. Moreover, it is applicable only for unimodal applications, not for multimodal applications. In [11], chaotic BSO, linearly decreasing function is multiplied by chaotic map function to update loudness. In [142], Bat Algorithm is used for optimization of weights and structure of neural network. This approach is useful in unravelling other real-world problems like pattern recognition. Moreover, concept of chaotic maps can be used to boost performance of algorithm. In [54], meta-heuristic approach introduced in 2010, by Xin.She Yang is applied to community detection problem. To avoid being trapped into local optimal solution, best among top ‘n’ bats is selected and for representation of solution, locus-based adjacency encoding scheme is used. Limitation of basic Bat Algorithm lies in diversification of all possible directions of search space. Due to that, success rate of BA is less. In [200], Doppler Effect on frequency shift is considered for updating frequency of artificial bat during exploration and exploitation. Concept of refined search and divers are considered to attain better solution. In [9], author has focused on frequency alteration and maintaining dynamic balance between exploration and exploitation. In future work, author has suggested to incorporate directional echolocation and the concept of Doppler Effect. In [4], author has suggested involvement of mutation and crossover operator to increase effectiveness of search and to speed up convergence rate of basic Bat Algorithm. Mostly, initialization of parameters like frequency, wavelength and position is done randomly. In this work [66], author has used chaotic sequences for initialization. Gauss Map is used to initialize frequency and loudness and to initialize pulse rate, Tent Map is used. Balance between different search capabilities, dispersion of solutions and more focus on exploitation are three advancements introduced by author in [172]. In all variants of Bat Algorithm, motive of researcher is to either increase convergence rate or to escape from getting stuck in local optima or initialization of parameters, to enhance performance of algorithm. In [85], author has used fuzzy logic to initialize parameters of Bat Algorithm. In [36], author has replaced poorer solution present in a subgroup with the best solution present in neighbor subgroup of search space, after execution of each iteration. This characteristic make algorithm suitable for applications which require parallel processing. Generally, population size is directly proportional to computational time. To diversify search space, simple restart is beneficial rather than opting for longer iterations intense search. In work of [97], concept of Differential operator and Levy flight Trajectory is applied to improve performance of Bat Algorithm. Differential operator is used to update frequency. This work has improved convergence rate of Standard Bat Algorithm.

1.9.2 Hybrid version

From last decade, hybridization plays a major role in optimizing solution in much more effective way. Various researchers have hybridized Standard BA with other techniques or algorithms, to optimize solutions. Integrated K-means algorithm, generally used for clustering, with BA for increasing the efficiency while clustering data sets of huge volume for the analysis of data and named it as K-Means Bat

Algorithm (KMBA) [61]. KMBA is not only able to achieve higher level of efficiency during clustering process but also minimizes requirement of computational resources and time required for computation. Apart from this, the author of [103] has incorporated pros of using BA to overcome shortcomings of fuzzy c- means algorithm (FCM). FCM is considered to be sensitive while initializing configuration and easily trapped in local optimal solution. Hybridized BA with differential evolution (DE) techniques [74]. Integration of BA with DE has increased ability of standard BA and also revealed motivating results when evaluated over mathematical benchmark functions. The researcher has utilized the same method of incorporating mutation scheme with BA and named it as differential operator, which is one of parts of DE algorithm and Levy flight trajectory [97]. This hybrid version aims not only to improve accuracy level but also improves convergence rate. Results have shown significant improvements in this hybridized version of algorithm and used for better quality of estimation, specifically advanced dimensional space. In [65], author has proposed concept of robust hybrid optimization technique by combining features of harmony search with characteristics of BA. Integrated harmony search factors are one of the operators for updating process of BA [64]. By adjusting pitch value, this hybrid technique has shown promising results, which has speed up rate of convergence, while solving global numerical optimization problems. In [181], PSO is hybridized with Bat Algorithm. To attain improved results, after each iteration, worst elements of PSO get replaced with best elements of BA and vice versa. Back Propagation Neural Network, identifies errors present in hidden layer by calculating errors of outer layers. In order to find location of nodes in wireless sensor network and to remove above-mentioned drawback of Bat Algorithm, author in [160] has suggested hybridization of Bacteria Foraging Algorithm with Bat Algorithm. In future, horizontal and vertical angle information can be included. To improve performance of Bat Algorithm, author in [173] has suggested some modifications. Linear decreasing inertia weight factor is familiarized to control exploration and exploitation. To enhance the exploration capability, artificial bee colony technique is applied.

1.9.3 Direct application

As BA has gained popularity among numerous researchers, Yang has become center of attraction and motivated other researchers to contribute in the field of optimization by proposing new techniques for solving single-objective & multi-objective optimization problems. In [102], researcher has utilized swarm intelligence algorithm along with fuzzy logic. This algorithm has been used for screening process of workspace offered by company having high ergonomic risk in a very short span of time. Another research related to field of ergonomic is carried out by [158] and has implemented BA. Here, each bat corresponds to every possible solution that can exist for skeletal of human body to identify optimal posture of human being. For mechanical engineering field, BA is proven to be beneficial. For instance, [174] has modelled an industrial gas turbine using BA method. Even, to monitor performance of thermal systems, BA based models can be used. The author of [23] has estimated amount of emissions emitted by fossil fueled power plant, using BA. Many applications have embedded BA with manufacturing processes, for example: data warehouse data, de-duplication of records [33], multi-stage hybrid flow shop scheduling [131] and multi-stage multi-machine multi-product scheduling [143]. In application areas related to electrical and electronics field, BA has been used in past to optimize solutions. For example, optimization of brushless DC motor wheel [175] and optimal location of capacitor placement [184] and many more. Moreover, researchers have linked implementation of BA for detecting phishing websites [148], training neural network of eLearning processes [104], classifying data sets associated with microarray’s [166], feature selection methods [52], path planning for combat air vehicle [98], optimization of topology [196] and image matching problem [98]. In [171], localization error is minimized by using range-free algorithm, i.e., mobile anchor positioning. To achieve optimization, meta-heuristic techniques are applied. This work is not suitable for underwater and indoor setup of ad hoc network. For this, natural extension of bat can be explored, because bat and frog have capability to identify location and type of object based on water ripple information. In [106], author has applied Bat Algorithm for speech enhancement and proved that Bat Algorithm outperform PSO in improving quality of speech. In [14], author has used concept of only two bats to attain optimal solution. These two bats exchange their responsibilities of exploration and exploitation from time to time. Movement direction of bat is considered here. Till now, planning of training sessions for sports is done by the sports coach. But, with the advancement in technology, nowadays, heuristic and meta-heuristic approaches have found applicability in this area as well. In [77], author has cited use of Bat Algorithm to plan training sessions for sports person, based on the one' physical and mental strength, recorded via sports watch. The motive of the research in [40], is to find the best system configuration while minimizing the loss rate. To achieve this, there is a need to select proper state of switches. Here, optimal selection of state of switches is accomplished using Bat Algorithm.

1.9.4 Discrete version

In [99], author has transformed continuous space into binary space, with the help of sigmoid function. In order to increase search space and convergence performance, complex number encoding is used. Initialization of all parameters, except bat position, is assigned randomly, in order to apply binary Bat Algorithm to graph coloring problem. The key focus of [67] is to color each vertex of graph with constraint that no two adjacent vertices have same color. RLF algorithm is used to initialize bat positions. In another work, concept of complex numbers is used to represent one dimensional number line, which enhances diversity and performance of Standard Bat Algorithm. Moreover, Cartesian coordinate system can be used to represent one dimensional number line using three-dimensional number line. In [168], author has applied concept of binary Bat Algorithm to real world application of optical engineering. In [157], based on natural behavior of real bat, feature selection technique is proposed. Binary Bat Algorithm, based on sigmoid function, used to maximize accuracy of algorithm, feature selection technique is used along with echolocation behavior of bat. In [170], to schedule workflow in cloud computing environment, binary Bat Algorithm is used. Resources are assigned to tasks in such a fashion, so that overall cost of workflow is minimal.

1.10 Research Aim and Objectives

This research focuses on resolving issues which are faced by various swarm intelligence techniques and simulating biological behavior of bats. This work primarily focuses on improvements and advancements to be introduced in Bat Algorithm. It is important to mention here that, it is not the aim of this work, to investigate existing swarm intelligence techniques and find out which technique has outperformed other prevailing techniques while solving diverse problems. However, it focuses on development of novel variant of Bat Algorithm, inspired from the echolocation and flight behavior of bats. This newly developed technique can then be used for solving different problems.

Below mentioned are the three main objectives of this research work:

1. To develop an approach for tracking targets moving at predictable speed using Constant Bearing strategy.
2. To develop an approach for tracking erratically (unpredictably) moving targets using Constant Absolute Target Detection strategy.
3. To design movement strategy for a target seeker in the presence of multiple target seekers.

1.11 Deriving new variants of Bat Algorithm

To improve performance of Standard Bat Algorithm, researchers have contributed a lot in the past. Still, it is attracting attention of many researchers. The motive of this research is to study biological features of bats and to incorporate those biological features to develop more variants of Bat Algorithm, which are definitely improved versions of Standard Bat Algorithm. The improvement can be observed either in convergence rate or accuracy or number of iterations or bat population or time taken to obtain global optimal solution.

The algorithm proposed for achieving objective-1 will help in tracking targets moving at constant speed or predictable speed. The proposed algorithm is based on Constant Bearing pursuit strategy, which will help in achieving more optimized solution. Scope of first objective is to analyze performance of Range-Determination phase and Constant Bearing strategy adopted for obtaining optimal solution. This type of algorithm is suitable for Cloud Computing environment, where motive is to select optimal virtual machine and selection of virtual machines is done more likely from same zonal area. So, movement of target (i.e. Virtual Machine) is from same zonal area, but different from one virtual machine to another. So, in this case, movement of target may be negligible (in case of selection of same optimal virtual machine) or may be predictable (in case of selection of optimal virtual machine from same zonal area, but different virtual machine).

The algorithm proposed for achieving objective-2 will help in tracking targets moving at unpredictable speed or moving erratically. The algorithm is based on Constant Absolute Target Detection pursuit strategy, which will help in achieving more optimized solution. This type of algorithm is suitable for localization of sensor nodes in WSN, where motive is to select optimal sensor node either for acting as sender or recipient or may be as intermediate node. So, movement of target (sensor node) is very frequent in most of the cases. So, deployment of such algorithm becomes necessity to track such unpredictable sensor nodes. Bats search for optimal solution individually while utilizing information available in search space and conserving their energy levels.

The algorithm proposed for achieving objective-3 will help in tracking the movement of bats with respect to the movement of other bats. This proposed algorithm will help in such scenarios, where energy conservation is one of the aspects, while obtaining optimal solution. It can be implemented considering flight behavior of other bats, i.e. following, converging or diverging, either for cloud computing environment or wireless sensor network or any other application area where objective is to provide solution to continuous optimization problems.

1.12 Organization of the Thesis

The flow of this research work is organized as below:

Chapter 1: Describes types of optimization, approaches used to solve different optimization problems, swarm intelligence algorithms and selection criteria of different swarm intelligence techniques as per requirements. It also describes population of bats, Bat Algorithm, variants of Bat Algorithm inspired from echolocation.

Chapter 2: Review of existing literature, analysis and comparison of existing variants of BA and research gaps, are presented.

Chapter 3: Investigates RD-Bat Algorithm and its applicability for load balancing in Cloud Computing environment.

Chapter 4: Presents CATD-Bat Algorithm and its applicability for routing in Wireless Sensor Networks.

Chapter 5: Analysis the performance enhancement of Standard Bat Algorithm after adoption of different Pursuit Strategies and applicability of proposed algorithm to solve Traveling Salesperson Problem..

Chapter 6: Focuses on Performance Evaluation of RD-Bat, CATD-Bat, FBI-BA with respect to Mathematical Benchmark Functions.

Chapter 7: Concludes the research work and also emphasis on strength and weakness of BA. It also presents future direction and scope of this research work.

1.13 Summary

This chapter primarily focused on various optimization techniques opted for solving single-objective, constrained, un-constrained and multi-objective optimization problems. The literature review carried out by numerous researchers, on various approaches for solving different category of optimization problems, are studied and incorporated in this section. This section establishes major portion of research methodology. This chapter prepared basis for categorization of optimization problems and suggest optimization technique to solve different set of problems. This chapter has also explored echolocation feature of bats and presented how optimization algorithms are inspired from bats’ echolocation.

This chapter has thrown light on the life of bats, which belong to the same colony and also stated about the echolocation behavior. Details of Standard Bat Algorithm, its applicability areas, enhanced and hybrid versions were elaborated in this chapter. This chapter laid down the basis of research methodology of this research work. The next chapter will highlight aspects related to investigation of biological features of bats, which are adopted to develop another variant of Bat Algorithm.

CHAPTER 2
REVIEW OF LITERATURE

This chapter throws light on existing variants of Bat Algorithm and presents the survey of the same. Further, analysis of survey is presented along with the comparison of few variants of Bat Algorithm, in subsequent sections. Based on the comparison conducted, research gaps are identified and derived new variants of Bat Algorithm.

2.1 Highlights of Bat Algorithm

Existing variants of Bat Algorithm can be reviewed either by considering the biological characteristics of bats or year wise advancements which are introduced in Standard Bat Algorithm to enhance the performance. Next section presents the literature survey conducted based on biological features of bats.

2.1.1 Survey of Biological Characteristics of bats

In 1980, author of [81] has studied sounds produced by bats. The author has analyzed that bats produce different types of sounds for foraging, for tracking location of prey and for identifying size/shape of prey.

In 1985, work is carried out by author in [71] has studied behavior of bats. The author has also studied concept of echolocation. Echolocation, generally, refers to the sound produced by bats and echo received by bats. But author has thrown light on various other hidden information which is encoded in echolocation. Echolocation also contains information about distance between prey and bat. It also tells about angular position or direction in which bat should fly to capture prey. Moreover, bats can even predict type, size and shape of prey, before capturing, using echo received.

In 1988, work stated in [22] has studied echolocation behavior of bats and concluded that bats produce different types of sounds for detection and identification of prey. Different types of sounds are produced by varying range, cutter-noise ratio and detection sensitivity. One of the major components which helps bats in categorizing the prey's is detection sensitivity, also known as detection threshold. Detection threshold is an informative feature of bats.

In 1989, researcher of [82] has highlighted features of bats, i.e. how bats perceive prey. Bats use idea of cross-correlation to discover out similarity among sound produced and the echo received. Based on similarity index, bat decides which prey to capture, in presence of multiple preys. Auditory system of bats helps them in identifying shape of prey and to determine range between prey and itself.

In 2001, based on experiments conducted on foraging bats, author has suggested that echolocation behavior of bats vary according to climatic conditions, their habitat and their diet [72]. Author has also suggested that sounds produced by bats differ in frequency, duration, structure and pressure level of sound produced. The author has categorized foraging and echolocation behavior of bats in three different stages: search flight, approach flight, capture of prey. Search flight stage is when prey is not yet detected by bat. Approach stage is that stage when bat adopts different flight strategies and fly towards the prey to capture. Once prey is captured by bat, it enters third and last stage. The author has studied foraging and echolocation behavior of three different types of pipistrelles bats.

In 2006, author has studied pursuit behavior of bats [101]. Generally, bats follow either one of the two types of pursuit strategies: Constant Bearing or Constant Absolute Target Detection. The studies reveal that in order to capture targets moving erratically, bats adopt constant absolute target detection strategy. While capturing targets moving at constant speed of predictable speed, bats follow constant bearing strategy.

In 2008, author of [34] has analyzed behavior of bats and suggested that bats behave differently in presence of other bats. Bats stay in silent mode so that sound produced by multiple bats in that region do not interfere. The author has suggested that the bats keep shifting their roles from active mode to passive mode and vice versa, while producing sounds. Spatial separation between the ears of bats and bat heading direction also affects this shifting of roles among bats.

In 2009, author of [62] has suggested an intercepting strategy which is followed by bats while capturing preys. To predict time and position of futuristic interception point, bats follow anticipatory strategy unlike humans. The author has conducted two experiments and suggests that bats follow constant bearing strategy while keeping bearing angle constant for a shorter duration.

In 2010, an experiment is conducted to analyze movement of two bats, which are present in same search space [35]. Generally, one bat (follower) is following other bat (leader). The leader bat is always closer to prey, whereas follower bat saves its energy while moving towards prey, by just following leader. Most of the times, it was noticed during experiment that follower bat is successful in capturing prey, in comparison to leader bat. Both uses sonar beam to coordinate with each other, even if they are coming from different directions, to avoid signal jamming.

2.1.2 Survey of variants of Bat Algorithm

This section presents literature survey conducted related to the existing variants of Bat Algorithm. This survey is presented in ascending order of their publication.

In 2002, BA is used to solve two kinds of data mining problems [153], i.e. classification and time-series problem. Moreover, concept of chaotic maps can be used to enhance performance of algorithm.

In 2003, author has surveyed of various heuristic and meta-heuristic approaches. The author has also listed characteristics based upon which an algorithm is said to be ‘metaheuristic’. Followed by classification of metaheuristic approaches and also explained thin line difference between term’s ‘intensification’ and ‘exploitation’ and ‘diversification’ and ‘exploration’ [31].

In 2010, researcher has proposed new SI technique. This technique is based on echolocation behavior of bats, with few assumptions. The author has also suggested the computation of distance between bat and prey, in magical way [188]. Time- Delay estimation, directional echolocation, convergence rate improvement and identification of object using bat behavior are research areas that yet to be explored.

In 2011, work done in [190] is the extended version of Bat Algorithm, to achieve multi-objective optimization using Pareto Front. Population size is directly proportional to computational time. This work is suitable for solving non-linear and global optimization problems. To prove its applicability to engineering problems, author has applied this algorithm to welded beam design problem and results are astonishing, when validated against benchmark functions. The author has suggested that, to diversify search space, simple restart is beneficial rather than opting for longer iterations intense search.

In 2012, author of [79] surveyed different methodologies adopted so far for solving traveling salesperson problems. The author has conducted survey by grouping different types of animals. This work involves study of swarm intelligence-based techniques, school-based techniques, flock-based techniques and herd-based techniques which can be used for optimization. The author has also stated meaning of all different categories of techniques and suggested to conduct survey based on mammal-based techniques, in future.

In 2012, author has proposed a technique which is based on natural behavior of bats, i.e. feature selection technique. Binary Bat Algorithm, which is based on sigmoid function, is further used to maximize accurateness of algorithm. Feature selection technique is used along with echolocation behavior of bat. The exploration capability of bats is combined with optimum-path forest classifier, to improve accuracy of Bat Algorithm. Further, this work is tested over five public data sets [157].

In 2012, performance of a new meta-heuristic approach is evaluated, i.e. Bat Algorithm with intermittent approach and results proved that Bat Algorithm outperforms intermittent approach [197]. Further, author has suggested implementation of Bat Algorithm for solving business-based and engineering design-based problems.

In 2012, researcher has proposed a technique to identify best geometrical configurations in order to achieve objectives, with usage of minimum resources. This technique is used to solve topology based problems. Further, it is tested over non-linear benchmark functions used for designing in various engineering problems [196].

In 2013, author of [37] has proposed three different alternates of Binary Bat Algorithm for solving problem of graph-based road network. Three variables are introduced to develop these three different variants. Weibull Coded Binary Bat Algorithm, Real Bat Algorithm and Hybrid version of Weibull coded and real Bat Algorithm. These proposed variants are used for solving multi-objective problems. Number of bats require to obtain optimal solution are also reduced to a great extent. Finally, results are compared with Ant Colony Optimization and Intelligent Water Drop-based Optimization technique.

In 2013, balance between different search capabilities, dispersion of solutions and more focus on exploitation are three advancements introduced by [173]. In all variants of Bat Algorithm, motive of researcher is to either increase the convergence rate or to escape from being stuck in local optima or initialization of parameters, to improve performance of algorithm.

In 2013, concept of Differential operator and Levy flight Trajectory is applied to improve performance of Bat Algorithm. Differential operator is used to update frequency. This work has improved convergence rate of Standard Bat Algorithm. This algorithm is verified over fourteen mathematical benchmark functions and then applied to resolve non-linear optimization problems. The results proved that algorithm is effective, robust and produces more feasible results in comparison to Standard Bat Algorithm [97].

In 2013, to improve performance of Bat Algorithm, author has suggested some modifications, by introducing bats' foraging behavior [173]. Linear decreasing inertia weight factor is introduced to control exploration and exploitation. Moreover, adaptive frequency modification and scout bee modification is introduced. To enhance exploration capability, artificial bee colony technique is applied.

In 2013, emphasis is given on intensification and diversification of Bat Algorithm, initialization of parameters and improvement of convergence performance. Many fields are yet to be explored which includes the time delay estimation between emission of pulse and echo received by bat. Directional echolocation and Angular information fields can also be explored to propose a new alternate of Bat Algorithm. In Standard Bat Algorithm, position of bat gets updated along with frequency and loudness, but position of prey is assumed to be constant. But, in real world, the position of prey also gets updated with time [75].

In 2013, author of [97] has used the concept of differential operator for accelerating the convergence speed whereas levy flight behavior is adopted to explore the diverse population of search space. This algorithm is then applied to provide solution to non-linear problems and it is worth mentioning that this algorithm not only provides solution to the problem at hand, but also outperforms Standard Bat Algorithm. Due to their superior quality of approximation capabilities, this algorithm is suitable for providing solutions to such problems, which have high dimensional search space.

In 2013, to find location of nodes in wireless sensor network, author of [9] has suggested hybridization of Bacteria Foraging Algorithm with Bat Algorithm. Here author has focused on frequency tuning and maintaining dynamic balance between exploration and exploitation. For further enhancements, author has suggested to incorporate directional echolocation and the concept of Doppler Effect. In future, horizontal and vertical angle information can be included.

In 2013, researcher has worked on relationship between number of bats and operation to be formed, as back-propagation suffers with problem of network stagnancy [144]. To remove this problem, population of bats is increased and inclusion of OR & XOR operations. To evaluate whether performance is increased and decreased along with increase in count of bats for 2-bit OR, simulation with E-bit XOR/4-bit XOR operation is performed. The results are computed in terms of average CPU time, average accuracy and convergence epoch.

In 2013, author has proposed a new variant of Bat Algorithm which obtains global optimal solution with less computational cost and effort. This proposed technique is then implemented to reduce operating cost of thermal power plant [159].

In 2013, author of [74] has carried out literature survey on various swarm intelligence techniques, physics-based optimization techniques, chemistry-based optimization techniques and bio-based optimization techniques. The motive of this research is to present summary of above-mentioned techniques to inspire researchers for further research.

In 2013, researcher in [38] has hybridized Bat Algorithm with Shuffled Frog Leap optimization algorithm to optimize voltage levels of multi-distributed generations. The algorithm is used to solve problems associated with wind-based, solar-based, fuel-based and artificial model-based distributed generated systems. The results of proposed technique are then compared with Genetic Algorithm and Standard Bat Algorithm.

In 2013, author has suggested the implementation of Standard Bat Algorithm for unravelling non-linear engineering optimization problems. Later, technique's performance is validated over eight benchmark functions. Further, author has also suggested implications for future research work [192].

In 2014, author of [52] has used wrapper feature selection-based methodology in collaboration with optimum path forest model. Here feature selection is used as classification model, binary Bat Algorithm is used for optimization and optimum path forest model is used for accuracy. To estimate quality of obtained solution set, experiments are conducted using six data sets, available for public use. The results suggest that classification of data-sets is improved significantly, using proposed approach.

In 2014, author has introduced communication strategy between different groups of bats. The algorithm has replaced poorer solution present in a subgroup with the best solution present in neighbor subgroup of search space, after execution of each iteration. This characteristic makes algorithm suitable for applications which require parallel processing. To increase accurateness, convergence rate and speed of algorithm, these advancements are introduced. Scheduling the workflow is a NP-hard problem. To crack different types of NP-hard problems, many researchers have suggested different optimization approaches [36].

In 2014, work mentioned in [207] has categorized bio-inspired computation as part of artificial intelligence, evolutionary and computational intelligence. Generally, it refers to self-organized nature, adaptive to situation and holds capability of dealing with random inputs. Bio-inspired computations include swarm intelligence, membrane and memetic-based computation, DNA and molecular-based computations, neural network-based computations, neuroscience computations, bio-informatics, natural language processing, software engineering, machine learning, data mining, algorithmic theories and many more.

In 2014, author has suggested that Bat Algorithm is preferable over Particle Swarm Optimization and Intelligent Water Drop-based Optimization [20] for better accuracy and efficiency. This proposed methodology is further evaluated over 3 unit and 6 unit systems.

In 2014, author has used fuzzy logic to initialize parameters of Bat Algorithm. The results of algorithm are equated with results obtained using Genetic Algorithm. Further, it has been evaluated over benchmark functions to test effectiveness of fuzzy based Bat Algorithm [94] [185].

In 2014, author used the concept of automatic zooming to maintain equilibrium between exploration phase and exploitation phase of Bat Algorithm. To boost intelligibility of speech and to improve quality, this work has been applied for speech enhancement. It has been verified that Bat Algorithm outperforms PSO in improving value of speech. The quality of speech is enhanced by focusing on reducing/cancelling the noise using dual-microphone systems [106].

In 2014, concept of complex numbers is used to represent one dimensional number line, which enhances diversity and optimization performance of Standard Bat Algorithm. In future work, rather than using only mathematical concepts to boost performance of algorithm, biological features of bat can be explored to develop different alternates of Bat Algorithm [112].

In 2014, cartesian coordinate system can be used to represent one dimensional number line using three-dimensional number line. As direct application of Bat Algorithm for solving various problems is not feasible. So, [168] has applied concept of binary Bat Algorithm to real world application of optical engineering. The performance of Binary Bat Algorithm is proven to be superior to Standard Bat Algorithm and PSO over twenty-two mathematical benchmark functions.

In 2014, author of [4] has suggested involvement of mutation and crossover operator to improve efficiency of search and to speed up convergence rate of basic Bat Algorithm. The work is then compared with hybridized Bat Algorithm with differential evolution. This work yields better results for unimodal and multimodal problems and have been tested over five different set of benchmark functions. In most of variants of Bat Algorithm, initialization of parameters like frequency, wavelength and position is done randomly.

In 2014, pseudorandom generation of initial values of parameters is replaced with chaotic sequences [66]. Gauss Map is used to initialize frequency and loudness and Tent Map is used to set pulse rate. This algorithm has improved convergence rate of Standard Bat Algorithm. Usage of chaotic sequences has added capability to Standard Bat Algorithm to avoid getting stuck in local optima and explore other solutions to generate global optima. This algorithm has yet to prove its efficiency in large scale optimization problems.

In 2014, author has focused on diversification of population and optimization performance of Bat Algorithm is enhanced using real and imaginary part separately of complex valued encoding [112]. In previous variants of Bat Algorithms, ability to learn from previous experiences is missing. To obtain optimal results, this algorithm has set range of complex modulus, phase angle and variables. The work has been tested over fifteen different functions.

In 2014, author of [14] has used the concept of only two bats to attain optimal solution. These two bats exchange their responsibilities of exploration and exploitation from time to time. Direction of movement of bat is considered here. The results are then equated with results of PSO and BA. The algorithm is proven to be improved in terms of accuracy and robustness.

In 2014, Doppler Effect on frequency shift is considered for updating frequency of artificial bat during exploration and exploitation [200]. Concept of refined search and divers are considered to attain better solution. Three main areas explored in this research work includes conflict among bat population, competition among bat population and cooperation among the same. The results are computed on basis of accuracy, success rate and problem-solving speed.

In 2014, initialization of all parameters, except bat position, is assigned randomly, in order to apply binary Bat Algorithm to graph coloring problem. The key focus of [67] is to color each vertex of graph with constraint that no two adjacent vertices have same color. RLF algorithm is used to initialize bat positions. Moreover, sigmoid function is used to ensure values either zero or one. The results are evaluated using different instances of Graph Coloring Problem.

In 2015, researcher has proposed a multi-swarm technique for obtaining global optimal solution [99]. Here, immigration operator is used to exchange information between different groups of swarm and accordingly need to do parameter adjustments. Selection operator is used to select elite solutions present in search space. This technique is tested over sixteen mathematical benchmark functions. Further results are equated with Standard Bat Algorithm and proven to be beneficial. Another author has identified two main tasks of process planning: selection of operations and sequencing of those operations. A hybrid Bat Algorithm is proposed to crack these kind of problems. To make Standard Bat Algorithm adaptable to process planning problem, encoding, decoding and initialization of population is done. Two new parameters are introduced to modify existing Bat Algorithm.

In 2015, author of [134] has proposed a technique of optimization while widening scope from single objective to multi-objective objectives in the field of wireless sensor networks. The author has focused on optimization of two aspects: area coverage and left-over energy of nodes. The aim is to select best-fittest node as cluster head. The author has used concept of Bat Algorithm to achieve above-mentioned objectives.

In 2015, researcher of [169] suggested a mechanism to schedule the workflow in cloud computing environment, binary Bat Algorithm is used. Resources are assigned to tasks in such a fashion, so that overall cost of workflow is minimal. The results are compared with Best Resource Selection algorithm and proven to be fifty percent better than former.

In 2015, work mentioned in [2] has described a new technique of wrapper feature selection for intrusion detection. The role of intrusion detection is to identify malicious and benign packets present in network. To do so, a technique is proposed by author, which is hybridization of two machine learning techniques along with modified version, i.e. binary version of Bat Algorithm. The motive of this technique is to increase attack detection rate and decrease ratio of false alarms raised due to false positive detection.

In 2015, author of [137] has suggested that to improve the performance of automatic voltage regulator and load frequency controller, one should work on optimizing parameters of proportional integral derivation controller. For the same, author has suggested usage of Bat Algorithm and thermal interconnected power system of three equal area is considered for simulation. The results are then compared with Genetic Algorithm, Artificial Bee Colony, Particle Swarm Optimization and Bacteria Foraging Optimization. Zieglar-Nicholas based proportional integral derivation is used as interconnected power system for simulation. For performance evaluation, robustness is considered as the parameter.

In 2015, it has been cited that use of Bat Algorithm to plan training sessions for sports person, based on one' physical and mental strength, recorded via sports watch [77]. In future, extraction of relevant and meaningful data from XML files, can be done using appropriate data mining techniques. Many researchers have has transformed continuous space into binary space, by applying different methods.

In 2015, author of [204] has suggested the integration of sigmoid function. The real and imaginary part of any complex number are updated irrespective of each other, so that diversity of solutions and parallelism among multiple operations can be increased. In order to increase search space and convergence performance, complex number encoding is used. This variant of Bat Algorithm is used to solve large-scale and small-scale knapsack problems. Further, to prove the capability of proposed algorithm, comparison with existing techniques for solving knapsack problems is done.

In 2015, PSO is hybridized with Bat Algorithm [181]. To attain improved results, after each iteration, worst elements of PSO get replaced with best elements of BA and vice versa. Back Propagation Neural Network, identifies errors present in hidden layer by calculating errors of outer layers. The results of hybrid PSO and BA are compared with PSO on six different functions and proven to be better in terms of accuracy and convergence rate.

In 2015, work carried out in [171], localization error is minimized by using range-free algorithm, i.e., mobile anchor positioning. To achieve optimization, meta-heuristic techniques are applied. This work has thrown light on range-based, range-free, mobile anchor-based and hybrid localization techniques. The proposed work is not suitable for underwater and indoor setup of ad hoc network. For this, natural extension of bat can be explored. This is because bats and frogs have the capability to identify location and type of object based on water ripple information.

In 2015, researcher focuses on self-adaptive search strategy [186]. To avoid being trapped in local solution, randomization approach of initialization of parameters is used. Updation of loudness, pulse emission rate and compensation rate affect performance of algorithm. Moreover, bat population of this algorithm exhibits quantum behavior and offers a wide range of diverse solutions, along with improvement in convergence rate.

In 2015, author has used concept of double sub-population Levy flight to introduce this ability to Standard Bat Algorithm [115]. Greedy algorithm is used here. Greedy algorithm fails to find global optimal solution. It updates position of solutions, either one of two subgroups: external and internal. External subgroup updates position using dynamic weight model and internal subgroup updates same using levy flight model. Moreover, it is applicable only for unimodal applications, not for multimodal applications.

In 2015, author has proposed Chaotic BSO [11]. It relies on linearly decreasing function which is multiplied by chaotic map function to update loudness. The results obtained are then compared with Standard Bat Algorithm, Cuckoo Search Optimization, Bing bang-big crunch Algorithm, Genetic Algorithm and Gravitational Search Algorithm. This algorithm outperforms other metaheuristic approaches in terms of obtaining consistently feasible results.

In 2015, Bat Algorithm is used for optimization of weights and structure of neural network [142]. This approach is useful in unravelling other real-world problems like pattern recognition. This algorithm focuses on improvement of trade-off between exploration and exploitation. The author of [99] has integrated concept of Cloud Model, along with Bat Algorithm.

In 2015, author has focused on re-modeling echolocation model based on foraging and habitat. The motive of research done in [40], is to find best system configuration while minimizing loss rate. To achieve this, there is a need to select proper state of switches. Here, optimal selection of state of switches is accomplished using Bat

Algorithm. The efficiency of algorithm is evaluated on thirty-three different bus distribution systems and resulted into reduction of 33% loss rate.

In 2015, author has proposed the technique for feature selection using appropriate learning algorithm, to reduce classification error [21]. Metaheuristic techniques are proven to be beneficial for solving such kind of problems. Here, author has applied Bat Algorithm for achieving optimization in feature/full selection-based model. This technique is tested over benchmark datasets of gene expressions. The results have proved its applicability and effectiveness on the given problem.

In 2016, discrete version of Bat Algorithm for solving traveling salesman problem is suggested in [203]. Results obtained using this methodology outperforms Standard Bat Algorithm. Moreover, results are evaluated using standard benchmark functions available at TSPLIB library for symmetric traveling salesman problem.

In 2017, Bat Algorithm is applied to community detection problem [54]. To avoid being trapped into local optimal solution, best among top 'n' bats is selected and for representation of solution, locus-based adjacency encoding scheme is used. From results, it can be derived that Bat Algorithm is suitable for small scale data-sets. But, when large amount of data-sets are involved, Bat Algorithm fails to perform well in comparison to other community detection problems. Limitation of basic Bat Algorithm lies in diversification of all possible directions of search space. Due to that, success rate of BA is less.

In 2018, author has mapped continuous Bat Algorithm to discrete version of Bat Algorithm [206]. The focus is to provide solution to antenna positioning problem, which is a NP-hard problem. Here five variants of Bat Algorithm are proposed and have been tested using well-known mathematical benchmark functions. The results are considered to be better than other existing meta-heuristic techniques.

After reviewing research work carried out in past, by various researchers, it has been identified that there is need of new variants of Bat Algorithm. These variants should be suitable for different scenarios. This work primarily focuses on proposal of new variants of Bat Algorithm by incorporating biological behavior of bats.

Below given work flow is embraced in this research work, to identify research gaps based on literature survey:

2.1 Analysis of Survey in which decision regarding the selection of appropriate optimization technique is carried out.
2.2 Comparison of existing optimization techniques and to analyze which of the optimization techniques are suitable for solving which type of problem. The comparison can be done on the basis of single vs. multiple objective optimization problems, constraint optimization problems, conventional or swarm intelligence-based optimization problem. It should be known in advance that how many iterations it will take to produce optimal solution or what could be the population size and what will be effect on performance of algorithm, if changes are incorporated in underlying concept of the algorithm?
2.3 Comparison of existing optimization techniques on the basis of time taken or number of iterations for obtaining optimal solution or convergence rate. This type of analysis will yield result, which focuses on selection of optimization algorithm, under different underlying constraints. In this sub-section, research gap of existing work is identified.

2.2 Analysis of Survey

In this thesis, literature survey is carried out in two different aspects. One type of survey is related to inclusion of biological features/characteristics of bats. Another type focuses on applicability of newly developed Bat Algorithm. Newly developed variants of Bat Algorithm focus on one of the following advancements:

1. Parameter Initialization: The basic parameters used by Bat Algorithm for obtaining optimal solution are frequency, pulse emission rate, velocity, position and loudness. The initial value of these parameters matters a lot while targeting optimal solution. Adoption of different strategies for initialization of these parameters will

definitely affect kind of solution obtained and time taken to obtain that optimal result.

2. Parameter Updation: After the initialization of parameters, one needs to update those parameters during the exploration and exploitation phase of Bat Algorithm. The way these parameters get updated depends upon either on basic characteristics of bats or on inclusion of mathematical algorithms.
3. Enhancement in Exploration & Exploitation: The way bats explore neighborhood search space and obtain local optimal solution, depend upon the advancements introduced during exploration and exploitation phase.
4. Mapping to Binary Search Space: Standard variant of Bat Algorithm is related to continuous search space. To extend applicability of Bat Algorithm, numerous variants of Bat Algorithm are mapped to discrete versions of Standard Bat Algorithm.
5. Hybridization: To further improve performance of Bat Algorithm, hybridization of two or more swarm intelligence-based optimization techniques is done. Moreover, few researchers have hybridized Bat Algorithm with conventional optimization techniques, to further improve its performance.

2.3 Comparison of existing variants of Bat Algorithm

The review conducted on different variants of Bat Algorithm are mentioned in previous section and are summarized in Table 2.1.

Table 2.1: Comparison of Bat Algorithm variants

Algorithm Used	Modifications Introduced	Result Validation
Guidable Bat Algorithm [200]	1. Velocity and Frequency Updation using Doppler Effect. 2. Low pass Filter to filter noise. 3. Avoid falling into local optima using Divers Search	1. Rastrigin Function 2.Griewangk Function
Binary Bat	1. Velocity and Frequency	1. Optical Buffer

Algorithm [206]	Updation 2. Use of Transfer Function to map Continuous Search Space to Binary Search Space.	Design in Optical Engineering
Bat Algorithm [188]	Standard Bat Algorithm	1. Rosenbrock Function 2.Ackley Function 3. Michalewicz Function
Enhanced Bat Algorithm [197]	1. Inertia Weigh Factor to avoid premature Convergence 2. To increase Exploration. 3.To increase local search capability, Invasive Weed Optimization is used	1. Welded Beam 2. Spring Design 3. Pressure Vessel Design
Multi-Objective Bat Algorithm [201]	Pareto Optimality to increase Exploration	1.Rastrigin Function 2.Griewangk Function
Complex Valued Bat Algorithm [204]	Enhances Exploration	Using Unimodal and Multimodal Function
Bat Algorithm [23]	Frequency Tuning and Dynamic Control of Exploration and Exploitation	1. Pressure Vessel Design 2.Welded Beam Design 3. Tension Compression Spring Design
Hybridized Bat Algorithm with Differential Evolution [75]	Addition of Mutation and Crossover to avoid trapping in local optima.	Using Convex, Seperable Non-Seperable, Continous Unimodal and

		Multimodal Functions
Chaotic Bat Algorithm [11]	Parameter initialization is done using chaotic sequences, rather than random initialization	1. High Dimensional and Low Dimensional Functions 2.Unimodal and Multimodal Function
Bat Algorithm [94]	Parameter Initialization using Fuzzy Logic	Inverted Pendulum Problem
Double Sub-population Levy Flight Bat Algorithm [90]	Improves Exploration and Exploitation	Using Standard Functions
Binary Bat Algorithm [67]	1. RLF algorithm for initialization 2.Sigmoid Function to represent continuous space in binary encoding	Graph Coloring Problem
Chaotic BAT Swarm Optimization [11]	Loudness computation is done using Chaotic Map Functions	Using Unimodal and Multimodal Functions
Modified Bat Algorithm [131]	Enhances Exploration and Exploitation	Classification of Time-Series Prediction
Dynamic Virtual Bat Algorithm [14]	Enhances Exploration and Exploitation	Using Unimodal and Multimodal Functions
Modified Bat Algorithm [77]	Mapping of Real-valued search space to Integer Vector	Planning of Sports Training Sessions
Discrete Bat Algorithm [54]	Locus based adjacent encoding scheme	Community Detection Problem
Modified Bat Algorithm [160]	1. Increase in Success rate in Localization	Localization in Wireless Sensor

	2.Increase in Exploration	Network
Hybridization of PSO and Bat Algorithm [146]	Replacement of worst solutions of Bat Algorithm with best particles of PSO and vice versa	1. Rosenbrock 2.Ackley 3. Griewank 4.Quadric 5.Rastrigin 6.Spherical
Binary Bat Algorithm [170]	Using Sigmoid Function	Scheduling of Jobs in Cloud Computing
Binary Bat Algorithm [52]	To improve Optimum Path Forest effectiveness	Using Standard Functions
Novel Bat Algorithm [66]	Better Habitat Selection technique and use of Doppler Effect	1.Welded Beam Design 2.Coil Compression Spring Design 3.Speed Reducer Design
Bat Algorithm [106]	Stochastic Optimization based Speech Enhancement	Dual Channel Speech Enhancement
Parallelized Bat Algorithm [36]	Improves Convergence by parallel searching best solution in subgroups	1.Rastrigin Functions 2.Griewank Function 3.Ackley Function 4.Spherical Function
Bat Algorithm and Cuckoo Search [189]	Standard BAT and Cuckoo Search Algorithms	Gaussian, Cauchy and Levy Distribution
Complex valued encoding Bat Algorithm [204]	1. Separate Updation of real and imaginary part 2. Use of Sigmoid function	0-1 Knapsack Problem
Bat Algorithm [104]	Standard Bat Algorithm	Back-propagation algorithm of ANN

Bat Algorithm with recollection [67]	1. Time-Delay disturbance factor 2.Time varying velocity inertia weight factor	Graph Coloring Problem
Bat Algorithm [155]	1. Hybridization of Bat Algorithm with Mobile Anchor Positioning Algorithm 2.Hybridization of Cuckoo Search with Mobile Anchor Positioning Algorithm 3. Hybridization of Firefly Algorithm with Mobile Anchor Positioning Algorithm	Localization of Wireless Sensor Network

2.4 Research Gaps

The shortcomings of various swarm intelligence techniques is that they are unsuccessful while maintaining good level of accuracy, optimum precision value and faster convergence rate while solving problems and obtaining global optimal solution. Though, a perfect equilibrium between intensification (exploitation) and diversification (exploration) phases of algorithm is essential. Generally, terms 'intensification' and 'diversification' are referred as 'exploitation' and 'exploration', respectively. But these terms are quite different in terms of deployment of strategies according to usage of memory. Exploration and Exploitation terms denote those strategies which meant for short term and are associated with initial random solutions. Whereas Intensification and Diversification terms are associated with medium to long term strategies. Another important aspect is to maintain trade-off between these two. As insufficient exploration or excessive exploitation may result into breakdown of the system, which leads to trapping to local optimal solution instead of global optimal solution. Above mentioned issues must be taken care of, so as to make swarm intelligence techniques more efficient, consistent and operative, which generates more prominent solutions to any single or multi-objective optimization problems.

Bat Algorithm relies on assumption of calculating distance between bat and prey, in a 'magical way'. In order to obtain optimal solution 'timely', while satisfying underlying constraints, it becomes important to track movement of prey (target). The movement strategy adopted by bats depend upon the movement of prey. Few preys tend to move at constant speed and direction, which is predictable to bats. This research work is motivated from assumption laid down by author Yang. Another research gap that is identified during this research work is that, if multiple prey (targets) are present in search space then it affects selection of optimal target. Selection of feasible solutions depends upon range between bat and target, and also depends upon movement of target.

As per biological features of bats, they adopt different pursuit strategies for capturing static prey, target moving at predictable speed and target moving at unpredictable speed. This objective primarily focuses on tracking of such preys, which are moving erratically, i.e. at unpredictable speed. One more aspect of bats is explored which is related to another characteristics of bats. Most of nature inspired optimization techniques relies on fact of obtaining optimal solution with collaborative work of swarm population. Bat Algorithm is such a nature inspired optimization technique, where one bat of a swarm may jam the sound produced by another bat of same swarm or it may steal information encoded in sound produced and received echo or by entering in 'silent' mode. In existing literature survey, presence of other target seekers, seeking for optimal solution in same search space is not considered.

2.5 Summary

This chapter has summarized different variants of Bat Algorithm. Also, analysis of survey is presented and research gaps. Based on research gaps, new variants can be derived. Further, tool used for implementation of proposed objectives is stated in previous section. The next chapter will highlight aspects related to investigation of Range-Determination based Bat Algorithm variant.

CHAPTER 3
DESIGN AND DEVELOPMENT OF BAT ALGORITHM VARIANT USING CONSTANT BEARING STRATEGY

This chapter focuses on first objective of this research work, i.e. "To develop an approach for tracking targets moving at predictable speed using Constant Bearing strategy". This chapter presents investigation Range-Determiner Bat Algorithm (RD-BA) which is inspired from echolocation behavior and flight behavior of bats.

3.1 Inspiration

In this work, selection of an optimal target depends upon range between target and target seeker, and direction in which target is moving. In this methodology, to calculate range between target and target seeker, strategy which is opted by real bat to compute distance between target and itself, will be employed. Most of the researchers have contributed in this field, by computing distance using hamming code and by applying 2-opt and 3-opt to get more optimized result during exploitation phase. To develop a new modified version of Bat Algorithm rather than computing fitness value using mathematical function, Rosenbrock, a new way of computing distance is proposed. The bat makes use of echo-delay estimation data, to compute distance between target and itself. Echo-Delay Estimation can be done by using Cross Correlation, which will lead to development of new variant of Bat Algorithm. Here, focus is on calculation of distance using mathematical technique, Cross Correlation. Cross Correlation technique is basically used for evaluating similarity between two signals. In this research work, Cross Correlation is used to identify similarity between pulse generated and echo received. Real bats have ability to differentiate between the multiple echoes received, on the basis of similarity between the pulse generated (signal transmitted) and the echoes received. The calculation of "similarity among various signals" forms basis of calculating distance between target and real bat.

Till now, in existing work, authors have worked on assumption of calculating distance in "some magical way" and no technique is adopted for same. "Range Determination" can act as basis for computing "fitness value" and will lead to development of new variant of Bat Algorithm (Range Determiner –Bat Algorithm). By considering "distance computation" or "range determination" as one of the factors for calculating fitness value, others objectives can be achieved.

Fitness value is calculated in terms of distance between bats and all feasible solutions, over multiple iterations. After iteration, local optimal solution is selected, whichever is having minimum fitness value, as goal is to minimize distance. After obtaining local optimal solution, bats will enter exploitation phase and try to obtain global optimal solution in neighborhood of local optimal solution. In proposed work, exploitation phase will remain as that of Standard Bat Algorithm.

Depending upon assumptions and biological features, basic steps of a new variant of Bat Algorithm (RD-BA) are explained below and steps of RD-BA are summarized in Figure 3.1.

3.2 Range Determiner-Bat Algorithm

Step 1: Initialization

Given bat population is 'n' and dimension of search space is represented as 'D'. The initial bat population is represented by $\{b_1,b_2,b_3,\ldots\ldots,b_n\}$ and its parameters are initialized randomly. The parameters of bat population include following:

1. Position of bat population is represented as $\{x_1,x_2,x_3,\ldots..x_n\}$.
2. Minimum and Maximum frequency is denoted by f_{min} and f_{max}, respectively.
3. Frequency of bat population is represented as $\{f_1,f_2,f_3,\ldots..f_n\}$.
4. Velocity of bat population is represented as $\{v_1,v_2,v_3,\ldots..v_n\}$.
5. Pulse Emission Rate of bat population is represented as $\{r_1,r_2,r_3,\ldots..r_n\}$.
6. Loudness of bat population is represented as $\{A_1,A_2,A_3,\ldots..A_n\}$.

After initialization of bat population and its parameters; based on nature of solution, either static (stationary) or dynamic (changing/moving); selection of Bat Algorithm variant is done. In case of stationary solution, Standard Bat Algorithm is applied to obtain optimal result; otherwise proposed variant of Bat Algorithm will be used.

Step 2: Computation of Fitness Value

To compute fitness value of Standard Bat Algorithm, mathematical function Rosenbrock is used. In this research work in order to compute fitness value of proposed Bat Algorithm variant, following step have been implemented:
1): Bats will produce sound and wait for echo.
2): Once the bat receives echo, it will try to detect presence of obstacle on way.

3): If any obstacle is present, attenuation will affect echo and some delay will be incurred.

$$RS_i = (SS_i * \alpha) + \beta$$

where RS_s and SS_i represents received signal and sent signal and α represents attenuation factor and β represents delay incurred.

4): In absence of obstacle, delay will be considered in the received sound.

$$RS_i = SS_i + \beta$$

5): Artificial bats will then compute cross correlation between pulse emitted and echo.

$$[Corr, Lags] = xcorr(SS_i, RS_i)$$

6): Among all emitted pulses and received echoes, whichever is having minimum similarity will be considered as actual echo of pulse emitted.

$$Delay\_Samples = Lags(find(Corr == max(Corr)))$$

7): Based on that, delay samples and time samples will computed, which will be used to compute distance.

Step 3: Exploration Phase

During this phase, for ‘m’ number of iterations, artificial bats will move randomly in search space and generate new solutions to explore for obtaining optimal solution, by adjusting parameters, using below given instructions. The equations (1) to (5) update frequency, velocity, position, loudness and pulse emission rate of each bat.

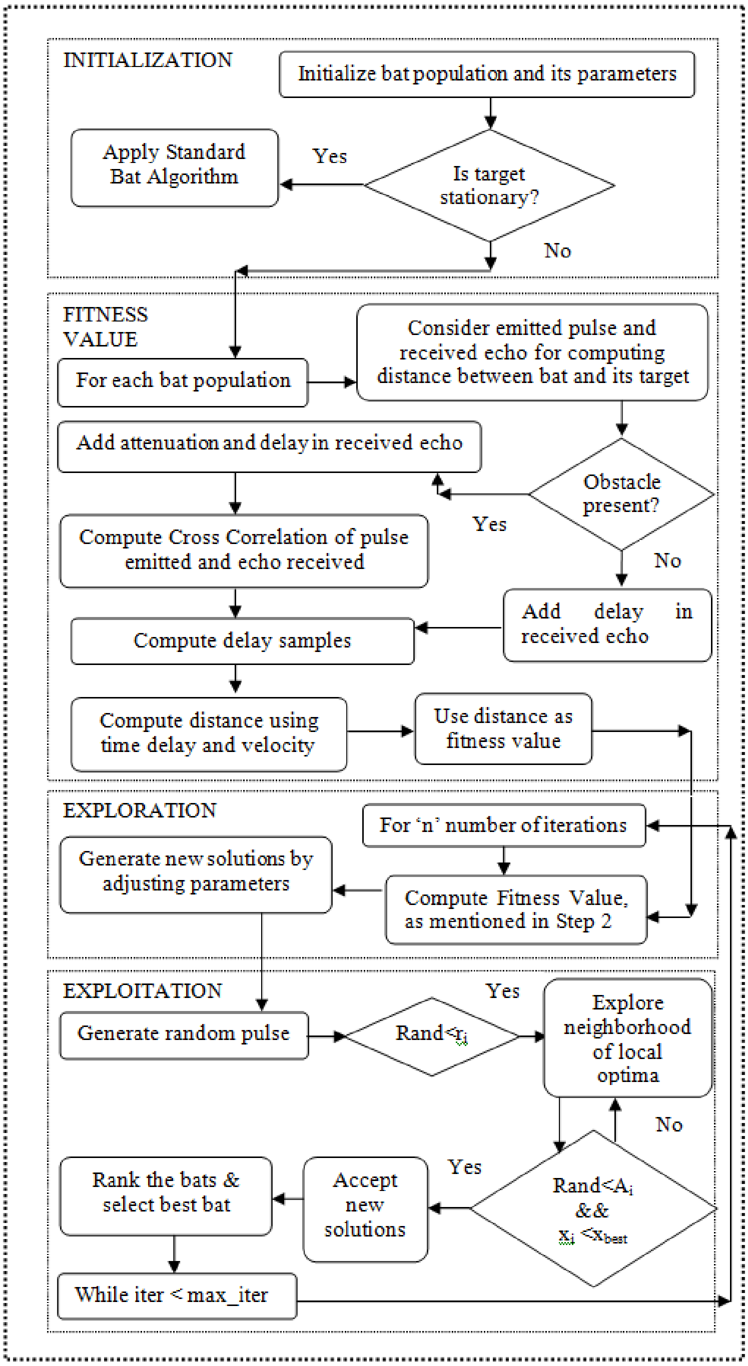


Figure 3.1: Process Flow Diagram of Range-Determiner Bat Algorithm

$$f_i = f_{min} + (f_{max} - f_{min}) * \beta \qquad -(1)$$

$$v_i^t = v_i^t - 1 + (x_i - x_{best}) * f_i \qquad -(2)$$

$$x_i^t = x_i^t - 1 + v_i^t \qquad -(3)$$

$$A_i^{t+1} = 0.9 * A_i^t \qquad -(4)$$

$$r_i^{t+1} = r_i^t * (1 - \exp(-1 * 0.9 * i)) \qquad -(5)$$

Step 4: Exploitation Phase

During this phase, search space in neighborhood of local optimal solution is exploited to obtain global best solution. The artificial bats will move in random directions to check for feasible solutions. If fitness value of any solution is minimum than currently selected local optimal solution, then it will be treated as best solution. Best solutions obtained for all iterations are ranked and solution having minimum fitness value is selected as global optima.

3.3 Computer Simulation and Discussion

It is evident from pseudo code, that it is easy to implement concept of Bat Algorithm using any programming language. In this research work, MATLAB is used as programming language to implement RD-Bat Algorithm.

3.3.1 Set-up of Execution environment

The main parameters of Bat Algorithm and its variants are frequency, loudness, pulse emission rate, position and velocity. Initial values assigned to these parameters play a major role during exploration and exploitation phase of algorithm, while obtaining global optimal solution. These parameters are represented using mathematical notations, as depicted in Table 3.1. Summary of parameters, along with their description and initial values are also mentioned in Table 3.1. Considering literature available, initial value of above-mentioned parameters are decided.

- Bat Population '*n*': Result Validation is carried out by varying bat population between 25 and 100, with the interval of 25.
- Frequency 'f': The movement of bats is based on the frequency. The lower and higher bound used are 0 and 2 respectively.

- Loudness ‘A’: The value of this parameter is dependent on selection of α. Here, α=0.9, which helps in computing loudness parameter, which helps in accepting new feasible solutions.
- Pulse Emission Rate ‘r_i’: The value of constant β is fixed to 0.9 to obtain value of this parameter. r_i keeps on increasing, as artificial bats start moving towards global optimal solution.
- Number of Iterations ‘m’: The number of iterations is varied over [250, 500, 750,1000].

Table 3.1: Parameters of RD-Bat Algorithm

Parameter	Notation	Value
Bat Population	N	[25 50 75 100]
Pulse Emission Rate	r_i	0.5
Loudness	A_i	0.25
Lower Bound Frequency	f_{min}	0
Upper Bound Frequency	f_{max}	2
Number of Iterations	M	[250 500 750 1000]
Loudness Constant	A	0.9
Pulse Emission Constant	B	0.9

Performance evaluation of Bat Algorithm can be done in two ways: achieve higher accuracy in finite number of rounds or for fixed number of rounds, compute the accuracy of the algorithm. Apart from this, one can evaluate performance based on accuracy rate, number of fitness function evaluations, comparison of mean value, best value, worst value, median of series and standard deviation, by varying initial values of constants and dimensions. In Standard Bat Algorithm, performance is evaluated by executing over multiple iterations, till higher value of accuracy is not achieved. But, in this research work, RD-Bat Algorithm is executed for predefined finite number of rounds to calculate accuracy of algorithm.

Table 3.2: Result Evaluation of RD-Bat Algorithm w.r.t. Standard Bat Algorithm

Maximum Number of Iterations	Parameters	25		50		75		100	
		Standard Bat	RD-Bat	Standard Bat	RD-Bat	Standard Bat	RD-Bat	Standard Bat	RD-Bat
250	Best	0.0270	0.0226	0.0173	0.0143	0.0072	0.0059	0.0056	0.0045
	Median	0.5740	0.4831	0.2663	0.2194	0.2137	0.1744	0.1041	0.0844
	Worst	2.2150	1.8756	1.7019	1.4228	0.9003	0.7417	0.6517	0.5298
	Mean	0.7218	0.6049	0.4414	0.3648	0.2805	0.2295	0.1679	0.1353
	SD	0.5819	0.4823	0.4616	0.3829	0.2496	0.2047	0.1774	0.1434
500	Best	0.1793	0.1462	0.0028	0.0023	0.0006	0.0005	0.0042	0.0033
	Median	0.4906	0.4047	0.2560	0.2074	0.1808	0.1448	0.1102	0.0874
	Worst	5.1062	4.2317	1.5470	1.2561	0.6557	0.5281	0.8123	0.6513
	Mean	0.7523	0.6209	0.3910	0.3169	0.2122	0.1703	0.1720	0.1369
	SD	1.8107	1.4962	0.3421	0.2771	0.1521	0.1223	0.1792	0.1435
750	Best	0.1424	0.1162	0.5556	0.4492	0.0017	0.0014	0.0040	0.0032
	Median	0.5039	0.4134	0.3757	0.2737	0.2048	0.1633	0.1480	0.1174
	Worst	3.1917	2.6254	1.6964	0.4520	0.8975	0.7175	0.7108	0.5681
	Mean	0.8852	0.7237	0.5266	0.3142	0.2513	0.2006	0.1893	0.1501
	SD	0.7643	0.6275	0.4858	0.2135	0.2611	0.2086	0.1696	0.1352
1000	Best	0.0724	0.0585	0.0166	0.0133	0.0247	0.0195	0.0295	0.0228
	Median	0.4915	0.3981	0.2306	0.1862	0.2278	0.1799	0.1181	0.0919
	Worst	4.0529	3.3044	0.9519	0.7668	1.2308	0.9761	0.6217	0.4872
	Mean	0.7998	0.6488	0.3297	0.2657	0.3136	0.2483	0.1876	0.1462
	SD	0.9104	0.7414	0.2762	0.2227	0.3202	0.2540	0.1750	0.1368

The RD-Bat Algorithm is executed for 25 times to produce results, so that some statistical comparison can be carried out. Here, results are shown for bat population ranging from 25 to 100. Each bat population is iterated over 250, 500, 750 and 1000 times and results are recorded at end. Table 3.2 shows optimization results with different bat population over different iterations. For evaluation, α and β have been assigned with 0.9 value for all iterations.

3.3.2 Initialization of parameters and performance study

For N=25, it has been observed from results that for lesser number of bats, optimal results can be obtained even with 250 iterations. In case of a greater number of iterations, bats will keep on searching for the optimal solution unnecessarily.

For bat population, N=50, maximum number of iterations required to obtain optimal result are 750. The graphical representation for bat population 25 and 50 is presented in figure 3.2 and 3.3. It is evident for bat population 25, with increase in number of iterations, more optimal result is obtained. Higher number of iterations force bats to update parameters and avoid getting trapped in local solution.

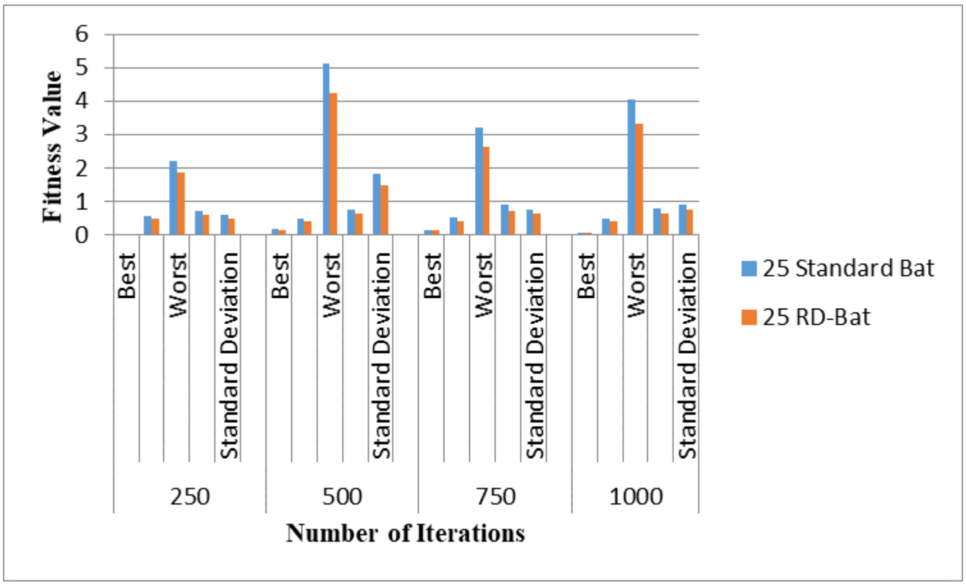


Figure 3.2: For 25 bat population, over [250, 500, 750, 1000] iterations, comparison of Standard Bat Algorithm vs. RD-Bat Algorithm

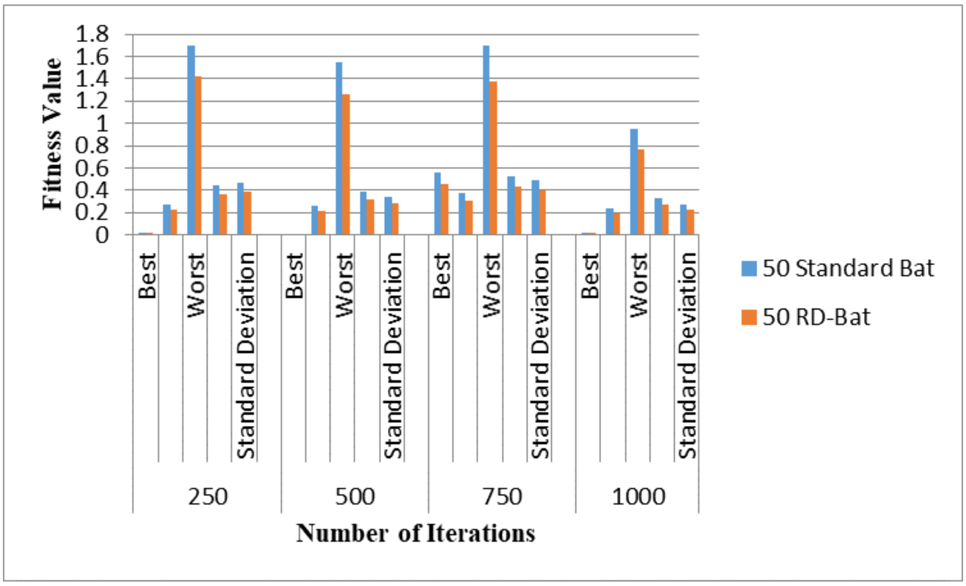


Figure 3.3: For 50 bat population, over [250, 500, 750, 1000] iterations, comparison of Standard Bat Algorithm vs. RD-Bat Algorithm

For 250 iterations, best solution can be obtained using 100 bat population and for 500 iterations, best solution can be obtained using 75 bat population. While comparing the mean values obtained for RD-BA in comparison to Standard Bat Algorithm, there is a significant improvement of 16 % and 17% over 250 and 500 iterations respectively. As bat population increases, better results can be produced by using proposed RD-Bat Algorithm. Moreover, in comparison to Standard Bat Algorithm, proposed RD-Bat Algorithm shows better convergence rate.

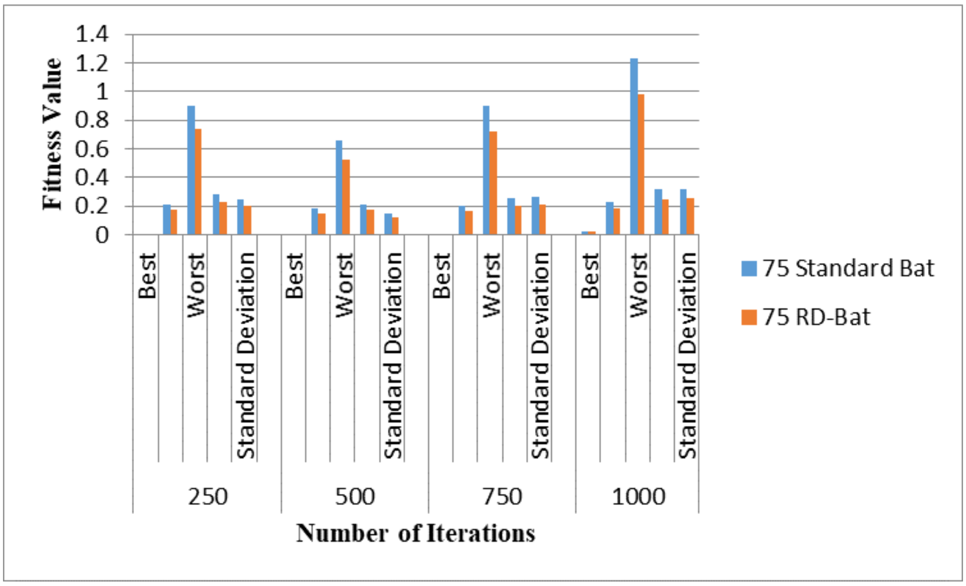


Figure 3.4: For 75 bat population, over [250, 500, 750, 1000] iterations, comparison of Standard Bat Algorithm vs. RD-Bat Algorithm

Figure 3.4 and 3.5 reflects comparison of Best value, Mean value, Median of series, Worst value and Standard Deviation of Bat Population equal to 75 and 100 over [250, 500, 750, 1000] iterations for Standard Bat Algorithm and RD- Bat Algorithm, for dimension D=3 and 25 independent runs. The value of best value, median of series, worst value, mean value and standard deviation are shown on x-axis and equivalent fitness values are shown on y-axis. Here, best represents minimum fitness value and worst being the maximum fitness value obtained.

In case bat population of 75, optimal results can be obtained over 500 and 750 iterations. It can be concluded that for lesser number of bats, iterations may range up to 500, but for greater number of bats, optimal results can be achieved at earlier stage of exploration and exploitation phase. Results obtained over 500 and 750 iterations are improved by 17%.

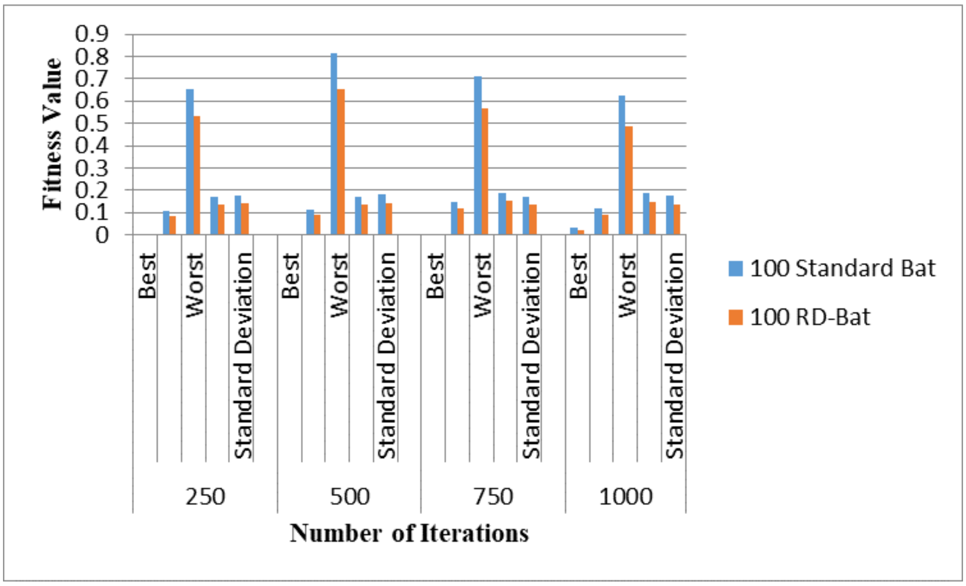


Figure 3.5: For 100 bat population, over [250, 500, 750, 1000] iterations, comparison of Standard Bat Algorithm vs. RD-Bat Algorithm

Over 1000 iterations, best solution can be obtained using 50 bats. Moreover, there is a significant advancement of 18% in the results of RD-BA in comparison to Standard Bat Algorithm.

3.4 Applicability of RD-Bat Algorithm

Load balancing in Cloud Computing environment has gained popularity in recent years. So, here to balance load among virtual machines and to evaluate performance on real world problem, RD-Bat Algorithm is applied. Steps for balancing load among virtual machines in Cloud Computing using Modified Bat Algorithm are mentioned in Algorithm 3.1. There are four main phases of Modified Bat Algorithm, which includes- Initialization of Parameters, Computation of Fitness Value, Selection of Optimal VM and Ensuring optimal VM is not overloaded.

3.4.1 Results & Discussions

To evaluate performance of Modified Bat Algorithm while selecting optimal virtual machine, results are contrasted with results of Standard Bat Algorithm when applied for balancing load in cloud computing scenario. Here number of jobs/tasks to be assigned to available virtual machines are varied over [10, 15, 20] and bat population is varied over [10, 15, 20] for minimum 10 iterations. Results are compared using standard deviation, mean, median, best and worst values of results obtained for Standard Bat Algorithm as well as Modified Bat Algorithm.

Algorithm 3.1: Modified Bat Algorithm

Data: Input number of bats, N and Number of Virtual Machines, V.

Set min_freq, max_freq, velocity, pulse emission rate, loudness and position for entire bat population.

Result: Selection of best suited virtual machine

Begin

For i = 1 to V

Calculate fitness value of every available virtual machine V.

1. Deploy 'N' number of bats, where each bat is responsible for computing the fitness value of V which is present in search space.
2. Every bat will emit pulse and received echo for the computation of distance.
3. Detect the presence of any obstacle. If present, delay, β, and attenuation, α, will affect the solution, as per following equation.

$$Echo_i = (Pulse\_Emitted_i * rand_1) + rand_2$$

else

$Echo_i$= $Pulse\_Emitted_i$+$rand_2$

4. Compute the similarity among sound produced $Pulse\_Emitted_i$ and $Echo_i$ using mathematical function, cross correlation and compute delay samples.

[Correlation]= xcorr($Pulse\_Emitted_i$, $Echo_i$)

DelaySample= Lags(find(Correlation==maximum(Correlation)))

5. Distance can be computed, using DelaySample and TimeSample.
6. Select solution as a best which is having minimum fitness value.

end for

Select optimal virtual machine as local solution, having minimum fitness value.

for i=1 to V

if VM_i == visited VM(i, :)

Increment variable and check for other VM's assigned load.

end if

Rank the best virtual machine and select the best among all.

End

Figure 3.6: Pseudocode Modified Bat Algorithm for Cloud Computing environment

To avoid such a situation, advancement is introduced in Modified Bat Algorithm, where selected optimal virtual machine will not be assigned any task for evaluation, if its existing task count is exceeding threshold value.

Table 3.3: Performance Evaluation of Modified Bat Algorithm-OOVM on the basis of Execution Time

On the basis of Execution Time		Standard Bat Algorithm			Modified Bat Algorithm (OOVM)		
Number of VM	Performance Evaluation Parameters	Bat Population					
		10	15	20	10	15	20
10	Best	3.976	4.25	4.633	4.374	0.499	0.557
	Median	4.058	4.337	4.728	4.633	0.498	0.553
	Worst	4.123	4.407	4.804	4.353	0.524	0.596
	Mean	4.057	4.337	4.727	4.623	0.501	0.527
	Standard Deviation	0.041	0.044	0.048	0.045	0.049	0.006
15	Best	4.187	4.552	4.684	4.405	0.526	0.572
	Median	4.273	4.645	4.78	4.4	0.532	0.579
	Worst	4.342	4.72	4.857	4.676	0.556	0.592
	Mean	4.272	4.644	4.779	4.499	0.537	0.564
	Standard Deviation	0.043	0.047	0.048	0.048	0.006	0.006
20	Best	4.234	4.889	5.142	4.758	0.547	0.617
	Median	4.321	4.989	5.248	4.773	0.602	0.61
	Worst	4.391	5.07	5.333	4.93	0.608	0.63
	Mean	4.32	4.988	5.247	4.052	0.6	0.621
	Standard Deviation	0.044	0.05	0.053	0.044	0.006	0.006

The results of Modified Bat Algorithm- Overloaded Optimal Virtual Machine (OOVM) for overloaded optimal virtual machine on the basis of execution time and cost are shown in Table 3.3 and 3.4 respectively. While evaluating results of Modified Bat Algorithm-overloaded optimal virtual machine variant, it has been observed that it has lesser cost and less variation in values of standard deviation in comparison to results obtained using Standard Bat Algorithm for solving same problem, if 10 bats are used for 10 virtual machines.

Table 3.4: Performance Evaluation of Modified Bat Algorithm-OOVM on the basis of Cost

On the basis of Cost		Standard Bat Algorithm			Modified Bat Algorithm (OOVM)		
Number of VM	Performance Evaluation Parameters	Bat Population					
		10	15	20	10	15	20
10	Best	155.26	165.98	180.91	152.26	162.98	177.91
	Median	156.75	167.11	182.15	153.75	164.11	179.15
	Worst	158.15	168.1	183.23	155.15	165.1	180.23
	Mean	156.76	167.09	182.13	153.76	164.09	179.13
	Standard Deviation	0.1	0.08	0.07	0.09	0.09	0.08
15	Best	163.49	177.76	182.91	160.5	174.77	179.92
	Median	165.05	178.98	184.15	162.06	175.99	181.16
	Worst	166.53	180.03	185.24	163.54	177.04	182.25
	Mean	165.07	178.96	184.14	162.08	175.97	181.15
	Standard Deviation	1.05	0.07	0.07	0.1	0.07	0.07
20	Best	165.36	190.92	200.82	161.52	187.08	196.98
	Median	166.93	192.22	202.19	163.09	188.38	198.35
	Worst	168.43	193.35	203.38	164.59	189.51	199.54
	Mean	166.95	192.2	202.17	163.11	188.36	198.33
	Standard Deviation	1.06	0.07	0.08	1.06	0.07	0.07

In case of 15 virtual machines, one should prefer deployment of 15 bats, as it aims at lesser standard deviation values at lesser cost. For 20 virtual machines, cost increase as number of bats increases. So, optimal solution can be obtained by deploying 15 bats.

Figure 3.7 represents varying values of performance evaluation parameters for 10 virtual machines and varying bat population from 10, 15 and 20. Similarly, Figure 3.8 and 3.9 represents results for 15 and 20 virtual machines for varying bat population, respectively.

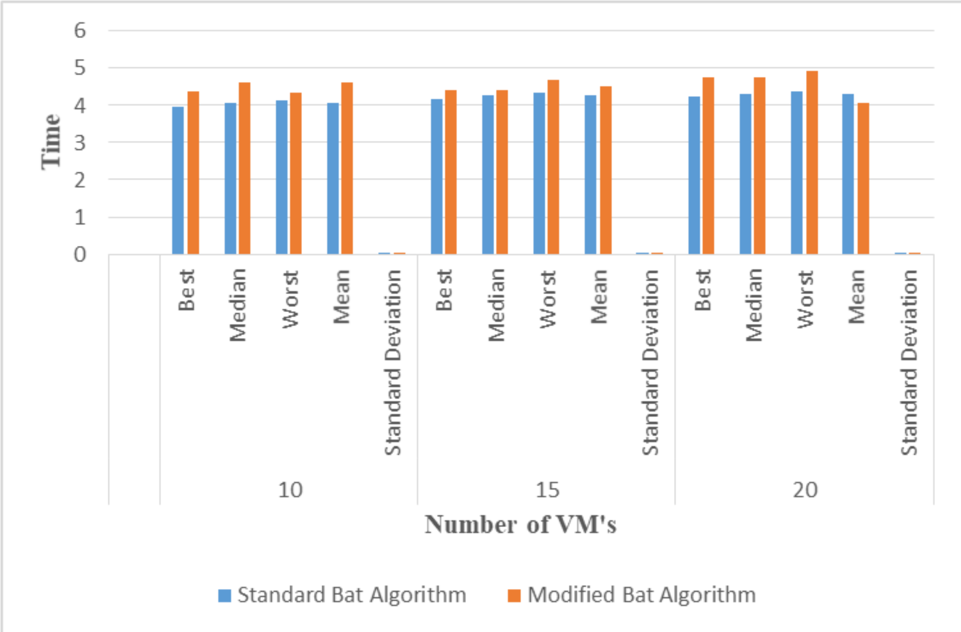


Figure 3.7: Comparison of Execution Time of Modified Bat Algorithm w.r.t. Standard Bat Algorithm for Bat Population=10

It has been noticed that while applying Standard Bat Algorithm for balancing load of VM's, for 10 VM's, if lesser number of bats are deployed, still an optimal result can be obtained with lesser cost, but standard deviation of the obtained results is high. But, when 15 VM's are considered and 10 bats are deployed, it has been noticed that standard deviation is high, but cost is less. If 15 bats are deployed for 15 VM's, standard deviation is reduced but cost is increased, whereas using 20 bats for 15 VM's are not suitable due to increase in cost and standard deviation. So for 15 VM's, 15 bats should be deployed for optimal result.

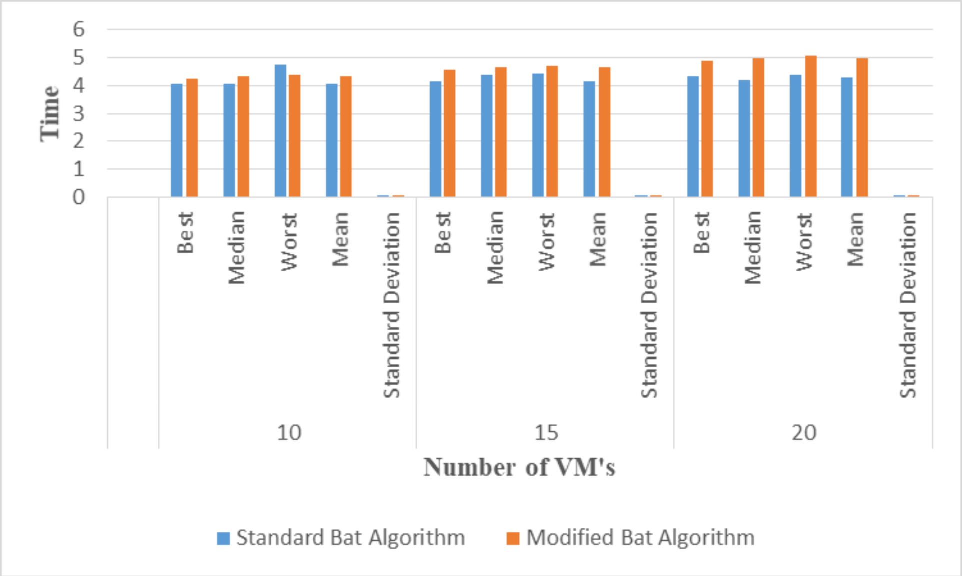


Figure 3.8: Comparison of Execution Time of Modified Bat Algorithm w.r.t. Standard Bat Algorithm for Bat Population=15

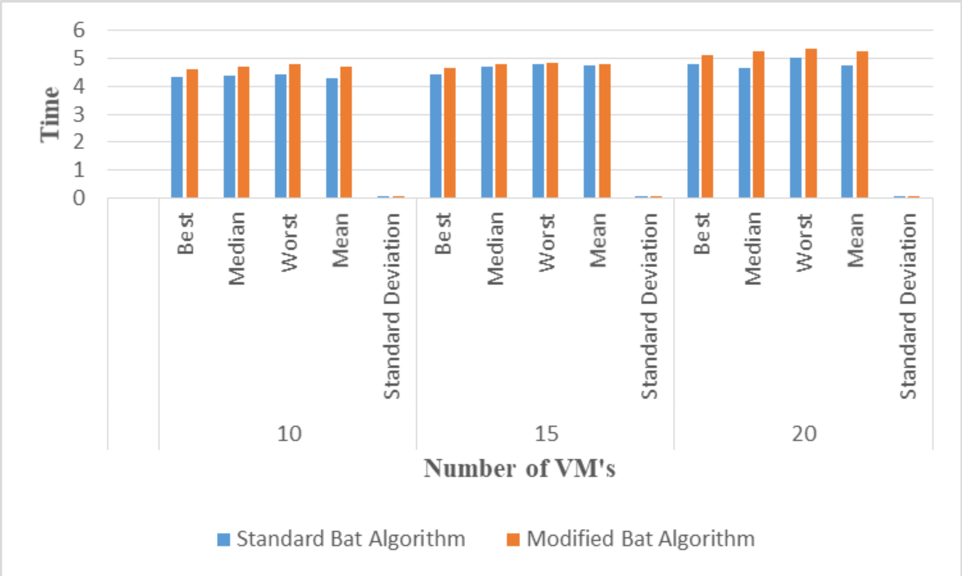


Figure 3.9: Comparison of Execution Time of Modified Bat Algorithm w.r.t. Standard Bat Algorithm for Bat Population=20

Considering 20 VM's and deployment of 10 bats, there will be trade-off between standard deviation and cost. Standard deviation will increase and cost will decrease. If number of bats are increased from 10 to 15 for 20 VM's, then standard deviation will decrease and cost increases. For 20 bats, standard deviation and cost both increases. So, for 20 VM's, 15 bats are suitable to obtain optimal results.

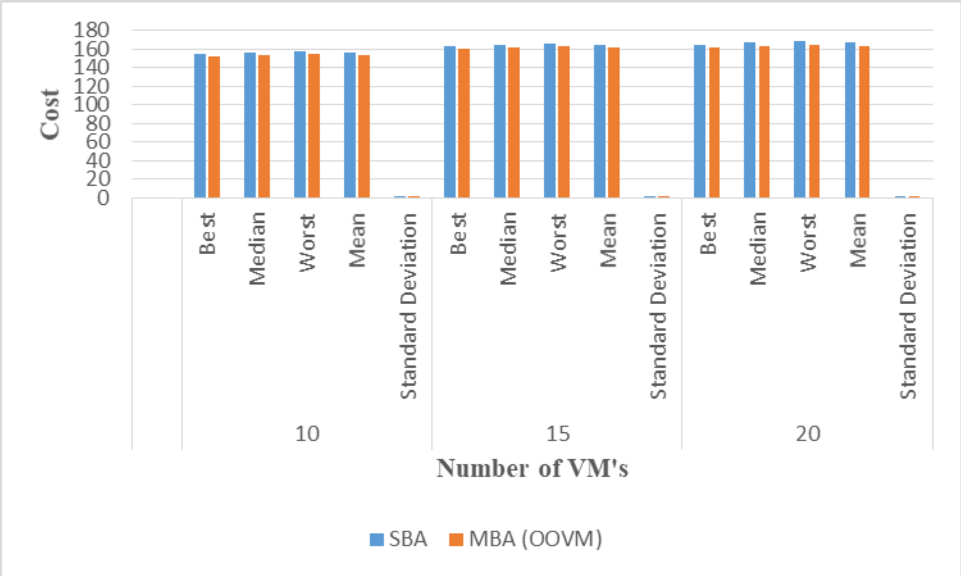


Figure 3.10: Comparison of Cost of Modified Bat Algorithm w.r.t. Standard Bat Algorithm for Bat Population=10

Results evaluated on basis of cost incurred during entire process are depicted in Figure 3.10, 3.11 and 3.12. Figure 3.10 represents varying values of performance evaluation parameters for 10 virtual machines and varying bat population from 10, 15 and 20. Similarly, Figure 3.11 and 3.12 represents results for 15 and 20 virtual machines for varying bat population, respectively.

As depicted in Figure 3.10, cost of selecting optimal virtual machine is lesser than Standard Bat Algorithm. The results which are computed on basis of best, worst, median, mean and standard deviation values by varying number of bats present in bat population and number of virtual machines are represented in graphs. For 10 virtual machines, 15 bats have yielded better results at lesser cost than Standard Bat Algorithm. But, the variation among the solution obtained has not been improved when compared with Standard Bat Algorithm.

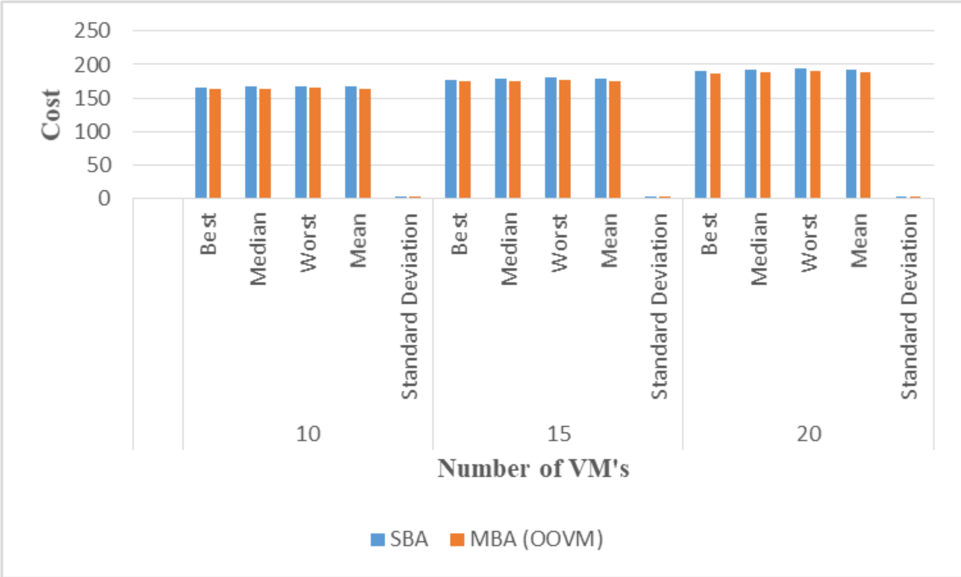


Figure 3.11: Comparison of Cost of Modified Bat Algorithm w.r.t. Standard Bat Algorithm for Bat Population=15

For 15 virtual machines, 10 bats are sufficient to produce efficient results at lesser cost than Standard Bat Algorithm. Moreover, the standard deviation at this step is also improved when compared with Standard Bat Algorithm.

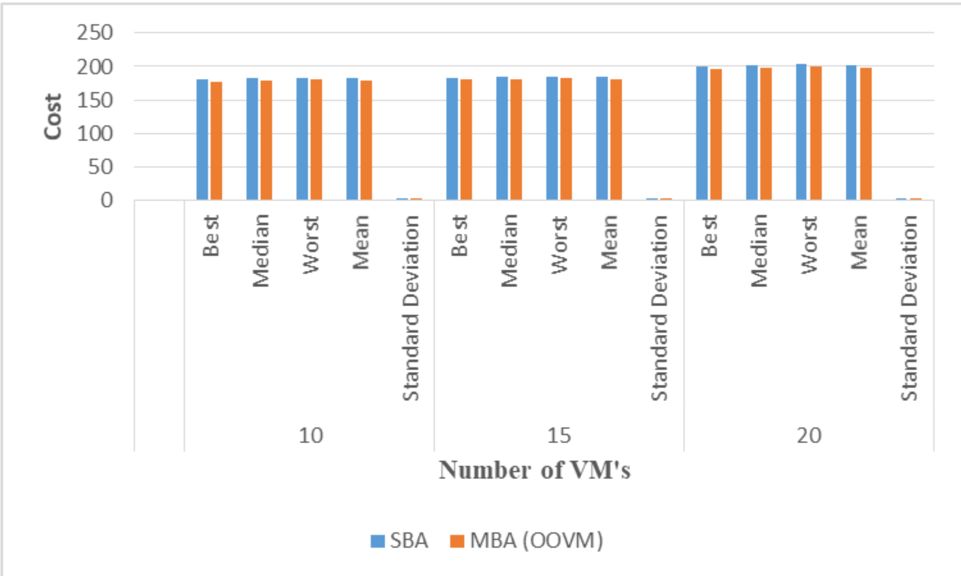


Figure 3.12: Comparison of Cost of Modified Bat Algorithm w.r.t. Standard Bat Algorithm for Bat Population=20

It has been noticed that for 20 virtual machines, 10 bats are able to produce promising results at lesser cost in comparison to Standard Bat Algorithm. Considering the standard deviation, it is same as that in the case of Standard Bat Algorithm.

While evaluating performance on basis of Execution Time, it has been observed that variation among optimal results obtained is reduced and difference between execution time of Standard Bat Algorithm and modified Bat Algorithm-OOVM version is almost negligible up to two decimal points. It is evident from results that Modified Bat Algorithm-OOVM has produced more optimal results in comparison to Standard Bat Algorithm while considering cost as the factor.

3.5 Summary

With rapid advancements introduced in field of IT and necessity of optimization has motivated to develop another variant of Bat Algorithm. Various researchers have contributed in this field, either by including biological features of real bats or by using mathematical functions or by doing hybridization with other algorithms. In this work, inclusion of biological characteristics of real bats is done to provide optimal result. In Standard Bat Algorithm, assumption is to compute distance in "some magical" way. So, motive of this work is to develop such a variant of Bat Algorithm. The main focus is to compute fitness value on the basis of "distance" as parameter, rather than considering any mathematical function. The experimental results show that RD-BA performs in improved fashion as compared to other algorithms. The reason behind this is that RD-BA computes distance and does not rely on any assumption. The results of RD-Bat Algorithm have motivated to explore other biological features of real bats, so that new variants of Bat Algorithm can be developed, which will further improve performance of algorithm and achieve higher level of accuracy while finding optimal solution of problem. An extension to this work is to consider existence of multiple bats to solve problems with multiple objectives. According to bats biological behavior, existence of one bat will affect the targeting behavior of another bat existing in same search space. By considering this behavior of bats, a new variant can also be developed.

CHAPTER 4

DESIGN AND DEVELOPMENT OF BAT ALORITHM VARIANT USING CONSTANT ABSOLUTE TARGET DETECTION STRATEGY

This chapter focuses on second objective of this research work, i.e. "To develop an approach for tracking erratically (unpredictably) moving targets using Constant Absolute Target Detection strategy". This chapter presents investigation of proposed Constant Absolute Target Detection strategy inspired Bat Algorithm (CATD-BA) which is inspired from echolocation behavior and flight behavior of bats.

4.1 Motivation

Bats are a diversified group of microchiroptera and nocturnal types of mammals, having 1100 species around the world, which are capable of continued flight. These bats are capable of flying at 40 kilometers per hour. These bats are only mammals, which are adaptable to different flight strategies and can detect presence of target using echolocation. Based on echolocation, bats compute distance between themselves and prey using time delay estimation technique. Echolocation also helps bats in determining range, angular direction and relative velocity. Bats can also create 3-dimensional picture of surrounding, studied and proved by [73]. The search space of bats depends upon search cone angle, shaped from mouth of bat. Larger the angle, more the search space and hence increases exploration. As per study carried out by [51], it has been proved that bats have capability to differentiate between various objects by echolocation. Another research carried 'out by [81] bats can even perceive target shape by emitted signal. Constant Bearing and Constant Absolute Target Detection are two interception strategies which are followed by bats while targeting preys moving at predictable and erratically unpredictable speed, respectively. The author of [101] has mentioned about these interception strategies in their research work. Literature suggests that avoiding predators, prey capturing and targeting prey having more energy level (optimal selection of prey), are primary cause of evolution of adopting different flight behaviors. Bats target their prey's using either constant bearing (CB) flight strategy or constant absolute target detection (CATD) flight strategy. For instance, to capture prey's moving at predictable speed, bats adopt CB flight strategy and to capture prey's moving erratically, i.e. unpredictable speed, bats adopt CATD flight strategy. This intelligent flight behavior of bats is main source of motivation for proposed algorithm. In this work, pursuit strategy of bats is modeled mathematically to present a new variant of Bat Algorithm for achieving further optimized solution.

4.2 Constant Absolute Target Detection Strategy

Bats have fleeting time frame window within which, they have to detect presence of prey (object), identify the location of its (prey's) presence and capture it. Prey moving erratically and adopting unpredictable flight behavior, build pressure on bat and enforces implementation of such flight strategy which is appropriate to capture such prey and reduces 'time to capture'. For the same, Constant Absolute Target Detection flight strategy is adopted. Till date, there is no flight strategy which offers global minimum intercept time for such erratically moving targets. The concept of CATD is shown in Figure 4.1.

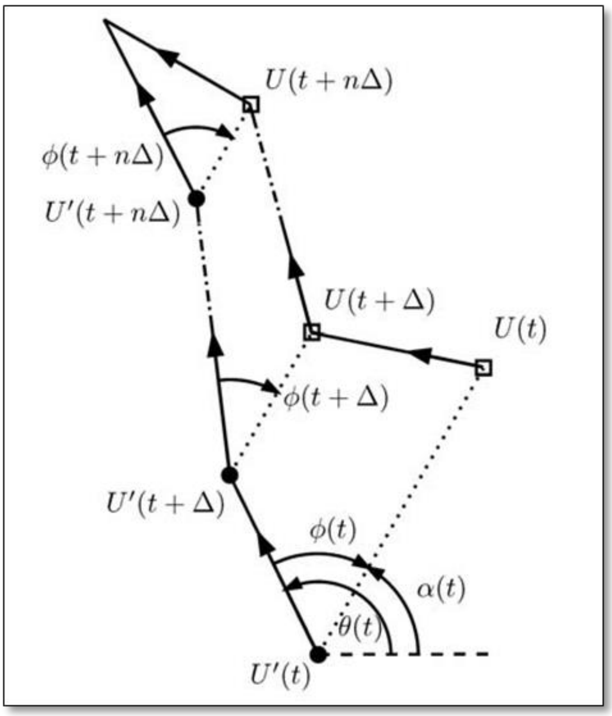


Figure 4.1: CATD strategy to intercept prey [101]

The prey (target) starts at position $U(t)$ which is moving erratically, varying both speed and direction, whereas bat starts at position $U'(t)$. Bat follows time optimum path by adjusting its movement, locally. The bearing lines drawn from bat to prey (target) ($U'(t)$ and $U(t)$) remain parallel to each other, while prey bearing (ϕ) and bat's direction (ϑ) may change continuously. All values of α and ϑ are measured with respect to an external reference.

4.3 New Variant of Bat Algorithm (CATD-BA)

The description of proposed algorithm (CATD-BA: Constant Absolute Target Detection-Bat Algorithm) is categorized into different phases. The explanation of all phases is given below and flow diagram is depicted in Figure 2.

Step 1: Initialization Phase

Given that population of bats is represented by 'M' and population of prey's (targets) is represented as 'N'. Initial population of bats are represented by $\{BP_1, BP_2, BP_3, \ldots\ldots, BP_M\}$ and prey's (targets) population are represented by $\{TO_1, TO_2, TO_3, \ldots\ldots, TO_N\}$. Following factors are considered as the parameters of each bat present in search space:

- Position of each bat present in search space is represented as x_{ip}.
- f_{min} and f_{max} represent the minimum and maximum frequency adopted by bats while capturing targets.
- Frequency of each bat present in search space is represented as f_i.
- Velocity of each bat present in search space is represented as v_{ip}.
- Pulse Emission Rate of each bat present in search space is represented as r_{ip}.
- Loudness of each bat present in search space is represented as a_{ip}.

Parameters of prey (target) include only Velocity and Position and are represented as follows:

1. Velocity of each prey present in search space is represented as v_{ie}.

2. Position of each prey present in search space is represented as x_{ie}.

The parameters of both prey's and bat population are initialized using equations (1)-(5).

$$f_i = f_{min} + (f_{max} - f_{min}) * \alpha \qquad -(1)$$

$$v_{ip}^{t} = v^{t}_{*e} + (x_{ip}^{t} - x_{*p}) * f_i \qquad -(2)$$

$$x_{ip}^{t} = x_{ip}^{t-1} + v_{ip} \qquad -(3)$$

$$v_{ie}^{t} = v_{ie}^{t-1} + (x_{ie}^{t} - x_{*e}^{t}) * \beta \qquad -(4)$$

$$x_{ie}^{t} = x_{ie}^{t-1} + v_{ie}^{t} \qquad -(5)$$

Depending upon movement of prey, selection of appropriate interception strategy during flight is adopted. Two cases are considered: *Case 1:* If prey is moving at predictable speed, use Constant Bearing flight strategy. *Case 2:* If prey is moving erratically (unpredictable speed), use Constant Absolute Target Detection Strategy.

Once movement of prey is determined, appropriate flight strategy is adopted by bat to capture prey. The motive behind adoption of different flight strategies is to minimize time to capture prey. In presence of multiple preys, computation of their fitness value is carried out in next phase.

Step 2: Fitness Evaluation

Bats are considered to be intelligent mammals, which assess energy level (fitness value) of their prey, before capturing them. This evaluation is carried out on basis of energy levels (i.e. fitness value). In this research work, to compute fitness value, distance between bat and prey is considered as factor. Longer the distance, higher will be fitness value and vice versa. To compute fitness value in proposed algorithm, following steps will be carried out:

1): Bats emit pulse and wait for echo.
2): On reception of echo, bats will evaluate received signal and try to identify presence of any obstacle, on the way.
3): In presence of obstacle, echo received will be attenuated and delay will be incurred.

$$Echo_i = (Pulse\ Emitted_i * \alpha) + \beta \qquad -(6)$$

where α represents attenuation factor and β represents delay incurred.

4): In absence of obstacle, delay will be incurred in reception of emitted pulse. But effect of attenuation will be negligible.

Algorithm 4.1: CATD-BA

Data: Initialize bat population as i, position of bat as x_p,

velocity as v_p, loudness as a_p and frequency as f_p.

Initialize target position as x_e and velocity as v_e.

Result: Optimized Solution

Begin

Set maximum number of iterations and represent it using max_iter.

while (curr_iter<max_iter)

Generate new solutions by updating frequency, position and velocity of bat, as mentioned below.

$f_i = f_{min} + (f_{max} - f_{min}) * \alpha$ -(1)

$v_{ip}^t = v^t_{*e} + (x_{ip}^t - x_{*p}) * f_i$ -(2)

$x_{ip}^t = x_{ip}^{t-1} + v_{ip}$ -(3)

Target generates new position to forward in search space, by using following equations.

$v_{ie}^t = v_{ie}^{t-1} + (x_{ie}^t - x_{*e}^t) * \beta$ -(4)

$x_{ie}^t = x_{ie}^{t-1} + v_{ie}^t$ -(5)

if (rand> r_i)

Select the best solution among all solutions.

Generate local solution around the selected best solution.

end if

if ((rand<a_{ip}) && (f(x_{ip}) < f(x_*))

Accept new solution.

Increase r_{ip} and decrease a_{ip}.

end if

Rank the bats and find current best solution x_i.

end while

Post Process results.

End

Figure 4.2: Pseudocode of CATD-BA

$$Echo_i = Pulse\ Emitted_i + \beta \qquad -(7)$$

5): Similarity of emitted pulse and received echo will be computed using concept of cross correlation.

$$[Correlation, Lags] = xcorr(Pulse\ Emitted_i, Echo_i) \qquad -(8)$$

6): Bats may receive multiple echoes for same emitted pulse. To identify accurate echo, bats apply maximization function to similarity index. The echo which satisfies this maximization similarity index, will be considered and corresponding delay samples will be used for computation of time samples, which will be further used to compute distance.

$$Delay = Lags(find(Correlation == max(Correlation))) \qquad -(9)$$

7): Time Samples are computed using frequency samples, as per equation (10).

$$Ts = 1/fs \qquad -(10)$$

Where Ts represents Time Samples and fs represents Frequency Samples.

8): Total time consumed in emitting pulse and receiving echo is computed as per equation given below.

$$time = delay\_samples * Ts \qquad -(11)$$

9): To compute distance between target (prey) and Bat, equation (12) is used.

$$distance = speed * time \qquad -(12)$$

Step 3: Exploring Search Space

During this phase, bats randomly fly in search space and emit pulse to produce novel solutions. These newly generated solutions should fulfil criteria of fitness function. To obtain new solutions, bats have to adjust its parameters, as per the equations (1)-(3) and bats compute fitness value, using equations (6)-(9). As preys are moving erratically, so they have to adjust their own speed and position, as per equation (4) and (5).

Step 4: Generation of New Solutions

Once local optimal solution is obtained, next task of bats is to obtain global optimal solution. For this, bats explore the neighborhood of local solution and update their parameters using below given equations:

$L_i^{t+1}=0.9*L_i^t$ -(13)

$P_i^{t+1}= P_i^t *(1-exp(-1*0.9*i))$ -(14)

During this phase, bats emit pulse randomly, in neighborhood and wait for echo. On reception of echo, if fitness value of randomly selected solution is minimum, than fitness value of local optima, then new solution is accepted and existing local solution is discarded and this way global solution is obtained.

Step 5: Flight Behavior Strategy

Bats adopt different flight strategies depending on maneuver of prey. In the past, two types of flight strategies are adopted by different animals or humans while playing football. Constant Bearing strategy is generally adopted when prey is moving at constant speed (i.e. at predictable speed), whereas Constant Absolute Detection Strategy is adopted when prey is moving erratically (i.e. at unpredictable speed). In this research work, preys are considered to be flying at unpredictable speed, so proposed algorithm has adopted CATD flight strategy to capture prey (i.e. to obtain the optimal solution) in lesser time.

Step 6: Stopping Criteria

To optimize solution, one can either execute program till highest count of iterations or optimum solution is not obtained. In this research work, numbers of iterations are varied over [250, 500, 750, 1000] and optimal solution obtained during these iterations for varied number of bats are discussed in next section.

4.4 Performance Evaluation of CATD-BA

The proposed algorithm (CATD-BA) is implemented using MATLAB. Performance Evaluation of an algorithm is done in two ways. First, algorithm is evaluated for fixed number of iterations. Secondly, algorithm is evaluated till optimal solution of problem is not obtained. Xin She Yang has adopted later case to assess performance of Standard Bat Algorithm. Here, to assess performance of an algorithm, number of iterations is set to be finite. The proposed algorithm is executed for 25 times, while varying number of iterations over [250, 500, 750, 1000] and bat population is varied over [25, 50, 75, 100]. Parameters and their initial values used for describing bat population are mentioned in Table 4.1.

Table 4.1: Parameters of CATD-Bat Algorithm

Parameter	Notation	Value
Bat Population	N	[25 50 75 100]
Pulse Emission Rate	r_i	0.5
Loudness	A_i	0.25
Lower Bound Frequency	f_{min}	0
Upper Bound Frequency	f_{max}	2
Number of Iterations	M	[250 500 750 1000]
Loudness Constant	A	0.9
Pulse Emission Constant	B	0.9
Number of best solution selected	N_{sel}	1

Table 4.2 describes result comparison of Standard Bat Algorithm and CATD-BA. Performance evaluation is carried out by considering mean, median, worst, best and standard deviation (SD) values for diverse bat population and varying count of iterations. Selection of initial values and constants α and β is done considering research work carried out by different researchers and 0.9 value assigned to these constants yield more promising results. The result evaluation is done according to number of bats deployed for obtaining optimal results.

From Table 4.2, it has been concluded that best optimal solution is obtained over 250 iterations for bat population= 25. As number of iterations increase, it will not yield more optimal results. Mean value for 25 bats keep increasing with increase in count of iterations, till count of iterations does not reach 750. But for 1000 iterations, mean value decreases. Lesser the value of standard deviation, lesser is the variation among data samples. Over 250 iterations and for 25 bats, there is minimum value of standard deviation. The standard deviation keeps increasing for 500, 750 and 1000 iterations. The dissimilarities among feasible solutions keep increasing with increase in number of iterations, as bats keep exploring different directions for possible solutions. This analysis suggested that for 25 bats, 250 iterations are sufficient to produce optimal results. So, there is no need to evaluate for more number of iterations.

Table 4.2: Performance Evaluation of Proposed Algorithm (CATD-BA) with respect to Standard Bat Algorithm (BA)

Bat Population		25		50		75		100	
Iterations	Parameters	BA	CATD-BA	BA	CATD-BA	BA	CATD-BA	BA	CATD-BA
250	Best	0.027	0.0218	0.0173	0.01371	0.0072	0.0056	0.0056	0.0042
	Median	0.574	0.4671	0.2663	0.21108	0.2137	0.1663	0.1041	0.0798
	Worst	2.215	1.8173	1.7019	1.371	0.9003	0.7076	0.6517	0.5016
	Mean	0.7218	0.5856	0.4414	0.35125	0.2805	0.2189	0.1679	0.1281
	SD	0.5819	0.4674	0.4616	0.36899	0.2496	0.1953	0.1774	0.1358
500	Best	0.1793	0.1408	0.0028	0.00218	0.0006	0.0005	0.0042	0.0031
	Median	0.4906	0.3893	0.256	0.19936	0.1808	0.138	0.1102	0.0826
	Worst	5.1062	4.0789	1.547	1.20972	0.6557	0.5039	0.8123	0.6194
	Mean	0.7523	0.5978	0.391	0.30488	0.2122	0.1624	0.172	0.1298
	SD	1.8107	1.4411	0.3421	0.26698	0.1521	0.1167	0.1792	0.1365
750	Best	0.1424	0.1117	0.5556	0.42718	0.0017	0.0013	0.004	0.003
	Median	0.5039	0.3992	0.3757	0.2904	0.2048	0.1578	0.148	0.1123
	Worst	3.1917	2.5344	1.6964	1.30992	0.8975	0.7017	0.7108	0.5454
	Mean	0.8852	0.6986	0.5266	0.40439	0.2513	0.1917	0.1893	0.1431
	SD	0.7643	0.6058	0.4858	0.37006	0.2611	0.199	0.1696	0.1292
1000	Best	0.0724	0.0558	0.0166	0.01259	0.0247	0.0184	0.0295	0.0215
	Median	0.4915	0.3774	0.2306	0.17633	0.2278	0.1699	0.1181	0.0867
	Worst	4.0529	3.1689	0.9519	0.73077	1.2308	0.9283	0.6217	0.4599
	Mean	0.7998	0.6201	0.3297	0.25181	0.3136	0.2351	0.1876	0.1378
	SD	0.9104	0.7108	0.2762	0.2114	0.3202	0.2409	0.175	0.129

The solutions obtained after applying Standard Bat Algorithm and Proposed Algorithm over different iterations for 25 bats are depicted in Figure 4.3.

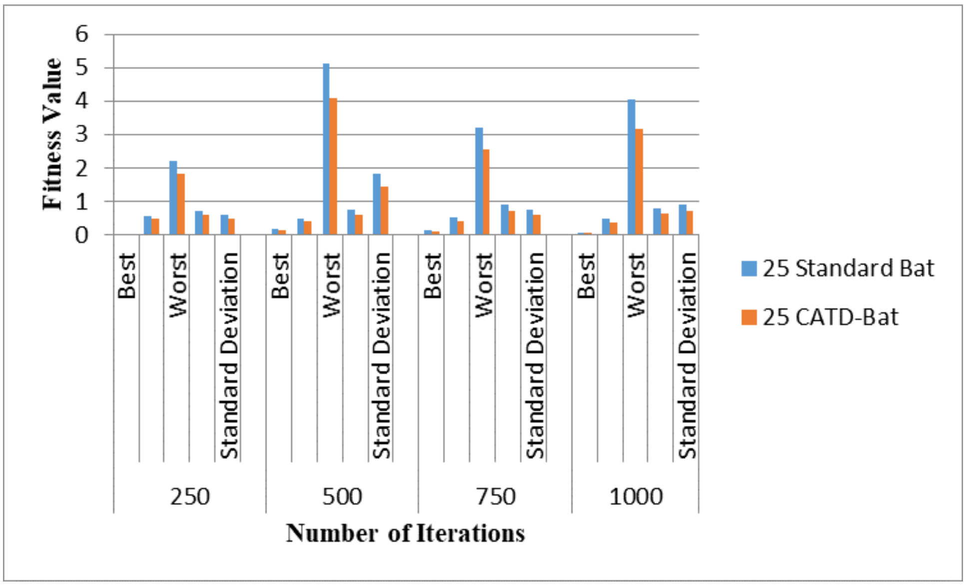


Figure 4.3: Result Evaluation of CATD-BA w.r.t. Standard Bat Algorithm for bat population=25

For bat population 50, best solution is obtained for 500 iterations. If number of iterations goes beyond 500, Standard Bat Algorithm is unable to produce more optimal results. But, for 1000 iterations, algorithm is able to produce similar solutions. In case of proposed algorithm, it has been noticed that for 50 bats, 500 iterations are sufficient to generate optimal and better solutions than Standard Bat Algorithm. There is significant improvement of 20%, which is reflected from results shown in Table 4.2. The corresponding standard deviation of proposed algorithm over 500 iterations is 0.2669. Standard deviation over 1000 iterations is 0.2113. But when compared with standard deviation over 500 iterations, there is a huge difference with respect to standard deviation over 1000 iterations. Considering this difference, usage of 500 iterations for obtaining results seems fruitful for obtaining results. The results are depicted in Figure 4.4.

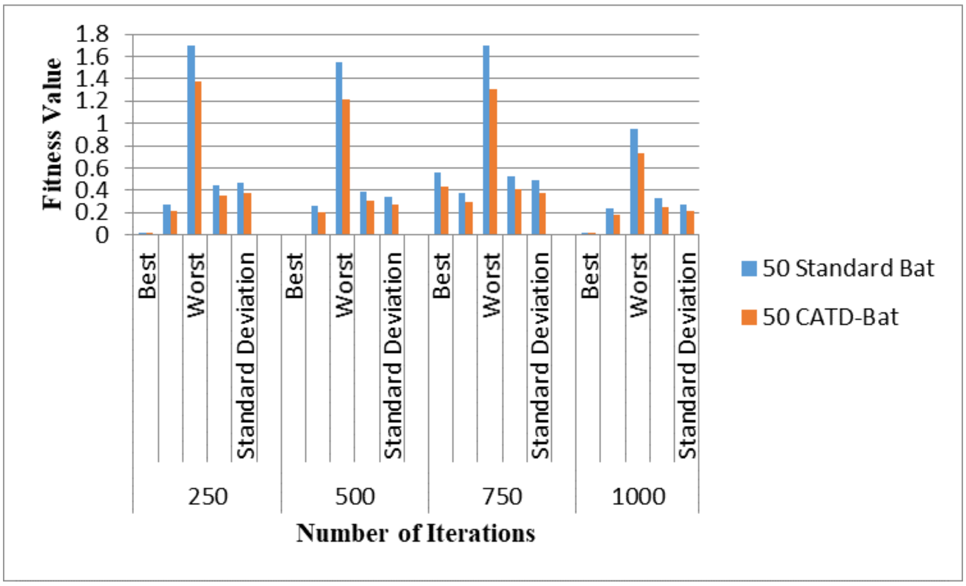


Figure 4.4: Result Evaluation of CATD-BA w.r.t. Standard Bat Algorithm for bat population=50

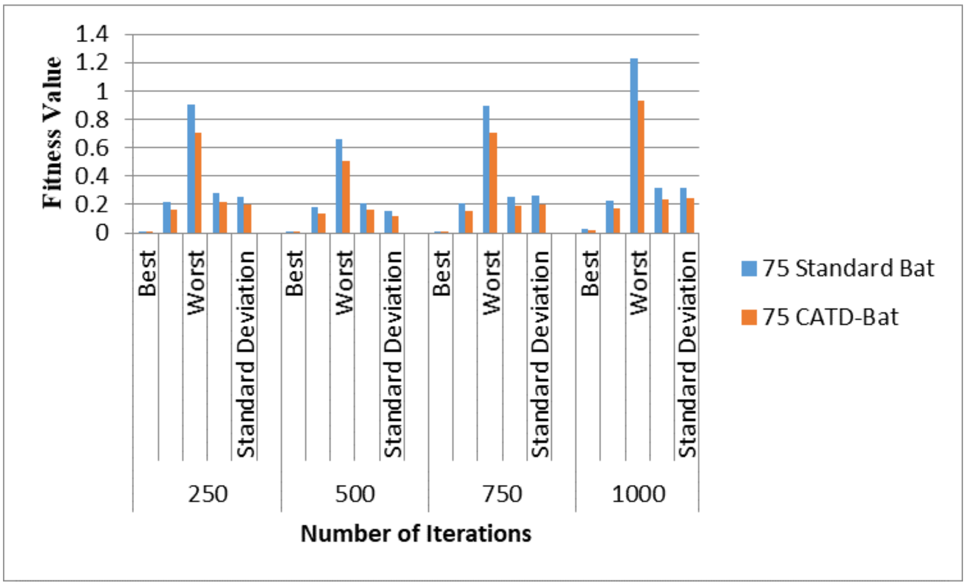


Figure 4.5: Result Evaluation of CATD-BA w.r.t. Standard Bat Algorithm for bat population=75

Considering worst values, 500 iterations are required to obtain optimal results for bat population lies in range of [25, 50, 75]. For selection of optimal solutions, best values are considered in this research work. For 75 bats, 500 iterations serve the purpose of obtaining optimal solution. The graphical format of results for 75 bats are shown in Figure 4.5.

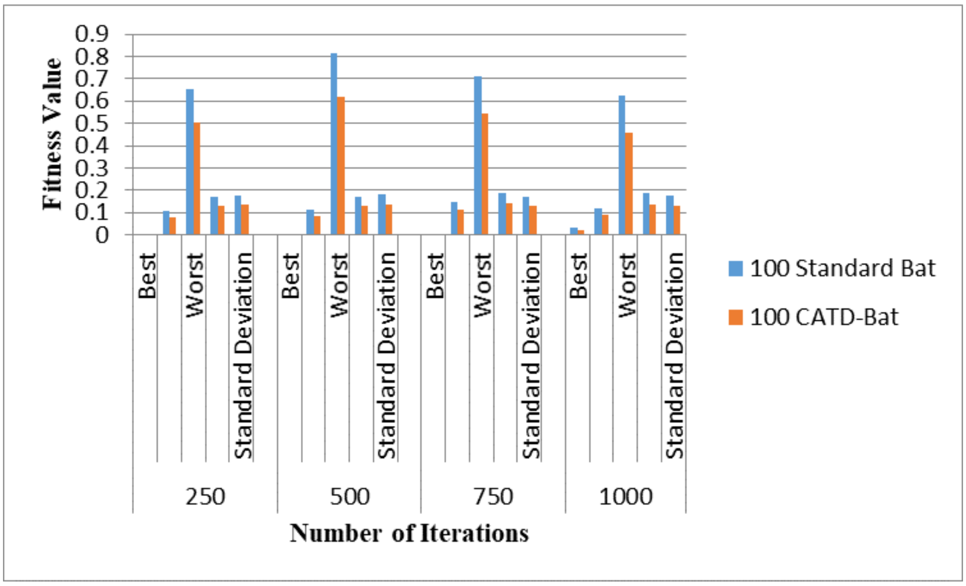


Figure 4.6: Result Evaluation of CATD-BA w.r.t. Standard Bat Algorithm for bat population=100

In consideration to worst values, there is very minute difference in worst case values obtained for 250 and 1000 iterations. The variation among feasible solutions also varies for different number of iterations. For 250 iterations, standard deviation is 0.1774 and for 1000 iterations, standard deviation is 0.1749. Results obtained for 100 bats fail to outperform results obtained by 75 bats. Moreover, standard deviation for different iterations is less varied for different bat population. The results are depicted in graphical format in Figure 4.6.

4.5 Comparison of CATD-BA with RD-Bat Algorithm

Results of CATD-Bat Algorithm are then compared with results obtained using RD-Bat Algorithm, another variant developed in this research work. Table 4.3 depicts results obtained for various parameters like best, median, mean, worst and standard deviation for different bat population varying over different number of iterations.

Table 4.3: Performance Comparison of RD-Bat Algorithm and CATD-Bat Algorithm

Maximum Number of Iterations	Parameters	25		50		75		100	
		RD-Bat	CATD-Bat	RD-Bat	CATD-Bat	RD-Bat	CATD-Bat	RD-Bat	CATD-Bat
250	Best	0.0226	0.0218	0.0143	0.0137	0.0059	0.0056	0.0045	0.0042
	Median	0.4831	0.4671	0.2194	0.2111	0.1744	0.1663	0.0844	0.0798
	Worst	1.8756	1.8173	1.4228	1.3710	0.7417	0.7076	0.5298	0.5016
	Mean	0.6049	0.5856	0.3648	0.3513	0.2295	0.2189	0.1353	0.1281
	SD	0.4823	0.4674	0.3829	0.3690	0.2047	0.1953	0.1434	0.1358
500	Best	0.1462	0.1408	0.0023	0.0022	0.0005	0.0005	0.0033	0.0031
	Median	0.4047	0.3893	0.2074	0.1994	0.1448	0.1380	0.0874	0.0826
	Worst	4.2317	4.0789	1.2561	1.2097	0.5281	0.5039	0.6513	0.6194
	Mean	0.6209	0.5978	0.3169	0.3049	0.1703	0.1624	0.1369	0.1298
	SD	1.4962	1.4411	0.2771	0.2670	0.1223	0.1167	0.1435	0.1365
750	Best	0.1162	0.1117	0.4492	0.4272	0.0014	0.0013	0.0032	0.0030
	Median	0.4134	0.3992	0.3038	0.2904	0.1633	0.1578	0.1174	0.1123
	Worst	2.6254	2.5344	1.3745	1.3099	0.7175	0.7017	0.5681	0.5454
	Mean	0.7237	0.6986	0.4261	0.4044	0.2006	0.1917	0.1501	0.1431
	SD	0.6275	0.6058	0.3933	0.3701	0.2086	0.1990	0.1352	0.1292
1000	Best	0.0585	0.0558	0.0133	0.0126	0.0195	0.0184	0.0228	0.0215
	Median	0.3981	0.3774	0.1862	0.1763	0.1799	0.1699	0.0919	0.0867
	Worst	3.3044	3.1689	0.7668	0.7308	0.9761	0.9283	0.4872	0.4599
	Mean	0.6488	0.6201	0.2657	0.2518	0.2483	0.2351	0.1462	0.1378
	SD	0.7414	0.7108	0.2227	0.2114	0.2540	0.2409	0.1368	0.1290

From Table 4.3, it has been concluded that best optimal solution is obtained over 250 iterations for bat population equal to 100. As number of iterations increase, it will not yield more optimal results. Mean value for 25 bats provides promising result over 250 iterations. But for 1000 iterations and bat population equal to 50, optimal mean value can be obtained, i.e. 0.2518. Lesser the value of standard deviation, lesser is the variation among data samples. Over 250 iterations and for 100 bats, there is minimum value of standard deviation, i.e. 0.136. The standard deviation keeps increasing for 500, 750 and 1000 iterations. The dissimilarities among feasible solutions keep increasing with increase in number of iterations, as bats keep exploring different directions for possible solutions. This analysis suggested that for 25 bats, 1000 iterations are sufficient to produce optimal results. So, there is no need to evaluate for more number of iterations. Moreover for 75 bats, 500 iterations will be sufficient to produce optimal solution. For 100 bats over 1000 iterations, best solution can be obtained, within standard deviation of 0.129. The solutions obtained after applying Standard Bat Algorithm and Proposed Algorithm over different iterations for 25 bats are depicted in Figure 4.7.

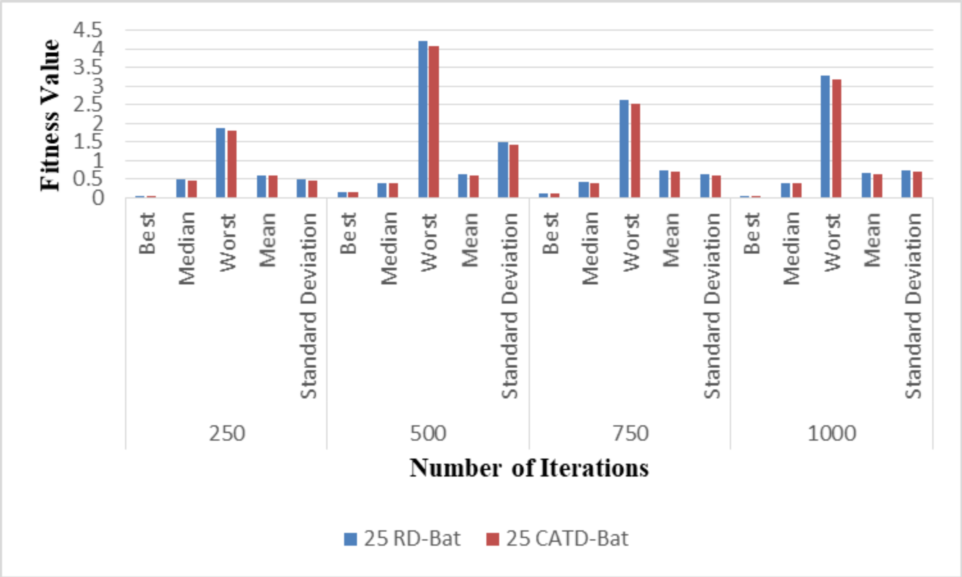


Figure 4.7: Result Evaluation of CATD-BA w.r.t. RD-Bat Algorithm for bat population=25

For bat population 50, best solution is obtained for 500 iterations. If number of iterations goes beyond 500, Standard Bat Algorithm is unable to produce more optimal results, which satisfy minimization criteria of fitness functions. In case of CATD-BA, it has been noticed that for 50 bats, 1000 iterations are required to generate optimal and better solutions than Standard Bat Algorithm. There is significant improvement of 4%, which is reflected from results shown in Table 4.3. The corresponding standard deviation of proposed algorithm over 500 iterations is 0.267. Standard deviation over 1000 iterations is 0.2114. But when compared with standard deviation over 500 iterations, there is very minute difference with respect to standard deviation over 1000 iterations. Considering this difference, usage of 500 iterations for obtaining results seems fruitful for obtaining the results. The results are depicted in Figure 4.8.

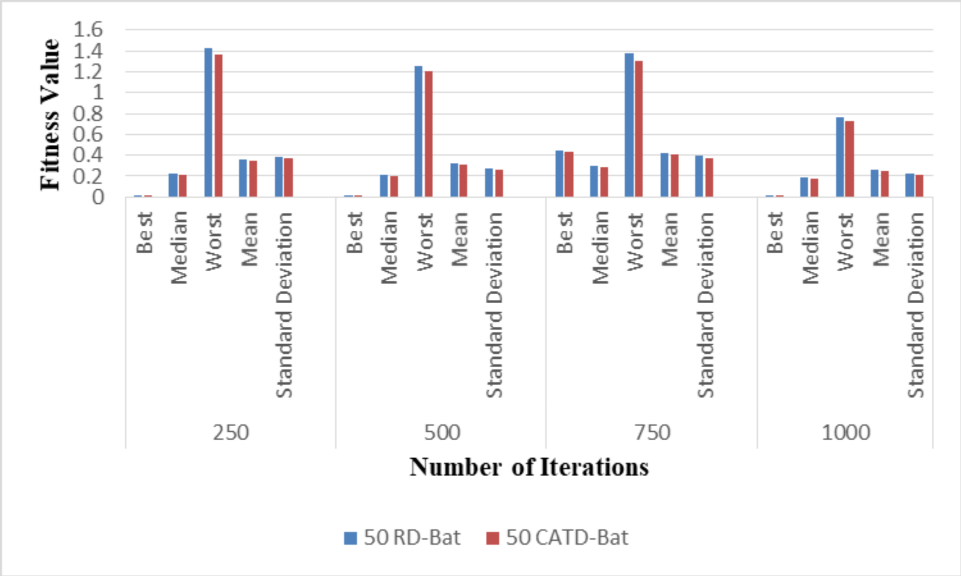


Figure 4.8: Result Evaluation of CATD-BA w.r.t. RD-Bat Algorithm for bat population=50

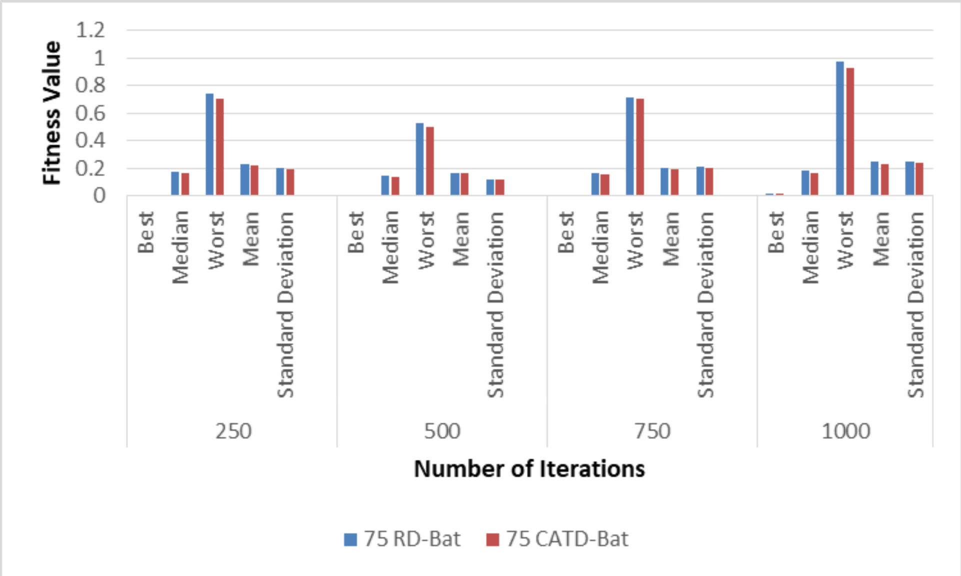


Figure 4.9: Result Evaluation of CATD-BA w.r.t. RD-Bat Algorithm for bat population=75

Considering worst values, 750 iterations are required to obtain optimal results for bat population lies in range of [25, 50, 75]. For selection of optimal solutions, best values are considered in this research work. For 75 bats, 750 iterations serve the purpose of obtaining optimal solution. The graphical format of results for 75 bats is shown in

Figure 4.9.In consideration to worst values, there is very minute difference in worst case values obtained for 250 and 1000 iterations. The variation among feasible solutions also varies for different number of iterations. For 250 iterations, standard deviation is 0.19532 and for 1000 iterations, standard deviation is 0.19896. Results obtained for 100 bats fail to outperform results obtained by 75 bats. Moreover, standard deviation for different iterations is less varied for different bat population. The results are depicted in graphical format in Figure 4.10.

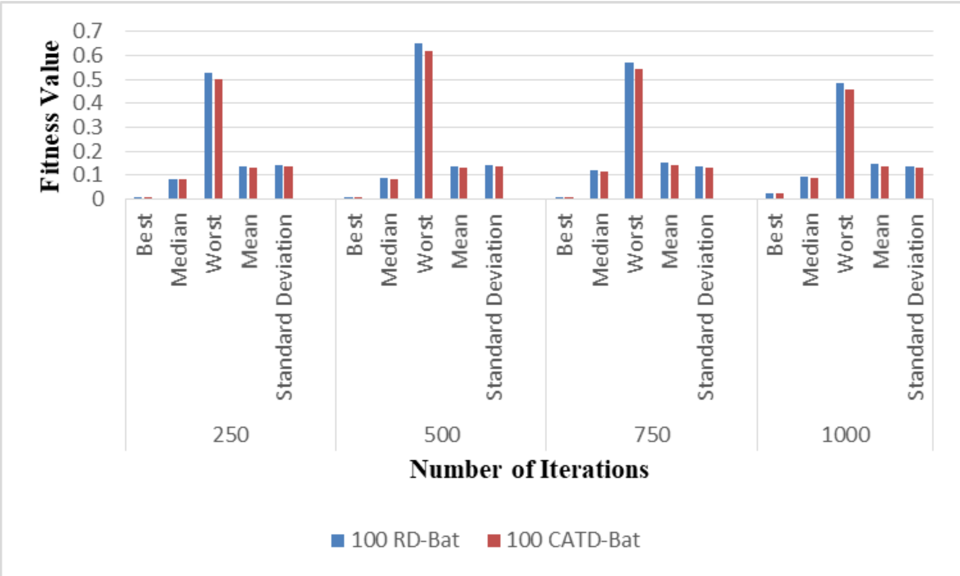


Figure 4.10: Result Evaluation of CATD-BA w.r.t. RD-Bat Algorithm for bat population=100

4.6 Applicability of CATD-BA in Wireless Sensor Networks

This section proposes and presents outcome of applicability of CATD-Bat Algorithm for providing route to packets in Wireless Sensor Networks, while avoiding congestion on optimized route. Some of the key areas like selection of optimal path, energy conservation, efficient routing techniques and use of swarm intelligence techniques are gaining lot of researchers' attention. It has been noticed that routing has been improved with the use of optimization techniques such as PSO, ACO and BA. However Bat Algorithm is as powerful as Particle Swarm optimization and Genetic algorithms and considered to be special cases of Bat Algorithm.

Algorithm 4.2: Congestion Avoidance Algorithm

```
Input number of packets, N
Begin
        For each packets
                For i = 1 to N
                [Start, End, Cost, Visited Node ] = Call_BAT ()
        End
        Avoiding traffic by suggesting other route to packets
        For i = 1 to N
                Check = Start ( i, : )
                For j = 1 to N
                        If check == start ( j,: )
                                Counter ++;
                                Check for other start node
                        Endif
                Endfor
        Endfor
        Optimized route by calling Call_BAT( )
        Display Cost for each route assigned to each packet
End
```

Figure 4.11: Pseudocode of Congestion Avoidance using CATD-Bat

In this work, main focus is to avoid congestion while selecting shortest path using Bat Algorithm. To solve routing problem and minimize energy consumption, various optimization techniques is preferred by researchers. Node deployment, Energy efficiency and network lifetime are main challenges of wireless sensor networks which is overcome by optimization techniques. The emphasis is to ensure that no such delay incur or to reduce delay. Another aspect of this research work is to reduce control packets and specifying size of queue in network. To find prey, bat emits pulse in different directions and waits for echo. After that, bat follows direction in which it observes less fitness value and follows path in which possibility of more food is there.

However congestion may occur in chosen path. Due to congestion, more delay will be there and load on single path will be increased due to traffic as well as response time. To overcome above problem, upper limit of usage of same path will be enforced in congestion avoidance algorithm for routing in wireless sensor network.

Algorithm 4.3: Bat Algorithm for Routing

```
Input number of bats, B and number of nodes
Begin
    For all nodes, initialize distance,
        For i = 1 to X
            Compute distance/Cost between cities using Bat Algorithm
            Assign similar cost to pair of nodes i.e,
            Cost ( A → B) = Cost ( B → A )
            Check for visited node
            If  selected node = visited node ( i , : )
                    Select another node = start node
            End if
            Explore more solution using exploitation phase of Bat Algorithm and
            return starting node, ending node and cost associated.
    End for
End
```

Figure 4.12: Pseudocode for Routing in WSN

While implementing proposed algorithm, count of packets and count of nodes are varied to assess performance of proposed algorithm. It has been observed that in search of optimal route, time taken by packets, will increase as count of nodes increases. The standard deviation depicts preciseness of algorithm. As discussed earlier to avoid congestion, threshold value is set for usability of proper path. If usability factor attains its maximum to threshold value, packets will be re-routed over new path. As number of same source node increases, it is quite natural that packets try to use same optimal path to minimize cost but at the same time, there is a need to avoid congestion over same path by rerouting packets which make network more efficient.

4.6.1 Result Evaluation

Table 4.4 and 4.5 depicts improvement in result of proposed work on basis of best, worst, mean and median for fixed number of packets 10 & 15 with varying count of nodes [10,15,20].

Table 4.4: Time comparison of CATD-BA inspired routing algorithm vs. Standard Bat Algorithm based routing algorithm for 10 packets

Time (in Seconds)	Nodes- 10, Packets-10		Nodes- 15, Packets-10		Nodes- 20, Packets-10	
	BA	CATD-BA	BA	CATD-BA	BA	CATD-BA
Best	2.13	1.97	12.50	12.29	48.56	39.12
Median	2.30	2.01	13.08	12.53	49.16	46.35
Worst	2.73	2.18	13.78	14.02	54.03	56.16
Mean	2.32	2.05	13.16	12.73	49.70	46.70
Standard Deviation	0.17	0.08	0.42	0.52	1.57	2.37

Here, time taken by Standard Bat Algorithm used for congestion avoidance and time taken by CATD-BA inspired Routing Algorithm, to obtain optimal results are considered. Figure 4.13 and Figure 4.14 represents graphical form of data as mentioned in Table 4.4 and Table 4.5. As it is clearly visible, as count of nodes increase, there is a significant change in time interval to find optimal route.

Table 4.5: Time comparison of CATD-BA inspired routing algorithm vs. Standard Bat Algorithm based routing algorithm for 15 packets

Time (in Seconds)	Nodes- 10, Packets-15		Nodes- 15, Packets-15		Nodes- 20, Packets-15	
	BA	CATD-BA	BA	CATD-BA	BA	CATD-BA
Best	3.36	3.02	18.87	17.12	71.92	69.88
Median	3.42	3.16	19.64	18.06	73.53	71.25
Worst	3.56	3.45	20.05	23.07	74.52	78.58

Mean	3.43	3.17	19.57	19.15	73.36	72.01
Standard Deviation	0.07	0.14	0.37	2.67	0.94	3.32

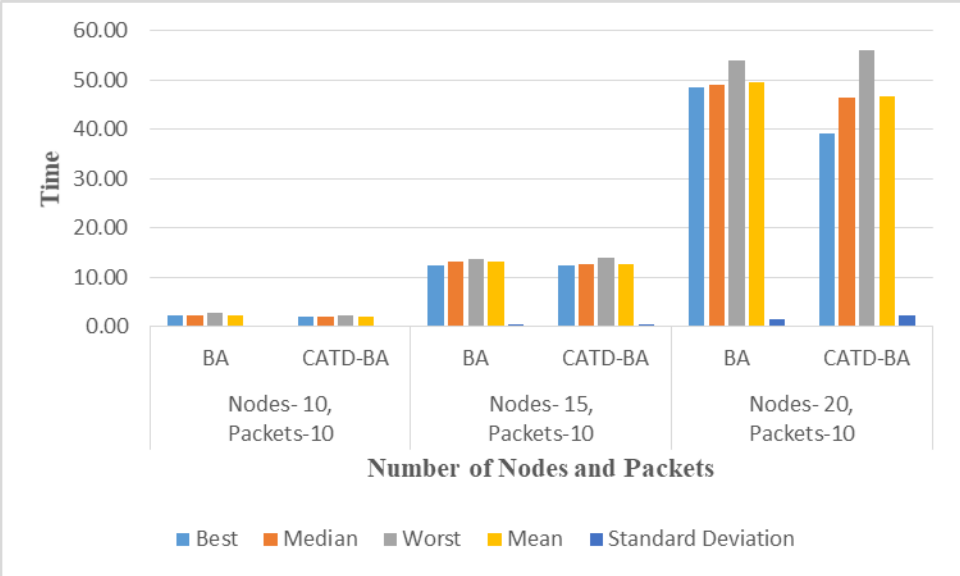


Figure 4.13: Graphical Representation of Comparison of Standard Bat Algorithm w.r.t. CATD-BA inspired Routing Algorithm, considering Time as factor for 10 packets

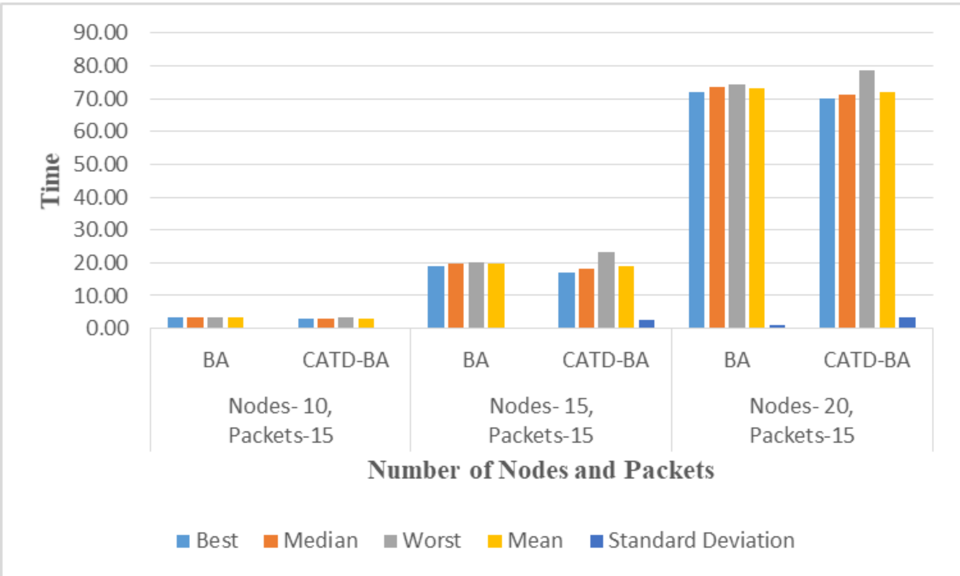


Figure 4.14: Graphical Representation of Comparison of Standard Bat Algorithm w.r.t. CATD-BA inspired Routing Algorithm, considering Time as factor for 15 packets

In Table 4.6 and 4.7, it has been observed that in search of optimal route, cost will increase as number of nodes increase. As bats have to explore more number of optimal nodes, which will lead to increase in cost factor. However, cost reflecting is minimal, as optimal path is traced by nature inspired algorithm.

Table 4.6: Cost comparison of Standard Bat Algorithm w.r.t. CATD-BA inspired Routing Algorithm for 10 Packets to explore different nodes

Distance/Cost	**Nodes- 10, Packets-10**		**Nodes- 15, Packets-10**		**Nodes- 20, Packets-10**	
	BA	CATD-BA	BA	CATD-BA	BA	CATD-BA
Best	250.35	143.65	243.05	186.79	239.94	240.83
Median	280.24	156.80	287.72	216.87	276.56	267.69
Worst	305.93	176.70	348.85	293.48	481.26	365.67
Mean	280.39	158.36	288.97	220.34	293.18	291.08
Standard Deviation	17.35	12.40	37.21	29.93	70.71	48.34

Table 4.7: Cost comparison of Standard Bat Algorithm w.r.t. CATD-BA inspired Routing Algorithm for 15 Packets to explore different nodes

Distance/Cost	**Nodes- 10, Packets-15**		**Nodes- 15, Packets-15**		**Nodes- 20, Packets-15**	
	BA	CATD-BA	BA	CATD-BA	BA	CATD-BA
Best	404.14	224.71	398.26	300.28	389.72	368.64
Median	430.01	236.08	413.24	327.85	418.42	413.38
Worst	451.10	260.19	467.87	368.17	573.48	498.42
Mean	428.60	240.45	420.00	325.10	432.95	418.67
Standard Deviation	15.24	12.18	22.60	21.20	51.24	40.91

It has been observed that cost is minimal in case of 10 packets for 10 nodes, in comparison to 15 and 20 nodes.

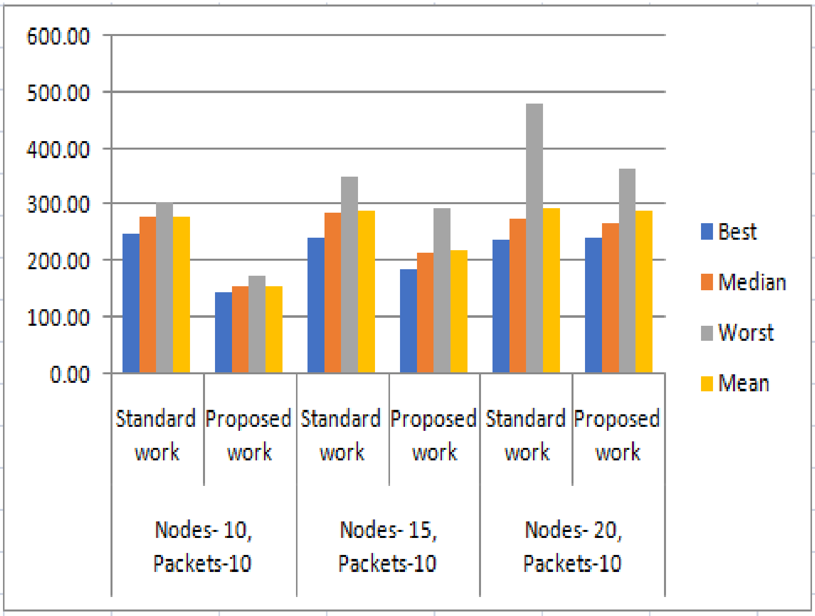


Figure 4.15: Graphical Representation of Comparison of Standard Bat Algorithm w.r.t. CATD-BA inspired Routing Algorithm, considering Cost as factor for 10 packets

For 15 packets, minimal cost is incurred in case of 10 nodes. It can be concluded that, as the number of nodes increases irrespective of the packet count, the cost increases. Considering total cost associated among different routes, while solving problem at hand and results computed, graphical representation is depicted in Figure 4.15 and 4.16.

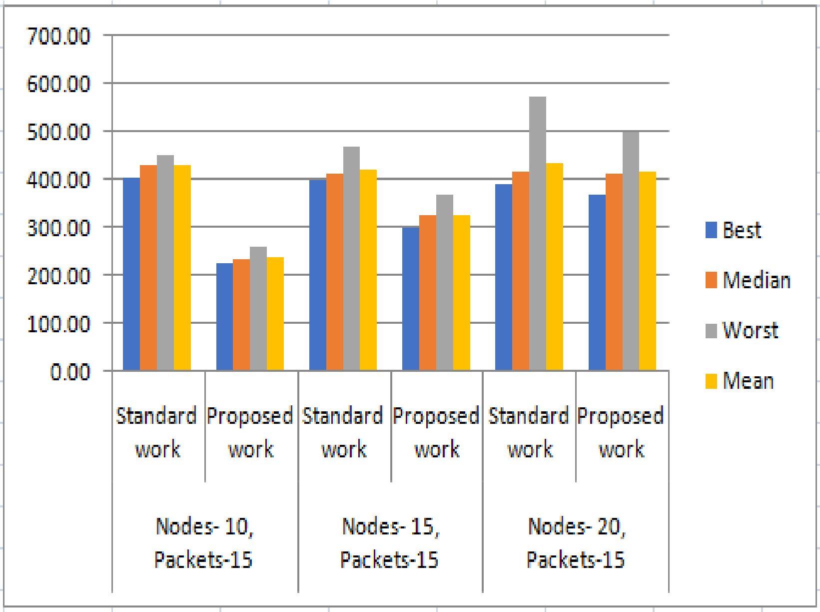


Figure 4.16: Graphical Representation of Standard Bat Algorithm w.r.t. CATD-BA inspired Routing Algorithm, considering Cost as factor for 15 packets

Here main focus is to overcome congestion problem as it causes major problems like packet loss, transmission delay and more energy consumption. This work primarily focused and found optimal route in wireless sensor network. The experimental results were compared with similar works on Bat Algorithm by other researchers considering best, median, worst and mean parameters for time and cost factor. It has been found that proposed algorithm is more cost effective when used for solving routing problem in WSN environment. It also proved to be more efficient than others as it will minimize the congestion and find best route at minimal cost.

4.7 Summary

A novel variant of Bat Algorithm, which is inspired from bat's flight behavior, is designed for solving combinatorial problems. The different flight behavior adopted by microchiroptera bats is studied and modelled mathematically. The proposed algorithm is tested and results validation is done in comparison to Standard Bat Algorithm. It is quite evident from comparative analysis that proposed algorithm explore wide variety of solutions, before obtaining local optimal solution. The proposed algorithm is able to generate more promising results when equated w.r.t. to Standard Bat Algorithm. The proposed work can be extended to solve multi-objective optimization problems. To develop other variants of Standard Bat Algorithm or to improve performance of proposed algorithm, other biological features of bat can also be explored.

CHAPTER 5
ADOPTION OF PURSUIT STRATEGIES FOR PERFORMANCE IMPROVEMENT IN BAT ALGORITHM

This chapter focuses on third objective of this research work, i.e. "To design movement strategy for a target seeker in the presence of multiple target seekers". This chapter presents investigation of proposed strategies adopted while targeting prey, when bat is surrounded by multiple bats in search space.

5.1 Inspiration

Bats are quantitatively analyzed in past for their astonishing echolocating behavior and flight behavior. Many experiments have been conducted in past to analyze biological behavior of paired big brown bats, as mentioned in [34] and [35]. The author has conducted experiments by keeping big brown bats in big test center competing with other bats while targeting same food source. The author has categorized bats in two categories: leader and follower. During experiments, conducted in closed laboratory, it has been noticed that, most of times follower was able to gain access to food source, before leader. Moreover, if both bats are flying in direction of each other and guide their sonar beams away from each other, to avoid signal jamming of both bats. In another research work, author has suggested that bats seize their vocalization, to avoid signal jamming. Even, another aspect of this is to capture information from other bats' without producing sound beams. Bats actually adopt strategy of 'silence' when surrounded by other bats. In this research work, behavior of bats' is adopted to refine results obtained using Bat Algorithm.

5.4 Proposed Strategies

The flight behavior of bat population is analyzed in this research work. As mentioned in previous section, bats either act as follower or leader, in case two or more bats are following each other. Otherwise, bats may either converge or diverge with respect to movement of other bats. The direction of movement is categorized as following, converging or diverging as per angle difference between two bats' head movement and the angle between two bats velocity factor. One bat is said to be follower of another, if one bat is flying behind another and in same direction. For this type of pursuit strategy, inter-bat angle should be less than 90 degree. Another type of pursuit strategy is converging. During converging, two bats fly towards each other, where angle of each bats' movement should be less than 90 degree. Another pursuit strategy is diverging, during which bats fly away from each other. The benefits of follower-leader strategy are quite visible. The relative position of two bats affects process of obtaining optimal solution. The bat fly in lead may be able to access optimal solution (target/prey) earlier as compared to follower. Whereas, follower always has advantage of tracking movement of leader, without utilizing its own energy/resources. Consider two bats are present in search space and also in neighborhood of each other. In order to avoid jamming of sounds produced, both bats will enter in silent phase and vocalization phase for different time period, as depicted in Figure 5.1, as per study carried out in [34]. Moreover, number of bats present in search space, also effects pursuit strategies opted by bats and their conspecifics.

Bat B	**Bat A**	
	Silence >0.2 seconds	**Vocalization**
Silence >0.2 seconds	Silent Phase	Silent Phase
Vocalization	Silent Phase	Vocalization Phase

Figure 5.1: Silent and Vocalization Phase

In Figure 5.2, Θ_1 is angle between two different bats' velocity parameter, whereas Θ_2 represents separation vector, where Θ_1 denotes angle for one bat and Θ_2 denotes angle for another bat. The vocalizing bat is depicted with dark black color in Figure 5.2, which is producing sound.

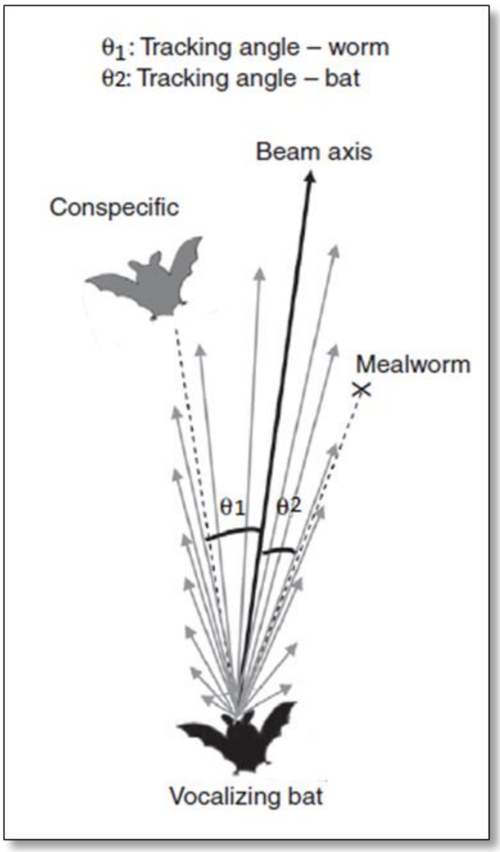
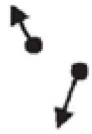
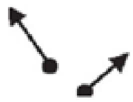
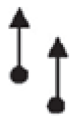


Figure 5.2: Bat producing Sonar Beam in the presence of Conspecifics [34]

The tracking angle with respect to bat is angle between vocalizing bat and other bat present in neighborhood. Tracking angle with respect to mealworm (prey/target) is angle between meal worm and axis of sonar beam. Two bats are said to be having ‘following’ as pursuit strategy, if inter-bat angle is acute angle, considering Θ_1 greater or equal than 90 degree and Θ_2 less than 90 degree. If acute angle lies in range of 0 degree to 30 degree, then inter-bat angle also lies in range of 0 degree and 30 degree. If acute angle lies in range of 30 degree to 60 degree, then inter-bat angle also lies in range of 30 degree and 60 degree. If acute angle lies in range of 60 degree to 90 degree, then inter-bat angle also lies in range of 60 degree and 90 degree. In case of converging, both bats form acute angles with respect to common external surface, which leads to formation of inter-bat angle between 0 degree and 180 degree.

Table 5.1: Types of Pursuit Strategies

Flight Behavior	Θ_1 and Θ_2	Inter-Bat Angle (α)	Movement Direction
Diverging I	Θ_1>=90 degree and Θ_2<90 degree OR Θ_2>=90 degree and Θ_1<90 degree	α>90 degree	
Diverging II	Θ_1>=90 degree and Θ_2>=90 degree	0 degree<= α<=180 degree	
Following	Θ_1>=90 degree and Θ_2<90 degree OR Θ_2>=90 degree and Θ_1<90 degree	α<90 degree	
Converging	Θ_1<90 degree and Θ_2<90 degree	0 degree <= α<=180 degree	

Algorithm 5.1: Flight behavior inspired Algorithm

Data: Initialize bat population as i, position of bat as x_p, velocity as v_p, loudness as a_p and frequency as f_p.

Initialize target position as x_e and velocity as v_e.

Result: Optimized Solution

Begin

Set maximum number of iterations and represent it using max_iter.

while (curr_iter<max_iter)

Generate new solutions by updating frequency, position and velocity of bat, as mentioned below.

$f_i=f_{min}+ (f_{max}-f_{min})*\alpha$ -(1)

$v_{ip}^{t}=v^{t}_{*p} + (x_{ip}^{t} - x_{*p}) *f_i$ -(2)

$x_{ip}^{t}=x_{ip}^{t-1} +v_{ip}$ -(3)

Target generates new position to forward in search space, by using following equations.

$v_{ie}^{t}= v_{ie}^{t-1} + (x_{ie}^{t} - x_{*e}^{t})*\beta$ -(4)

$x_{ie}^{t}= x_{ie}^{t-1} +v_{ie}^{t}$ -(5)

if (i==2)

$v1*Cos\theta_1=- v2*Cos\theta_2 + \frac{\vec{r}_{12}}{dt}$ -(6)

endif

if (i>2)

Select ‘competitor’ bat and ‘leader’ bat.

endif

if (rand> r_i)

Select the best solution among all solutions.

Generate local solution around the selected best solution.

$\theta_2=cos^{-1}\left[\frac{-v_1*cos\theta_1+distance}{v_2}\right]$ -(7)

end if

if ((rand<a_{ip}) && (f(x_{ip}) < f(x_*))

Accept new solution.

Increase r_{ip} and decrease a_{ip}.

end if

Rank the bats and find current best solution x_i.

end while

Post Process results.

End

Figure 5.3: Psuedocode of Flight behavior inspired algorithm

Two cases are described in diverging, i.e. diverging case I and diverging case II. Diverging case I denotes such a situation where both bats form obtuse angles and results into such an inter-bat angle which lies in range of 0 degree and 180 degree. Diverging case II denotes such a scenario where one bat forms acute angle, whereas another bat forms obtuse angle and results into inter-bat angle which lies in range of 0 degree and 180 degree. But, it has been noticed during the experiments that 'following' is the most common pursuit strategy adopted by bats, when present in the neighborhood of other bats/conspecifics. This following pursuit strategy is adopted 65% of the times, which is even more than half of the time period.

While targeting optimal solution, basis of identifying type of pursuit strategy to adopt, is inter-head bat angle between two bats. The three types of pursuit strategies, i.e. following, converging and diverging are described in Table 5.1. Based on pursuit strategies, pseudocode for achieving third objective is depicted in Figure 5.3.

5.5 Performance Analysis of Strategies adopted

Flight behavior inspired Bat Algorithm (FBI-BA) is implemented using MATLAB. Performance evaluation of an algorithm can be done in two ways. Firstly, algorithm can be evaluated for fixed number of iterations. Secondly, algorithm can be evaluated till optimal solution of problem is not obtained. Xin She Yang has adopted later case to assess performance of Standard Bat Algorithm. Here, to assess performance of an algorithm, number of iterations is set to be finite. The FBI-BA is executed for 25 times, while varying number of iterations over [250, 500, 750, 1000] and bat population is varied over [25, 50, 75, 100].

Table 5.2: Result Evaluation of Flight behavior inspired Algorithm w.r.t. Standard Bat Algorithm

Maximum Number of Iterations	Parameters	Bat Population							
		25		50		75		100	
		BA	FBI-BA	BA	FBI-BA	BA	FBI-BA	BA	FBI-BA
250	Best	0.0050	0.0085	0.0017	0.0009	0.0003	0.0008	0.0000	0.0029
	Median	0.0302	0.0758	0.0274	0.0117	0.0109	0.0109	0.0040	0.0073
	Worst	0.4279	0.1584	0.1256	0.0405	0.0534	0.0495	0.0425	0.0482
	Mean	0.0810	0.0681	0.0341	0.0187	0.0148	0.0203	0.0116	0.0140
	SD	0.1162	0.0421	0.0342	0.0133	0.0244	0.0190	0.0249	0.0131
500	Best	0.0021	0.0010	0.0014	0.0020	0.0001	0.0004	0.0005	0.0004
	Median	0.0185	0.0108	0.0063	0.0085	0.0076	0.0064	0.0034	0.0039
	Worst	0.0361	0.0511	0.0272	0.0466	0.0725	0.0373	0.0091	0.0435
	Mean	0.0207	0.0149	0.0084	0.0134	0.0143	0.0111	0.0040	0.0075
	SD	0.0181	0.0151	0.0250	0.0153	0.0268	0.0134	0.0286	0.0161
750	Best	0.0037	0.0009	0.0000	0.0002	0.0001	0.0001	0.0003	0.0001
	Median	0.0291	0.0167	0.0038	0.0047	0.0051	0.0023	0.0026	0.0033
	Worst	0.1101	0.0425	0.0182	0.0231	0.0244	0.0151	0.0149	0.0173
	Mean	0.0330	0.0221	0.0049	0.0079	0.0063	0.0048	0.0036	0.0055
	SD	0.0295	0.0145	0.0279	0.0139	0.0276	0.0136	0.0291	0.0136
1000	Best	0.0010	0.0005	0.0004	0.0010	0.0001	0.0013	0.0003	0.0001
	Median	0.0059	0.0074	0.0032	0.0056	0.0023	0.0040	0.0033	0.0028
	Worst	0.0258	0.0517	0.0284	0.0144	0.0109	0.0098	0.0156	0.0072
	Mean	0.0091	0.0103	0.0065	0.0069	0.0029	0.0043	0.0051	0.0038
	SD	0.0249	0.0156	0.0271	0.0121	0.0300	0.0137	0.0290	0.0142

Table 5.2 describes result comparison of Standard Bat Algorithm and FBI-BA. Performance evaluation is carried out by considering mean, median, worst, best and standard deviation (SD) values for diverse bat population and varying count of iterations. Selection of initial values and constants α and β is done considering research work carried out by various researchers and 0.9 value assigned to these constants yield more promising results. The result evaluation is done according to number of bats deployed for obtaining optimal results. From Table 5.2, it has been concluded that best optimal solution is obtained over 1000 iterations for bat population is equal to 25. Lesser the value of standard deviation, lesser is the variation among the data samples. Over 250 iterations and for 25 bats, there is minimum value of standard deviation. The standard deviation keeps increasing for 1000 iterations. The dissimilarities among feasible solutions keep increasing with increase in number of iterations, as bats keep exploring different directions for possible solutions. This analysis suggested that for 250 iterations, 25 bats are not sufficient to produce optimal results. So, there is a need to evaluate for more number of iterations. The best solution

obtained over 250 iterations is using 75 bats. In case of RD-BA and CATD-BA, best solution obtained over 250 iterations is using 100 bats. The solutions obtained after applying Standard Bat Algorithm and Flight behavior inspired Algorithm over different iterations for 25 bats are depicted in Figure 5.4.

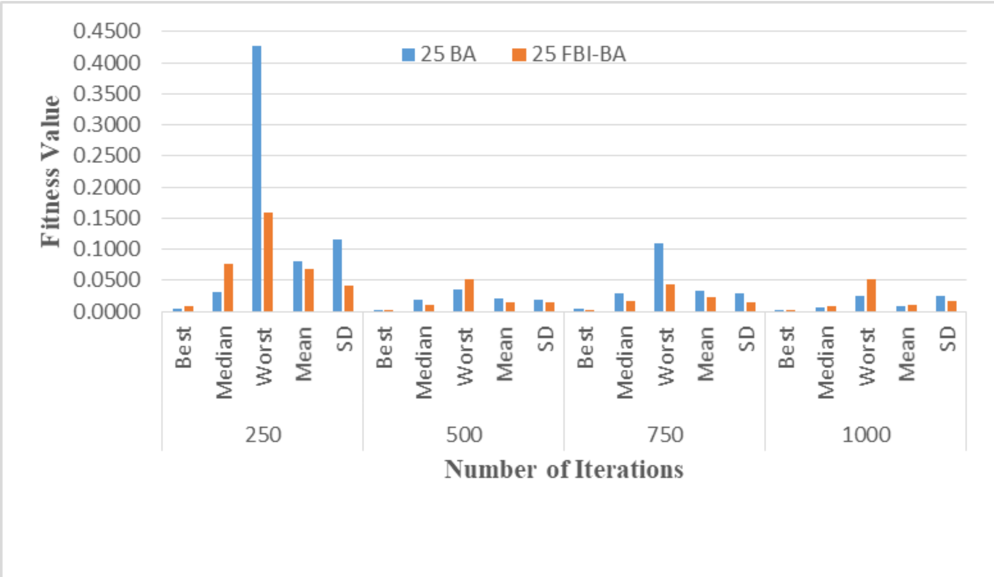


Figure 5.4: Result Evaluation of Flight behavior inspired Algorithm w.r.t. Standard Bat Algorithm for bat population=25

For bat population 50, best solution is obtained for 750 iterations. If number of iterations goes beyond 750, Standard Bat Algorithm is unable to produce more optimal results, which satisfy the minimization criteria of fitness functions.

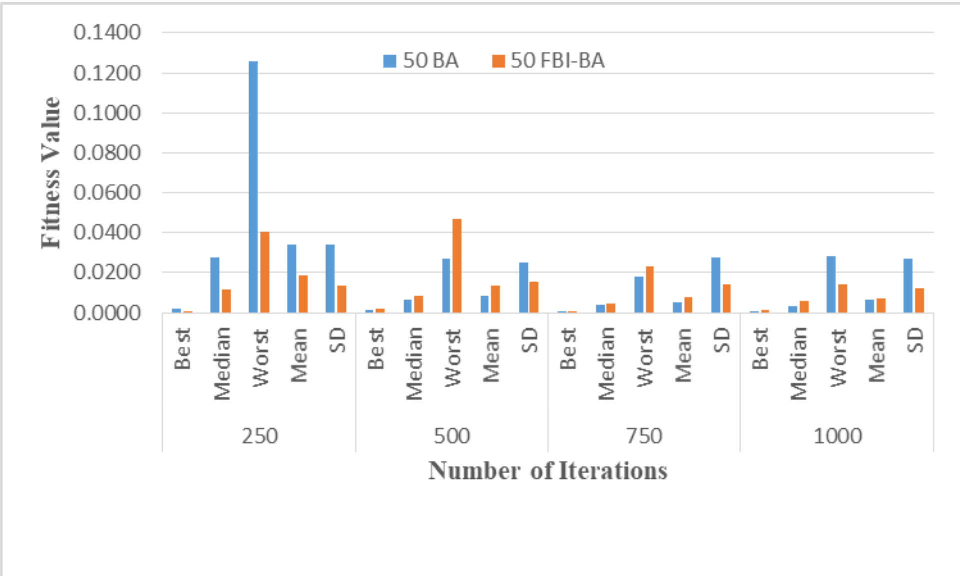


Figure 5.5: Result Evaluation of Flight behavior inspired Algorithm w.r.t. Standard Bat Algorithm for bat population=50

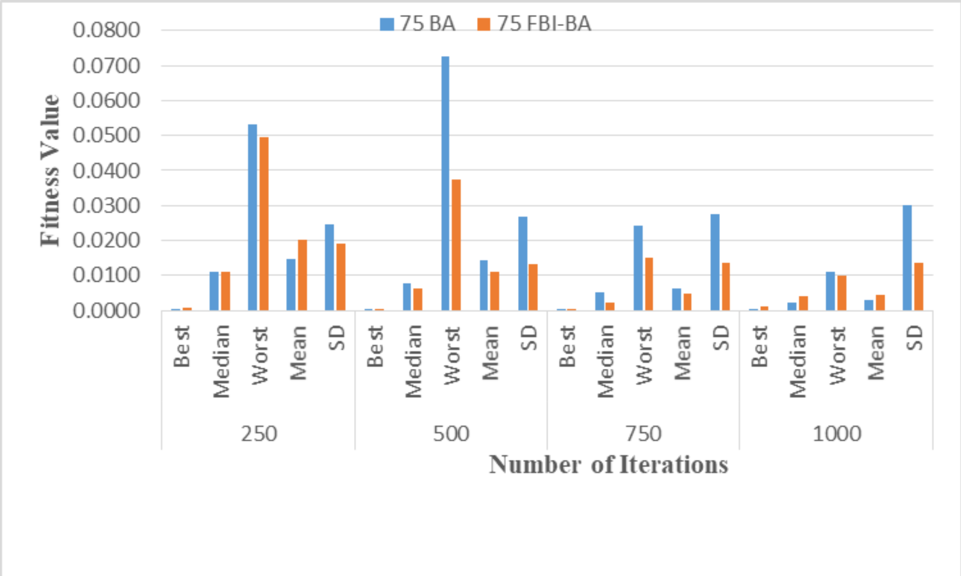


Figure 5.6: Result Evaluation of Flight behavior inspired Algorithm w.r.t. Standard Bat Algorithm for bat population=75

There is significant improvement of 20%, which is reflected from results shown in Table 5.2. The corresponding standard deviation of FBI-BA over 750 iterations is 0.01339. Standard deviation over 1000 iterations is 0.0121. But when compared with standard deviation over 500 iterations, there is a huge difference with respect to standard deviation over 1000 iterations. Considering this difference, usage of 250 iterations for obtaining results seems fruitful for obtaining results. The results are depicted in Figure 5.5. Considering worst values, 1000 iterations are required to obtain optimal results for bat population equal to 50. For selection of optimal solutions, best values are considered in this research work. For 75 bats, 750 iterations serve the purpose of obtaining optimal solution. The graphical format of results for 75 bats is shown in Figure 5.6.

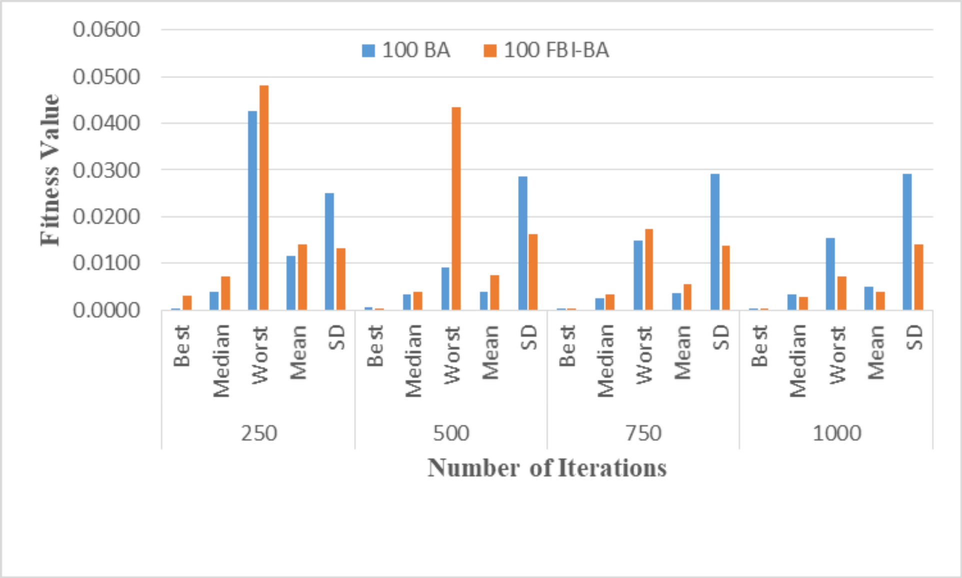
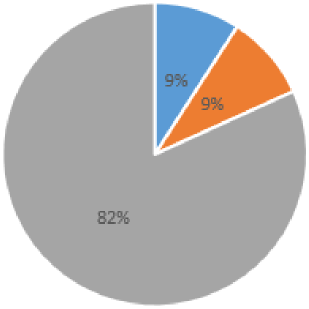


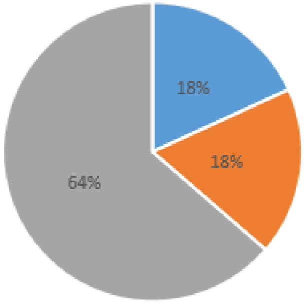
Figure 5.7: Result Evaluation of Flight behavior inspired Algorithm w.r.t. Standard Bat Algorithm for bat population=100

In consideration to worst values, there is very minute difference in worst case values obtained for 750 and 1000 iterations, for bat population equal to 100. The variation among feasible solutions also varies for different number of iterations. For 750 iterations, standard deviation is 0.0136 and for 1000 iterations, standard deviation is 0.0142. Moreover, standard deviation for different iterations is less varied for different bat population. The results are depicted in Figure 5.7.

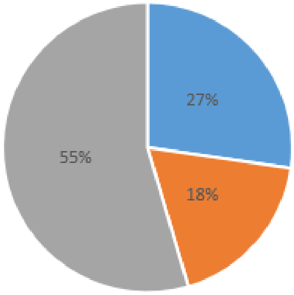
To evaluate performance of strategies adopted while targeting optimal solution, bat population and number of iterations are varied over [25,50,75,100] and [250,500,750,1000] respectively. It has been observed that for given number of iterations, 'follower' bat tends to follow 'leader' bat more, in presence of 25 bats. As bat population starts increasing over [25,50,75], bats tend to follow less leader bat. But, for bat population equal to 100, 73% of bats follow leader bats for obtaining optimal solution, as depicted in Figure 5.8 (a) to (d).



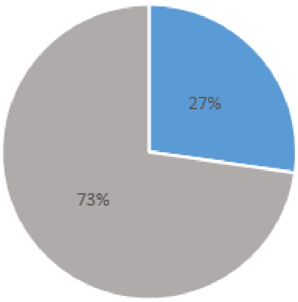
(a) N=25, Iter= 250



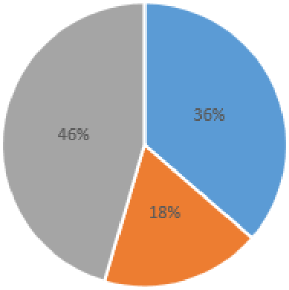
(b) N=50, Iter= 250



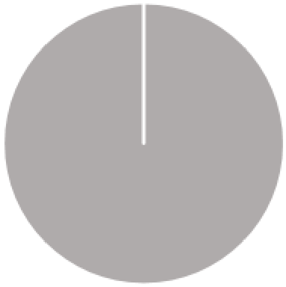
(c) N=75, Iter= 250



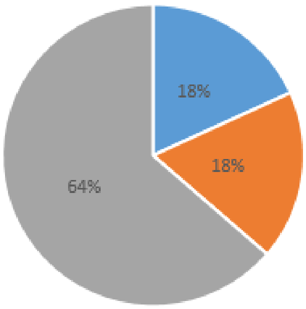
(d) N=100, Iter= 250



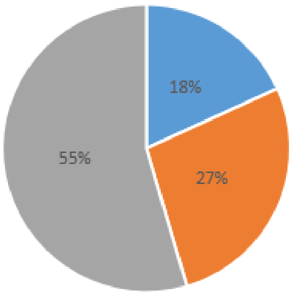
(e) N=25, Iter= 500



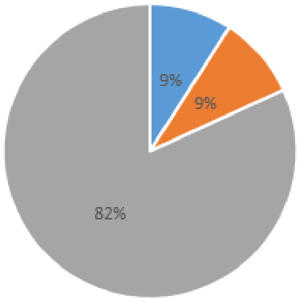
(f) N=50, Iter= 500



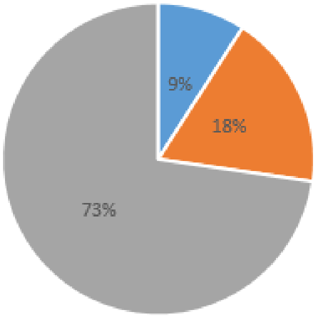
(g) N=75, Iter= 500



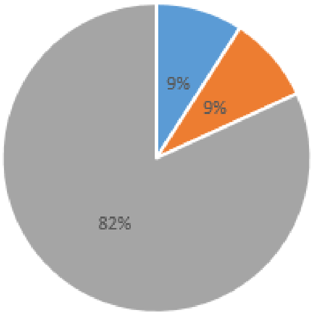
(h) N=100, Iter= 500



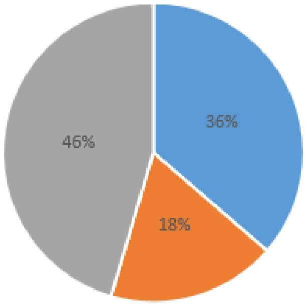
(i) N=25, Iter= 750



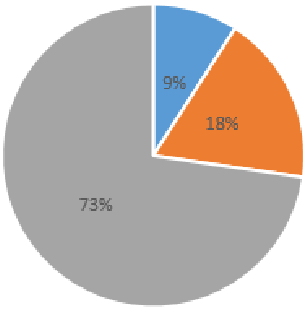
(j) N=50, Iter= 750



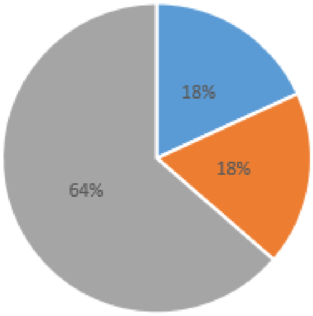
(k) N=75, Iter= 750



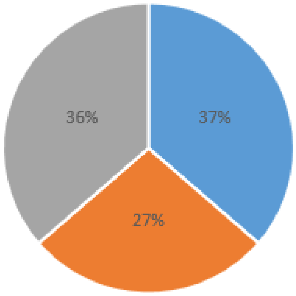
(l) N=100, Iter= 750



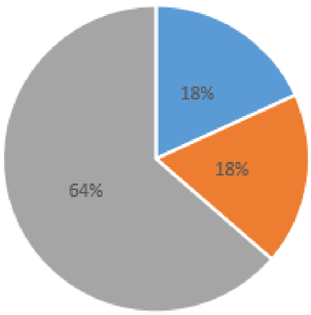
(m) N=25, Iter= 1000



(n) N=50, Iter= 1000



(o) N=75, Iter= 1000



(p) N=100,Iter= 1000

Figure 5.8: Performance Evaluation based on Pursuit Strategies adopted

Apart from 'following' strategy adopted by bats, bat tend to adopt 'converging' and 'diverging' strategies. While executing proposed algorithm over 250 iterations for 25% of bats, only 9% of bat population has adopted converging and diverging strategy, while rest of the bats follow leader bat. As bat population increases to 50, 18% of bats have adopted converging and diverging strategy. This proportion of converging and diverging strategy increases to 18% and 27% for bat population 75. For bat population equal to 25 for 250 iterations only 46% follower bats follow leader bat. Moreover, for bat population equal to 50 over 500 iterations, all bats present in search space prefer to follow leader bat. This proportionate number starts decreasing with increase in number of bats, i.e. for only 64% and 55% bats follow leader in case of bat population 75 and 100 for 500 iterations. In the beginning for N=25 for 500 iterations, 36% bats are able to converge successfully and 18% bats are able to diverge from each other, as depicted in Figure 5.8 (e) to (h). Whereas for 750 iterations, it has been noticed that bat population 25 and 75, yields same results. On the other hand, there is drastic downfall in follower category of bats for N=50 and 100 over 750 iterations. This situation is quite visible in Figure 5.8 (i) to (l). The bats tend to diverge more when 100 bats are present in search space and increases its proportion to 36%. For evaluation over 1000 iterations, bat population 50 and 100 yields same results. Whereas for bat population 25 and 75, there is huge difference in bat percentage of follower's list, i.e. 73% and 36%, as shown in Figure 5.8 (m) to (p). Bats tend to converge and diverge more in comparison to following strategy, when run over for 1000 iterations for 75 bats. This ratio is maximum among varying number of bats for fixed number of iterations, i.e. 1000.

Table 5.3: Comparison of Mean solution values of Leader and Follower bats

Bat Population	Leader			Follower		
	Best 1	Best 2	Best 3	Best 1	Best 2	Best 3
25	0.999959	0.999958	0.574343	0.999739	0.999504	0.421832

50	1.000076	1.00016	0.327596	0.999809	0.99963	0.11798
75	0.999927	0.999858	-0.5313	0.999975	1.016577	0.200867
100	0.99991	1.012333	1.161486	0.99998	0.999981	-0.06856

As mentioned in related research work, followers are able to capture prey/obtain optimal solution more quickly than leader. Based on this assumption, results obtained for following strategy are analyzed for bat population lying in range of [25,50,75,100]. Generally, for performance analysis, three best solutions and corresponding fitness value is recorded and it has been observed from mean value of results obtained that ‘follower’ bats tend to take longer steps in comparison to the ‘leader’ bat. This results into capturing of prey/obtaining optimal value, is mostly (60% times) done by follower bats. Table 5.3 summarizes mean value of obtained for best three solutions for both leader and follower bat, for varying bat population over [25,50,75,100].

5.4 Applicability of Flight behavior inspired algorithm (FBI-BA) for solving Traveling Salesman Problem

This section proposes and presents outcome of applicability of FBI-BA for solving Traveling Salesman Problem. This NP-hard problem has gained attention of many researchers. It has been noticed that solving such problems using these optimization techniques such as PSO, ACO and BA, has improved results. However Bat Algorithm is as powerful as Particle Swarm optimization and Genetic algorithms and considered to be special cases of Bat Algorithm. In this work main focus is to visit every city only once using newly developed Bat Algorithm version. Depending upon nature of algorithm and desired solution, suitability of Genetic algorithm, Particle Swarm Optimization, Bat Algorithm are introduced. The emphasis is to ensure that cost incurred in traveling all these cities should be minimal and obtaining optimal route.

5.4.1 Result Evaluation

In Table 5.4, fitness value obtained after applying Bat Algorithm and Flight behavior inspired algorithm is recorded. Here, bat population is varied over [20,40,60] and number of cities are varied over [25,50,75]. The result evaluation is carried out on the basis of four parameters: best, median, mean and worst. Comparison shows that FBI-BA generates more promising results than Standard Bat Algorithm.

Table 5.4: Result Evaluation of FBI-BA w.r.t. Bat Algorithm

Parameters	Bat Population=20, Cities =25		Bat Population=40, Cities =50		Bat Population=60, Cities =75	
	BA	FBI-BA	BA	FBI-BA	BA	FBI-BA
Best	0.01477	0.01420	0.01230	0.01160	0.00512	0.00489
Median	0.11775	0.11514	0.15348	0.13536	0.11900	0.11767
Worst	1.41162	1.35540	1.91122	1.70870	0.27320	0.23843
Mean	0.29142	0.28790	0.34356	0.32317	0.10938	0.10429

Figure 5.9 depicts values of best, mean, median and worst parameters for bat population equal to 20 and number of cities equal to 25. It has been observed that very minute difference is obtained. There is improvement of 4% in results obtained for best solution.

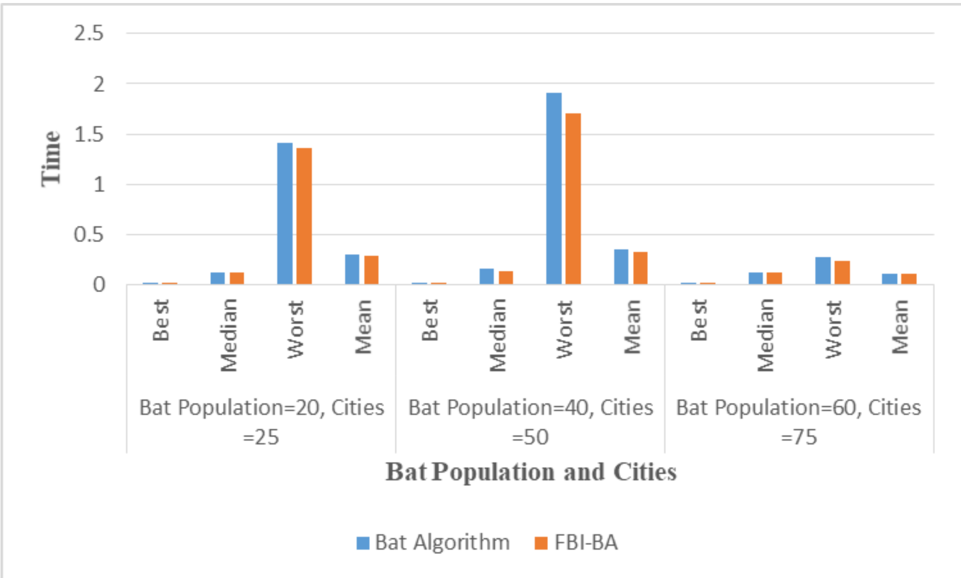


Figure 5.9: Graphical Representation of Comparison of Standard vs. FBI-BA, for varying Bat Population and Cities

There is improvement of 6% in results obtained for best solution, due to increase in bat population for bat population equal to 40 and cities equal to 50. The worst value obtained using FBI-BA is better than value obtained using BA. Moreover, 5% improvement in results is there for bat population equal to 60 and number of cities equal to 75 while obtaining optimal solution.

5.6 Summary

A study of three different pursuit strategies is carried out in this research work, in order to enhance performance of Standard Bat Algorithm. It has been noticed that 65% times, bats adopt following pursuit strategy while flying towards the prey (target), in the presence of two or more bats. Whereas, 16% of the times, bats opted for converging pursuit strategy and leftover 9% times, bats' have opted for diverging pursuit strategy. The proportion of 'following' bats is much greater than the other two types of pursuit strategies.

The next chapter presents evaluation of proposed variants of Bat Algorithm over different mathematical benchmark functions, followed by conclusion and future scope of this research work.

CHAPTER 6

PERFORMANCE EVALUATION USING MATHEMATICAL BENCHMARK FUNCTIONS

6.1 Numerical Experiments

In this chapter, performance of proposed algorithms of optimization are evaluated with respect to various mathematical benchmark optimization functions. In subsequent sections, characteristics of benchmark test functions are described and parameter settings are described in previous chapters. The function names are described in Table 6.2.

Table 6.1: Mathematical Benchmark Functions

Function	Definition
F1	$f(x) = \sum_{i=1}^{D-1} \left[100\left(x_{i+1} - x_i^2\right)^2 + (x_i - 1)^2\right]$

F2	$f(x) = \sum_{i=1}^{D} x_i^2$
F3	$f(x) = \sum_{i=1}^{D} \left(x_i^2 - 10\cos(2\pi x_i) + 10\right)$
F4	$f(x) = \frac{1}{4000}\sum_{i=1}^{D} x_i^2 - \prod_{i=1}^{D} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$
F5	$f(x) = 0.5 + \frac{sin(\sqrt{x_1^2+x_2^2})}{(1+0.001(x_1^2+x_2^2))^2}$
F6	$f(x) = x_1^2 + 2x_2^2 - 0.3cos(3\pi x_1) - 0.4cos(4\pi x_2) + 0.7$
F7	$f(x) = \sum_{i=1}^{n} x_i^2 + (\sum_{i=1}^{n} 0.5ix_i)^2 + (\sum_{i=1}^{n} 0.5ix_i)^4$
F8	$f(x) = \left\{1 + (x_0 + x_1 + 1)^2(19 - 14x_0 + 3x_0^2 - 14x_1 + 3x_1^2)\right\}$ $\left\{30 + (2x_0 - 3x_1)^2(18 - 32x_0 + 12x_0^2 + 48x_1 - 36x_0x_1 + 27x_1^2\right\}$
F9	$f(x) = 20 + exp(1) - 20exp\left(-\frac{1}{5}\sqrt{\frac{1}{D}\sum_{i=1}^{D} x_i^2}\right) - exp\left(\frac{1}{D}\sum_{i=1}^{D}\cos(2\pi x_i)\right)$
F10	$f(x) = (x_1 - \frac{5.1}{4\pi^2}x_0^2 + \frac{5}{\pi}x_0 - 6)^2 + 10(1 - \frac{1}{8\pi})cos(x_0) + 10$

F11	$f(x) = -cos(x_1)cos(x_2)exp(-((x_1-\pi)^2 + (x_2-\pi)^2))$
F12	$f(x) = -\sum_{i=1}^{4} \alpha_i exp(-\sum_{j=1}^{3} A_{ij}(x_j - P_{ij})^2)$ $\alpha = [1 \quad 1.2 \quad 3 \quad 3.2] \; A = \begin{bmatrix} 3 & 10 & 30 \\ 0.1 & 10 & 35 \\ 3 & 10 & 30 \\ 0.1 & 10 & 35 \end{bmatrix}$ $P = \begin{bmatrix} 0.3689 & 0.117 & 0.2673 \\ 0.4699 & 0.4387 & 0.747 \\ 0.1091 & 0.8732 & 0.5547 \\ 0.03815 & 0.5743 & 0.8828 \end{bmatrix}$
F13	$f(x) = \sum_{j=1}^{5} jcos(j+1)x_1 + j) \sum_{j=1}^{5} jcos((j+1)x_2 + j)$

The proposed algorithms of this research work are tested on 13 mathematical benchmark functions, with various properties (unimodal and multimodal). These functions are recorded in Table 6.1 and their characteristics are listed in Table 6.2.

Table 6.2: Characteristics of Mathematical Benchmark Functions

Function	Name	Bounds	Optimal Value
F1	Rosenbrock	$[-30,30]^D$	0
F2	Sphere	$[-100,100]^D$	0
F3	Rastrigin	$[-5.12,5.12]^D$	0

F4	Griwank	$[-600,600]^D$	0
F5	Schaffer	$[-100,100]^D$	0
F6	B2	$[-100,100]^D$	0
F7	Zakahrov	$[-10,10]^D$	0
F8	Goldstein and Price	$[-2,2]^D$	3
F9	Ackley	$[-32,32]^D$	0
F10	Branin	$[-10,10]^D$	0.397887
F11	Easom	$[-100,100]^D$	-1
F12	Hartmann	$[0,1]^D$	-3.86278
F13	Shubert	$[-10,10]^D$	-186.7309

In order to investigate performance of proposed algorithms of this research work, 13 benchmark functions are applied to all proposed variants of Bat Algorithm. The parameters considered for evaluation of these algorithms include best, mean, median, standard deviation and worst. The run is successfully executed for 500 iterations and for the bat population equal to 50. The results are depicted in Table 6.3.

Table 6.3: Comparison Results of Proposed Algorithms for F1 to F13 Benchmark Functions

Function	Parameters	RD-Bat	CATD-Bat	FBI-BA
Rosenbrock	Best	1.45E-07	3.07E-04	7.60E-02

	Median	6.07E-07	3.19E-04	9.54E-02
	Worst	3.56E-06	3.56E-06	4.20E-08
	Mean	9.75E-07	3.85E-04	1.39E-01
	SD	6.44E-07	4.81E-04	2.26E-01
Sphere	Best	9.08E-08	1.69E-04	6.96E-02
	Median	2.33E-07	2.53E-04	8.47E-02
	Worst	6.62E-07	6.62E-07	6.62E-07
	Mean	2.74E-07	2.94E-04	1.02E-01
	SD	7.57E-07	3.82E-04	1.18E-01
Rastrigin	Best	3.98E-10	7.10E-04	5.25E-02
	Median	1.90E-07	1.37E-04	1.45E-01
	Worst	6.30E-07	6.30E-07	6.30E-07
	Mean	2.27E-07	4.46E-04	1.89E-01
	SD	8.22E-07	6.68E-04	2.59E-01
Griwank	Best	1.00E+00	1.19E-04	1.00E+00
	Median	1.00E+00	2.66E-04	1.00E+00
	Worst	4.20E-08	4.20E-08	4.20E-08
	Mean	1.00E+00	5.90E-04	1.00E+00
	SD	1.05E+00	9.70E-04	1.05E+00
Schaffer	Best	2.58E-11	2.80E-04	1.10E-04
	Median	1.22E-11	4.65E-04	1.79E-04
	Worst	4.20E-08	4.20E-08	4.20E-08
	Mean	1.41E-11	5.19E-04	2.98E-04
	SD	6.29E-07	6.72E-04	3.99E-04
B2	Best	2.69E-08	5.23E-04	1.24E-01
	Median	9.90E-08	1.88E-04	2.76E-01
	Worst	4.13E-01	4.13E-01	4.13E-01
	Mean	4.13E-02	4.34E-04	2.85E-01
	SD	1.38E-01	6.78E-04	3.43E-01
Zakharov	Best	9.07E-10	4.10E-05	1.79E-02
	Median	2.64E-09	4.17E-04	2.42E-02

	Worst	4.20E-08	4.20E-08	4.20E-08
	Mean	9.25E-09	4.18E-04	3.67E-02
	SD	1.02E-06	5.44E-04	5.64E-02
Goldstein and Price	Best	3.00E+00	2.20E-04	7.96E-01
	Median	3.00E+00	1.83E-04	7.41E-01
	Worst	4.20E-08	4.20E-08	4.20E-11
	Mean	3.00E+00	2.81E-04	7.59E-01
	SD	3.16E+00	4.14E-04	8.04E-01
Ackley	Best	6.98E-04	7.06E-04	1.40E+00
	Median	9.55E-04	1.97E-04	9.26E-01
	Worst	4.20E-08	4.20E-08	3.56E-06
	Mean	2.13E-01	2.88E-04	8.69E-01
	SD	7.07E-01	4.03E-04	1.14E+00
Branin	Best	8.46E-01	6.31E-05	8.77E-01
	Median	8.46E-01	2.47E-04	1.04E-01
	Worst	4.20E-08	4.20E-08	4.20E-08
	Mean	8.46E-01	5.67E-04	3.29E-01
	SD	8.92E-01	9.50E-04	5.28E-01
Easom	Best	-1.00E+00	2.21E-04	-2.33E-05
	Median	-1.00E+00	3.63E-04	-2.37E-05
	Worst	4.20E-08	4.20E-08	4.20E-08
	Mean	-1.00E+00	4.05E-04	-3.21E-02
	SD	1.05E+00	5.40E-04	1.06E-01
Hartmann	Best	-3.35E+00	1.45E-04	-3.14E+00
	Median	-3.35E+00	2.74E-04	-3.15E+00
	Worst	4.20E-08	4.20E-08	4.20E-08
	Mean	-3.35E+00	4.25E-04	-3.13E+00
	SD	3.54E+00	6.04E-04	3.30E+00
	Best	-1.87E+02	9.71E-04	9.64E-02
Shubert	Median	-1.87E+02	3.11E-04	9.70E-02
	Worst	4.20E-08	4.20E-08	4.20E-08

	Mean	-1.87E+02	3.19E-04	1.22E-01
	SD	1.97E+02	4.39E-04	1.70E-01

On equating results obtained using benchmark functions, Schaffer has produced better results. The best value obtained is 2.58E-11, median is 1.22E-11, worst value is 4.20E-08, mean is 1.41E-11 and standard deviation is 6.29E-07. Here, while comparing results, benchmark functions which generates negative results are excluded. These functions are Easom, Hartmann and Shubert.

The results in Table 6.3 shows that Schaffer function obtains more optimal result using same number of iterations and bat population, in comparison to other benchmark functions.

6.2 Performance Evaluation of RD-BA w.r.t. Mathematical Benchmark Functions

Here, analysis of results obtained for RD-Bat Algorithm using different mathematical benchmark functions is done. Figure 6.1, depicts the values of different parameters, like best, median, mean, standard deviation and worst, for 13 different functions. If results of all benchmark functions are considered and compared, then based on parameters of interest, different functions will be preferred for different scenarios. In order to obtain best optimal solution, Shubert function can be used as benchmark function, as it offers minimum and optimal solution. In case of maximization function, B2 function can be preferred over other functions, as it offers maximum value for 50 bat population over 500 iterations.

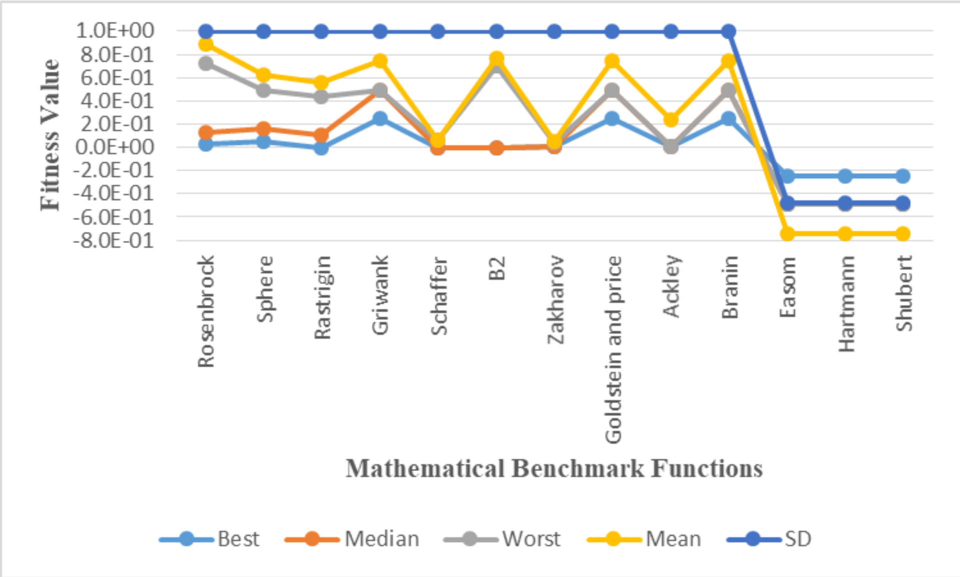


Figure 6.1: Comparison of RD-BAT over Mathematical Benchmark Functions

On the basis of Mean and Median, Shubert function can be used for benchmarking, as it offers minimum value over 500 iterations and for 50 bats. But, in order to reduce gaps between solutions obtained in search space, Sphere function offers best results with respect to standard deviation parameter. Even though, it does not offer best optimal solution, but can be used in those scenarios where motive is to optimize solutions present in search space.

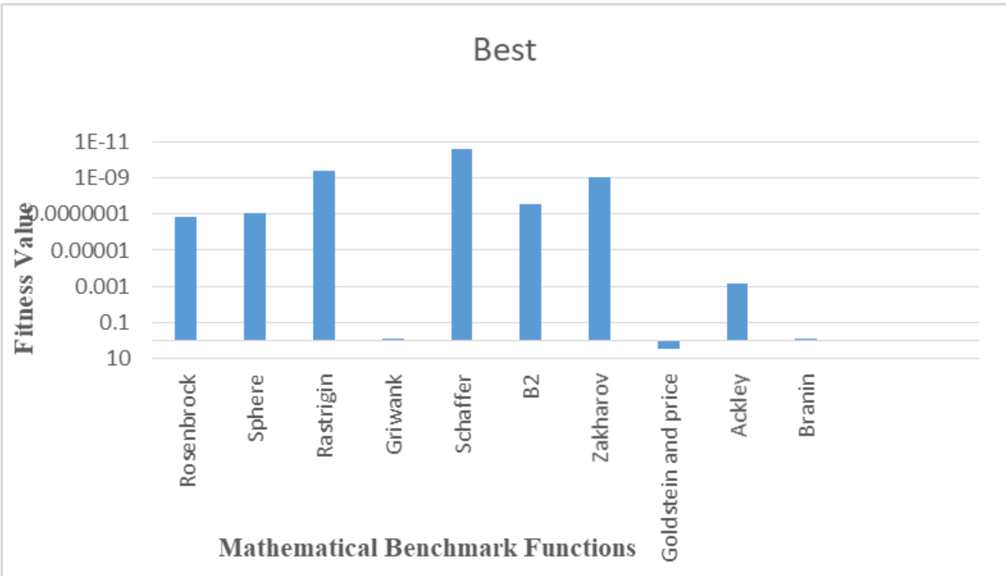


Figure 6.2: Comparison of 'Best' Solutions obtained for different benchmark functions

Figure 6.2 depicts best values obtained for RD-Bat Algorithm. RD-Bat Algorithm is capable of generating best result, as per underlying minimization function and this newly obtained solution is better than solutions offered by other mathematical optimization functions. The time taken to converge towards global optima is 49 seconds (approximately) for all benchmark functions. So, it is worth mentioning that proposed algorithm has proven to be efficient one while obtaining optimal solutions. Figure 6.2 compares best results obtained by all mathematical benchmark functions, which are used for optimization. When comparing best solutions on all benchmark functions with other variants of Bat Algorithm, value lies between 1.4504E-07 and 3 seems reasonable and fruitful. Lesser value of best parameter will motivate algorithm to settle down as early as possible, without exploring other solutions. But this may lead to stuck in trap of local optimal solutions. On the other hand, if too many iterations are used, algorithm may keep on discovering new solutions and waste time to acquire better solution than existing one.

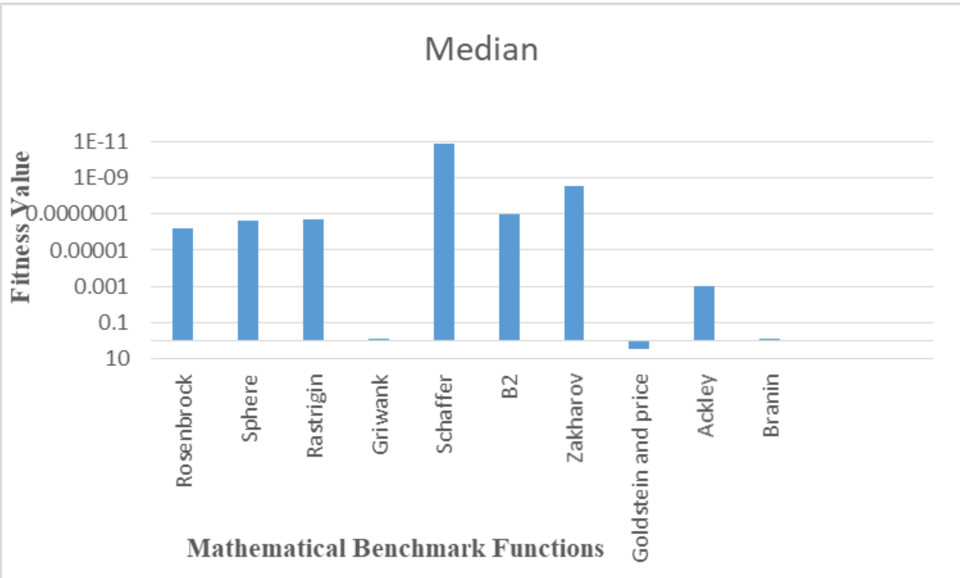


Figure 6.3: Comparison of 'Median' Solutions obtained for different benchmark functions

Figure 6.3 compares median results obtained by all mathematical benchmark functions, which are used for optimization. When comparing medians of all benchmark functions with other variants of Bat Algorithm, value lies between 6.0742E-07 and 3 seems reasonable and productive. The best value of median obtained using Rosenbrock in comparison to 13 other mathematical benchmark functions, i.e. 6.07E-07. After Rosenbrock, Schaffer has obtained second best median value, among other benchmark functions.

Figure 6.4 compares worst results obtained by all mathematical benchmark functions, which are used for optimization. When comparing worst solutions on all benchmark functions with other variants of Bat Algorithm, value lies between 3.5564E-06 and 0.41332 seems to reasonable and productive. The best value of median obtained using Rosenbrock in comparison to 13 other mathematical benchmark functions, i.e. 6.07E-07. After Rosenbrock, Schaffer has obtained second maximum worst value, among other benchmark functions.

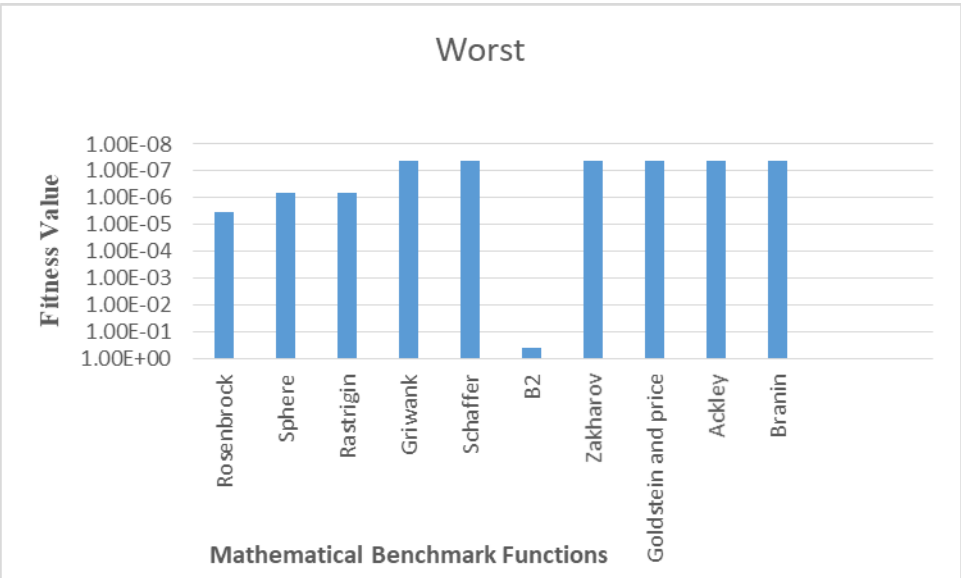


Figure 6.4: Comparison of 'Worst' Solutions obtained for different benchmark functions

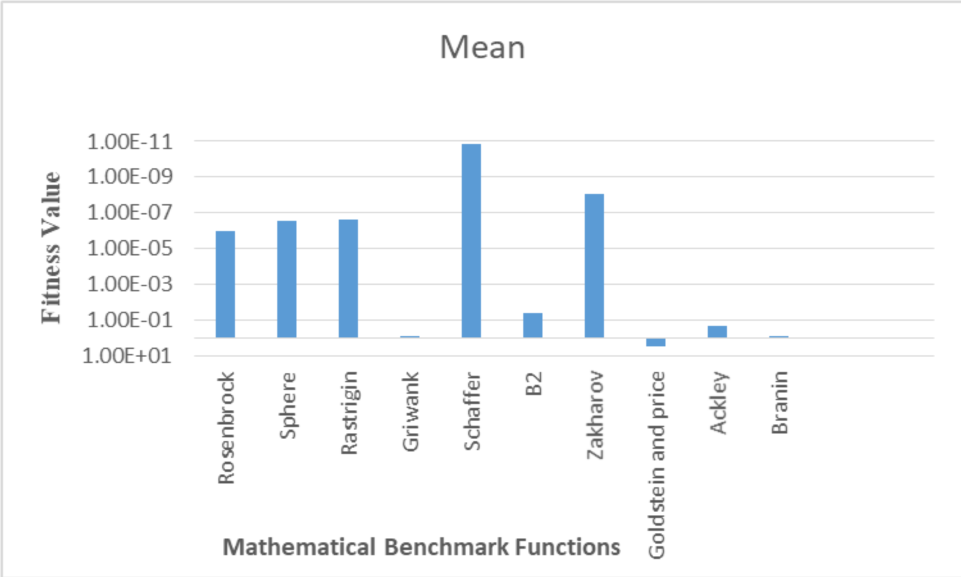


Figure 6.5: Comparison of 'Mean' Solutions obtained for different benchmark functions

Figure 6.5 compares mean results obtained by all mathematical benchmark functions, which are used for optimization. When comparing mean solutions on all benchmark functions with other variants of Bat Algorithm, value lies between 9.74748E-07 and 3 seems to reasonable and productive. The best value of median obtained using Rosenbrock in comparison to 13 other mathematical benchmark functions, i.e. 6.07E-07. After Rosenbrock, Schaffer has obtained second best mean value, among other benchmark functions.

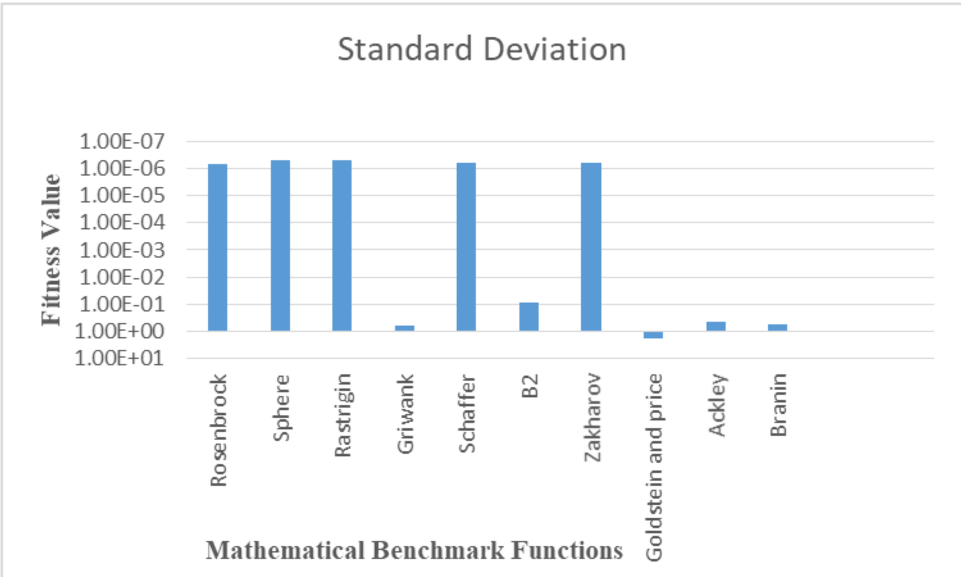


Figure 6.6: Comparison of 'Standard Deviation' Solutions obtained for different benchmark functions

Figure 6.6 compares standard deviation obtained by all mathematical benchmark functions, which are used for optimization. When comparing standard deviation solutions on all benchmark functions with other variants of Bat Algorithm, the value lies between 6.4356E-07 and 1.936491044 seems to reasonable and productive. The best value of median obtained using Rosenbrock in comparison to 13 other mathematical benchmark functions, i.e. 6.07E-07. After Rosenbrock and Sphere has obtained second best standard deviation, among other benchmark functions.

6.3 Performance Evaluation of CATD-BA w.r.t. Mathematical Benchmark Functions

Here, analysis of results obtained for CATD-Bat Algorithm using different mathematical benchmark functions is done. Figure 6.7, depicts values of different parameters, like best, median, mean, standard deviation and worst, for 13 different functions. Different mathematical functions will be preferred for different scenarios. In order to obtain best optimal solution, Shubert function can be used as benchmark function, as it offers minimum and optimal solution. In case of maximization function, B2 function can be preferred over other functions, as it offers maximum value for 50 bat population over 500 iterations. On the basis of Mean and Median, Shubert function can be used for benchmarking, as it offers minimum value over 500 iterations and for 50 bats. But, in order to reduce gaps between solutions obtained in search space, Sphere function offers best results with respect to standard deviation parameter. Even though, it does not offer best optimal solution, but can be used in those scenarios where motive is to optimize solutions present in search space.

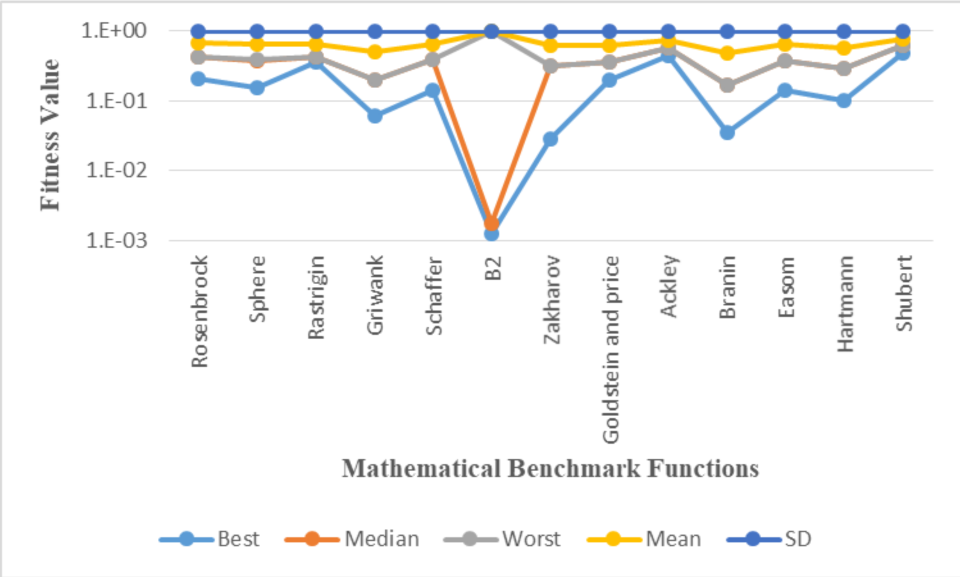


Figure 6.7: Comparison of CATD-BAT over Mathematical Benchmark Functions

Figure 6.8 compares best results obtained by all mathematical benchmark functions. When best solutions on all benchmark functions are compared with other variants of Bat Algorithm, value lies between 4.10E-05 and 9.71E-04 seems reasonable and fruitful. Lesser value of best parameter will motivate algorithm to settle down as early as possible, without exploring other solutions. But this may lead to stuck in trap of local optimal solutions. On the other hand, if too many iterations are used, algorithm may keep on exploring solutions and waste time to acquire better solution than existing one.

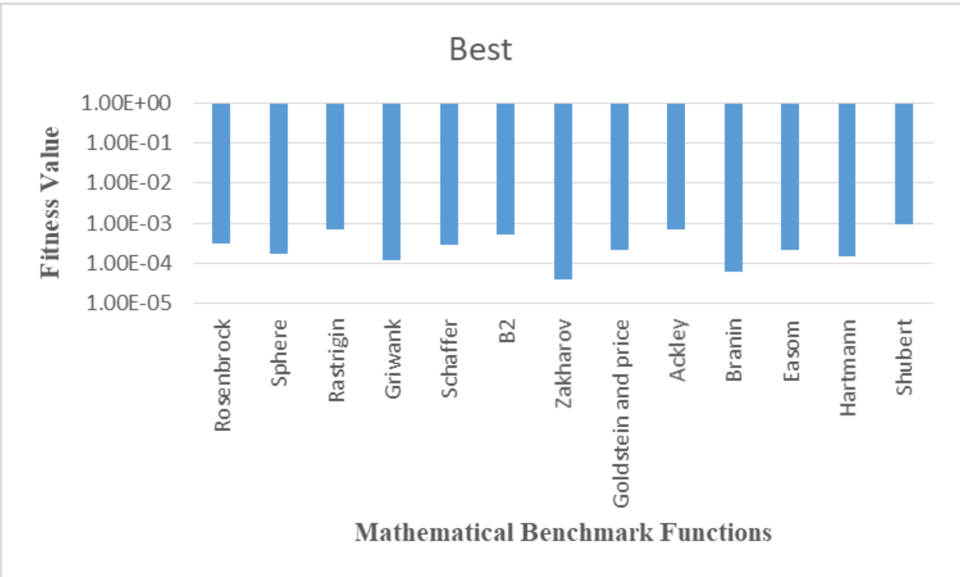


Figure 6.8: Comparison of 'Best' Solutions obtained for different benchmark functions

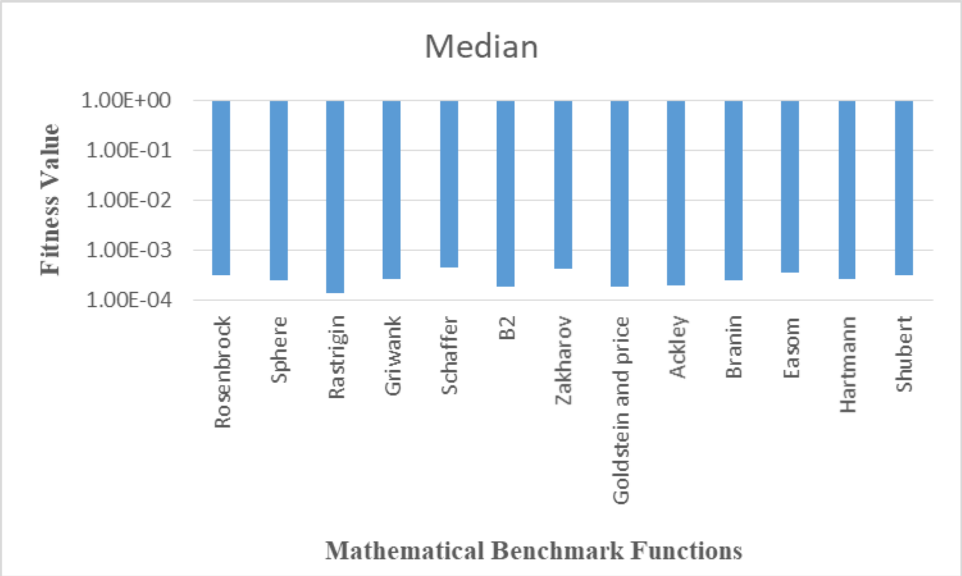


Figure 6.9: Comparison of 'Median' Solutions obtained for different benchmark functions

Figure 6.9 compares median results obtained by all mathematical benchmark functions, which are used for optimization. When comparing medians of all benchmark functions with other variants of Bat Algorithm, value lies between 1.37E-04 and 4.65E-04 seems reasonable and productive. The best value of median obtained using Rastrigin in comparison to 13 other mathematical benchmark functions, i.e. 1.37E-04. After Rastrigin, Golstein and Price has obtained second best median value, among other benchmark functions.

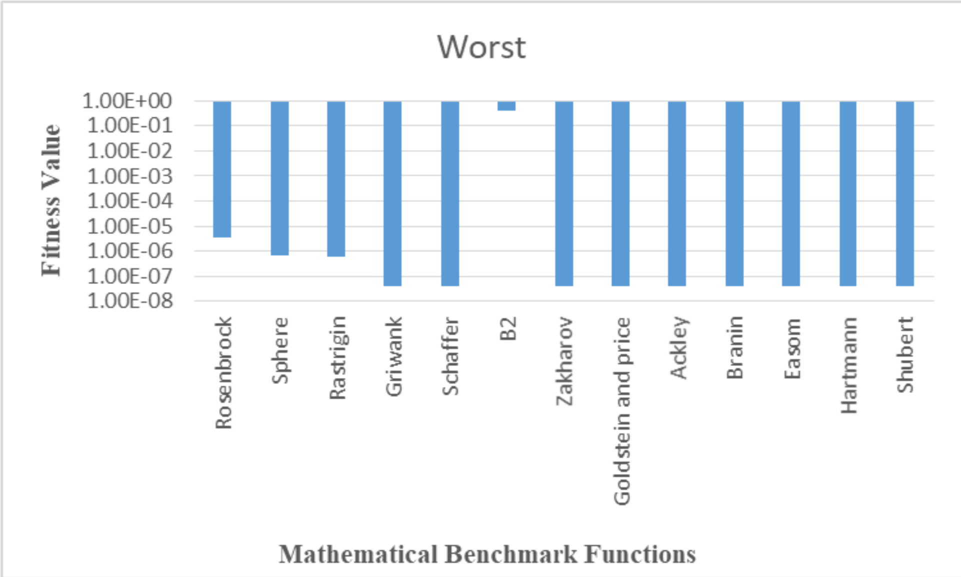


Figure 6.10: Comparison of 'Worst' Solutions obtained for different benchmark functions

Figure 6.10 compares worst results obtained by all mathematical benchmark functions, which are used for optimization. When comparing worst solutions on all benchmark functions with other variants of Bat Algorithm, value lies between 4.20E-08 and 4.13E-01 seems reasonable and productive. The worst value obtained using schaffer in comparison to 13 other mathematical benchmark functions, i.e. 4.20E-08. After schaffer, griwank has obtained second maximum worst value, among other benchmark functions.

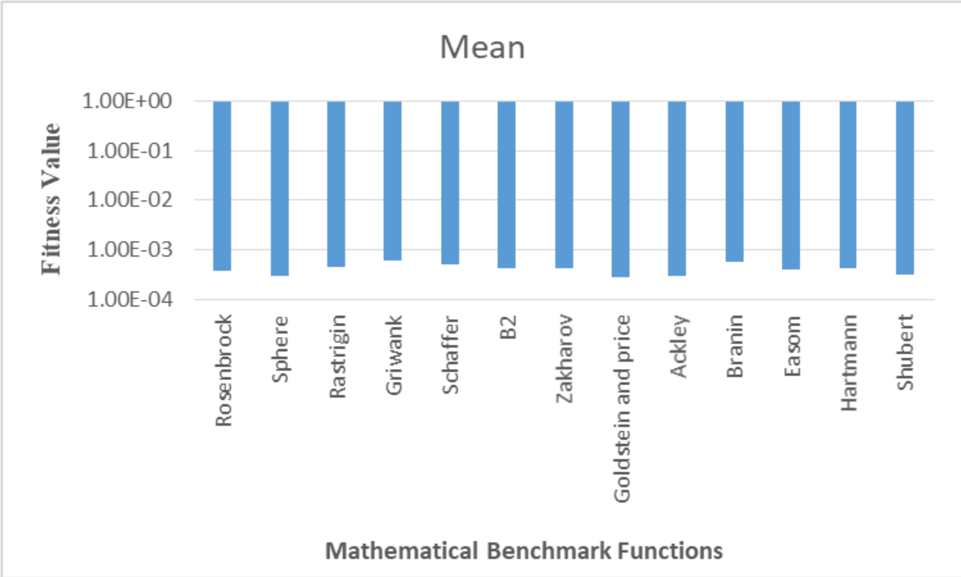


Figure 6.11: Comparison of 'Mean' Solutions obtained for different benchmark functions

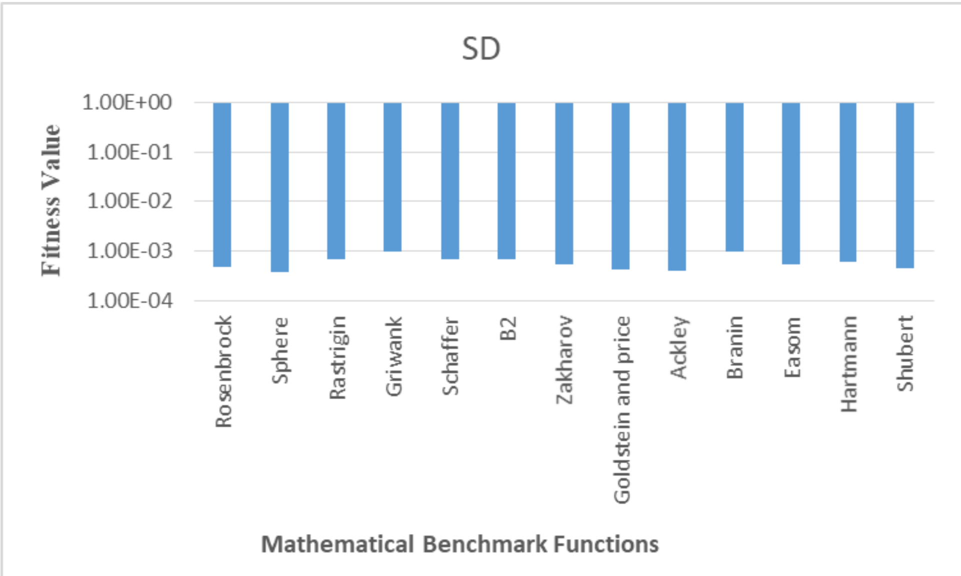


Figure 6.12: Comparison of 'Standard Deviation' Solutions obtained for different benchmark functions

Figure 6.11 compares mean results obtained by all mathematical benchmark functions, which are used for optimization. When comparing mean solutions on all benchmark functions with other variants of Bat Algorithm, value lies between 2.81E-04 and 5.90E-04 seems reasonable and productive. The best value of mean obtained using goldstein and price in comparison to 13 other mathematical benchmark functions, i.e. 2.81E-04. After Goldstein and price, ackley has obtained second best mean value, among other benchmark functions. Figure 6.12 compares standard deviation obtained by all mathematical benchmark functions. When comparing standard deviation solutions on all benchmark functions with other variants of Bat Algorithm, the value lies between 3.82E-04 and 9.70E-04 seems to reasonable and productive. The best value of standard deviation obtained using sphere in comparison to 13 other mathematical benchmark functions, i.e. 3.82E-04. After sphere, Ackley has obtained second best standard deviation, among other benchmark functions.

6.4 Performance Evaluation of FBI-BA w.r.t. Mathematical Benchmark Functions

Here, analysis of results obtained for FBI-BA using different mathematical benchmark functions is done. Figure 6.13, depicts values of different parameters, like best, median, mean, standard deviation and worst, for 13 different functions. If results of all benchmark functions are considered and compared, then based on parameters of interest, different functions will be preferred for different scenarios.

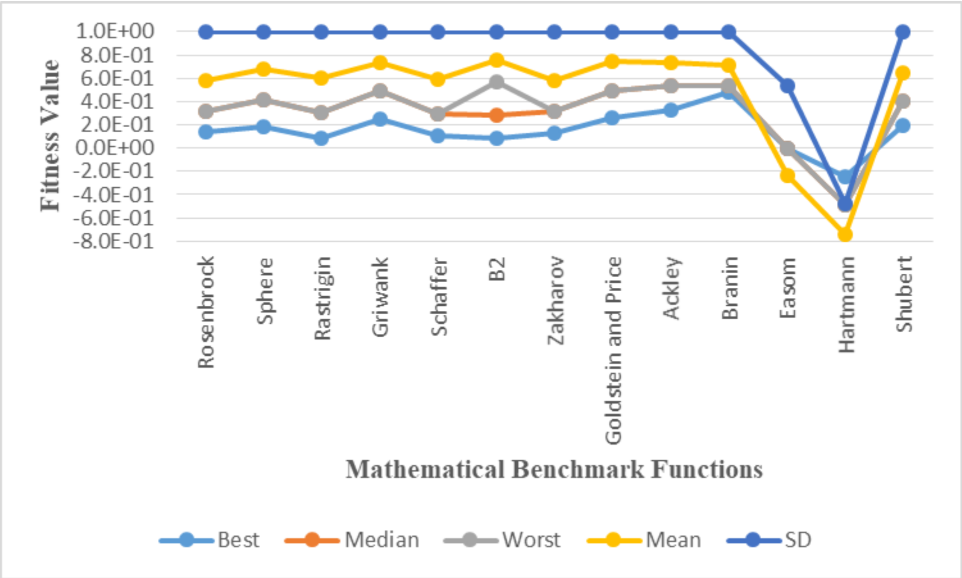


Figure 6.13: Comparison of FBI-BA over Mathematical Benchmark Functions

In order to obtain best optimal solution, Hartmann function can be used as benchmark function, as it offers minimum and optimal solution. In case of maximization function, B2 function can be preferred over other functions, as it offers maximum value for 50 bat population over 500 iterations. On the basis of Mean and Median, Shubert function can be used for benchmarking, as it offers minimum value over 500 iterations and for 50 bats. But, in order to reduce gaps between solutions obtained in search space, Schaffer function offers best results with respect to standard deviation parameter. Even though, it does not offer best optimal solution, but can be used in those scenarios where motive is to optimize the solutions present in search space.

Figure 6.14 depicts best values obtained for FBI-BA. FBI-BA is capable of generating best result, as per underlying minimization function and this newly obtained solution is better than solutions offered by other mathematical optimization functions. The time taken to converge towards global optima is 37 seconds (approximately) for all benchmark functions. So, it is worth mentioning that FBI-BA has proven to be efficient one while obtaining optimal solutions.

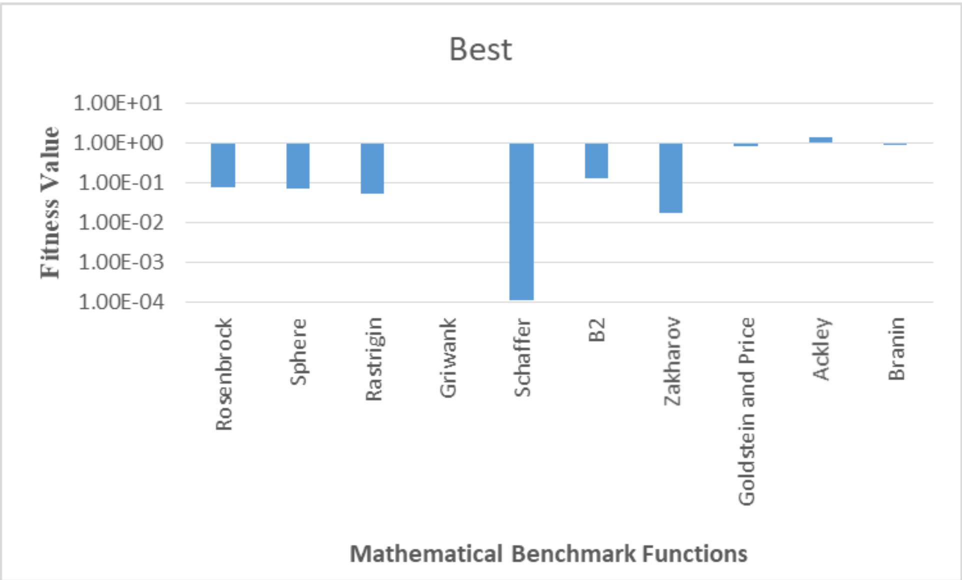


Figure 6.14: Comparison of 'Best' Solutions obtained for different benchmark functions

Figure 6.14 compares best results obtained by all mathematical benchmark functions. When comparing best solutions on all benchmark functions with other variants of Bat

Algorithm, value lies between -3.14 and 1.4 seems reasonable and fruitful. Lesser value of best parameter will motivate algorithm to settle down as early as possible, without exploring other solutions. But this may lead to stuck in trap of local optimal solutions. On the other hand, if too many iterations are used, algorithm may keep on discovering new solutions and waste time to acquire better solution than existing one.

Figure 6.15 compares median results obtained by all mathematical benchmark functions, which are used for optimization. When comparing median of all benchmark functions with other variants of Bat Algorithm, value lies between -3.15 and 1 seems reasonable and productive. The best value of median obtained using Hartmann in comparison to 13 other mathematical benchmark functions, i.e. -3.15. After Hartmann, Easom has obtained second best median value, among other benchmark functions.

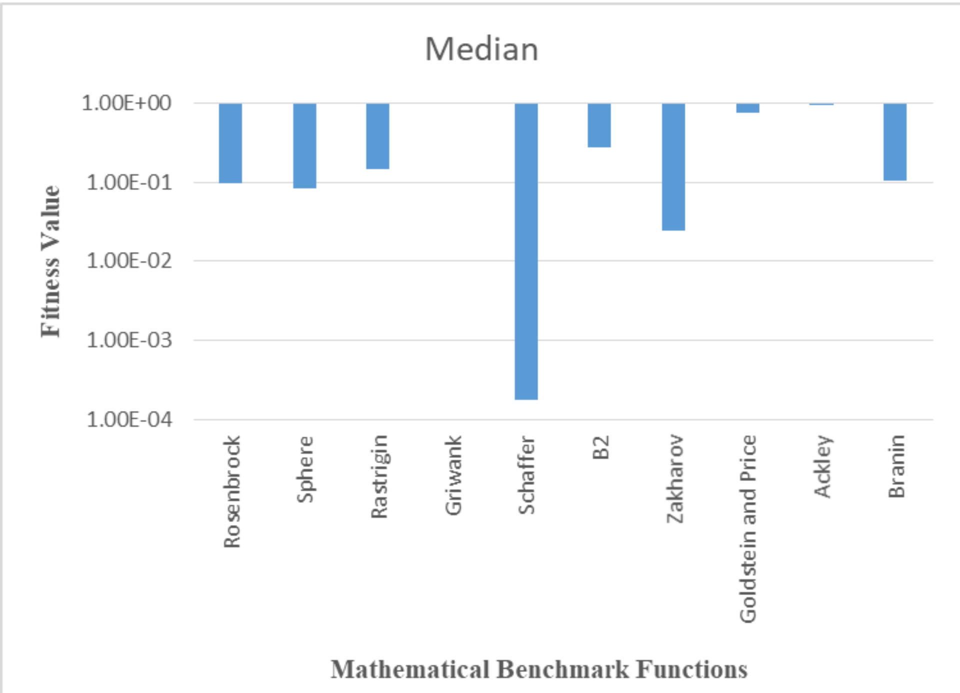


Figure 6.15: Comparison of 'Median' Solutions obtained for different benchmark functions

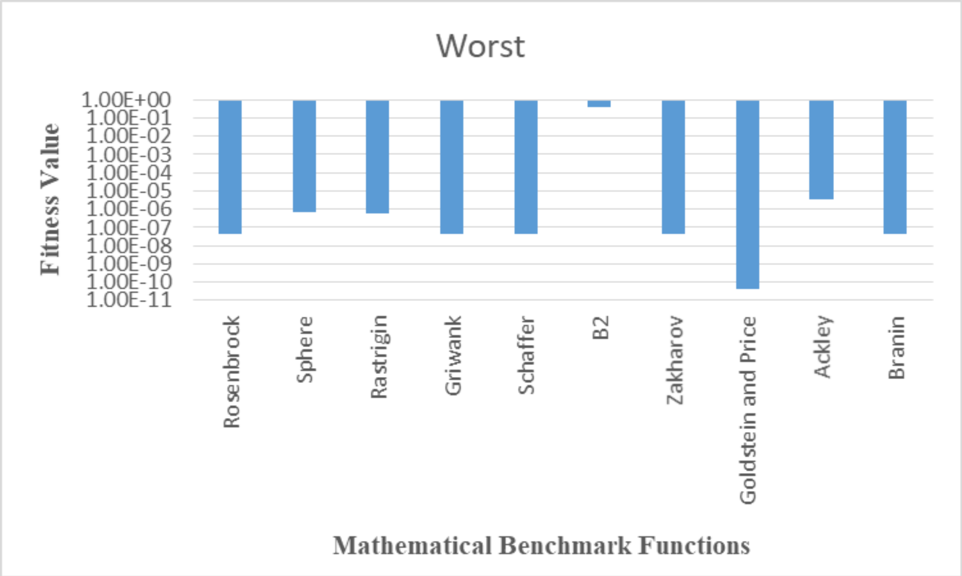


Figure 6.16: Comparison of 'Worst' Solutions obtained for different benchmark functions

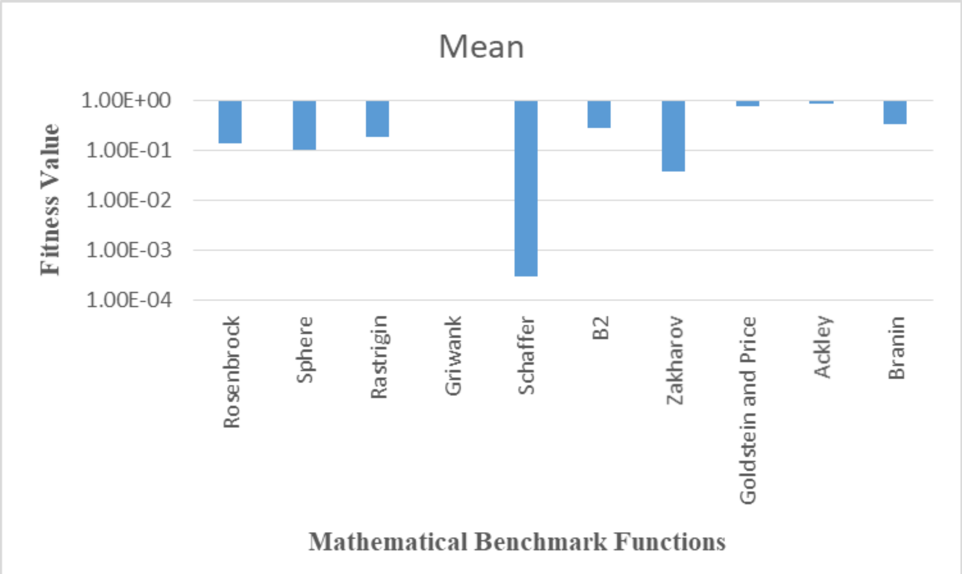


Figure 6.17: Comparison of 'Mean' Solutions obtained for different benchmark functions

Figure 6.16 compares worst results obtained by all mathematical benchmark functions, which are used for optimization. When comparing worst solutions on all benchmark functions with other variants of Bat Algorithm, value lies between 4.20E-11 and 4.13E-01 seems reasonable and productive. The worst value obtained using

Goldstein and Price in comparison to 13 other mathematical benchmark functions, i.e. 4.20E-11. After Goldstein and Price, Rosenbrock has obtained second maximum worst value, among other benchmark functions. Figure 6.17 compares mean results obtained by all mathematical benchmark functions. When comparing mean solutions on all benchmark functions with other variants of Bat Algorithm, value lies between -3.13E+00 and 1.00E+00 seems reasonable and productive. The best value of median obtained using Hartmann in comparison to 13 other mathematical benchmark functions, i.e. -3.13E+00.

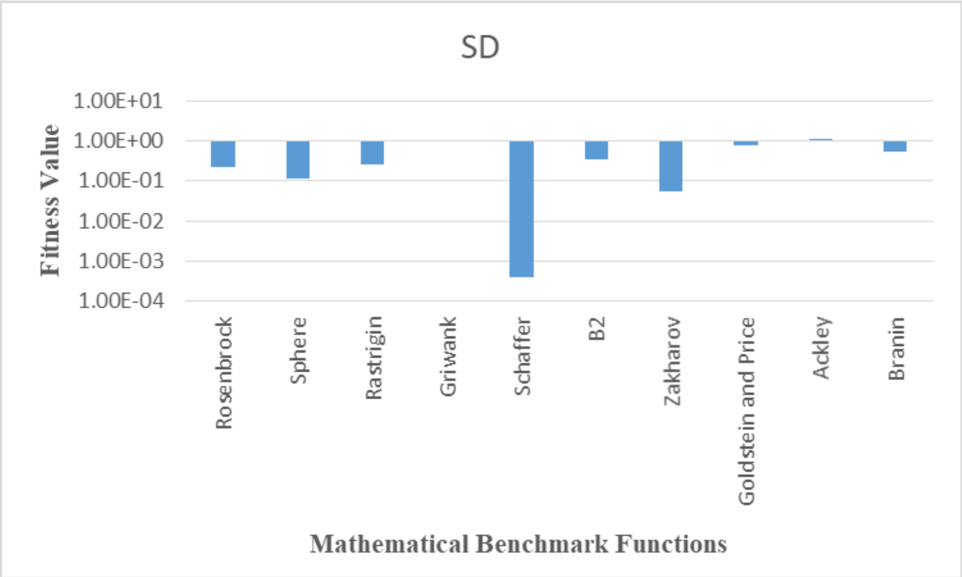


Figure 6.18: Comparison of 'Standard Deviation' Solutions obtained for different benchmark functions

Figure 6.18 compares standard deviation obtained by all mathematical benchmark functions, which are used for optimization. When comparing standard deviation solutions on all benchmark functions with other variants of Bat Algorithm, value lies between 3.99E-04 and 3.30E+00 seems to reasonable and productive. The best value of median obtained using Schaffer in comparison to 13 other mathematical benchmark functions, i.e. 6.07E-07.

6.6 Summary

The performance of two variants of Bat Algorithm, i.e. RD-Bat, CATD-Bat and FBI-BA are evaluated over 13 mathematical benchmark functions. The performance of all three variants of Bat Algorithm is dependent on optimization problem at hand. The next chapter presents conclusion of this research work, followed by strengths, weaknesses of Bat Algorithm and future scope of BA.

CHAPTER 7
CONCLUSION AND FUTURE SCOPE

Bat Algorithm is one of the widely used and prominent swarm intelligence based optimization algorithm which has been used for solving different types of problems.

In this research work, existing variants of Bat Algorithm are studied. The motive of doing literature survey is to examine behavior of different variants of Bat Algorithm, to do performance evaluation and to identify research gaps, where advancements can be introduced to propose an improved variant of Bat Algorithm. It has been noticed during literature survey that Bat Algorithm focuses on solving combinatorial optimization problems. Despite of the fact that Bat Algorithm can be used to solve discrete problems, literature survey related to same is quite limited. Here, study related to Bat Algorithm variants is carried out. This work focuses on Standard Bat Algorithm and variants of the same, developed by various researchers by modifying one or the other aspect or by integrating biological features of bats. This work has extensively reviewed advancements introduced in Standard Bat Algorithm and also reviewed biological characteristics of bats, which has impact on performance improvements in comparison to Standard Bat Algorithm. Three different variants of Bat Algorithm are proposed in this work, primarily focuses on parameter tuning. However, another motive of this research is to enhance accurateness of optimal solution obtained, convergence rate towards global optimal solution of problem at hand.

Here, three variants of Bat Algorithm are proposed, implemented and then evaluated their performance with respect to Standard Bat Algorithm. The author who has developed Bat Algorithm, mentioned in research work that bats compute distance between prey and itself ‘in a magical way’. The first variant of Bat Algorithm is enthused from range determination feature of bats. Bats compute the correlation of sound produced and the echo received, to determine location of prey. Based on this way of computing distance or determining range, first variant of Bat Algorithm is developed. This way of computing distance is integrated with Constant Bearing strategy which is used to capture preys moving at predictable speed. The performance is evaluated against Standard Bat Algorithm by varying population of bats against different number of iterations. The results are evaluated by considering Mean, Median, Standard Deviation, Best and Worst results obtained. The second variant of Bat Algorithm is inspired from another fact related to bats, i.e. pursuit strategy of bats vary depending upon flight behavior adopted by preys. Constant Absolute Target

Detection strategy is adopted by bats, when they have to target preys moving erratically or moving at unpredictable speed. Inspired from this behavior of bats, another variant of Bat Algorithm is developed. Result evaluation is carried out in consideration to Standard Deviation, Mean, Median, Best and Worst solutions obtained over varying iterations from [250, 500, 750, 1000] and by varying bat population from [25, 50, 75, 100]. The third variant of Bat Algorithm is inspired from fact that bats do seize their vocalization in presence of other bats, in order to either avoid signal jamming or to save energy, while targeting some other bats' prey. Bats start following other bats, if they receive sound pertaining information of prey capturing. Here one bat will act as leader and other will act as follower. The motive of third variant of Bat Algorithm is to determine direction of follower with respect to leader and utilize this information for obtaining optimal result. The calculations are done considering angle between follower and leader, with respect to an external surface. Here, results are evaluated for different bat population over varying iterations.

Newly developed variants of Bat Algorithm have proven their applicability in providing solutions to numerous number of real world applications. To conclude the research work carried out in this thesis, next section presents strengths of Bat Algorithm and areas where there is a scope of improvement in Bat Algorithm.

7.1 Strengths of Bat Algorithm

- Bat Algorithm has proven its applicability for solving numerous problems related to different application areas over other existing swarm intelligence techniques. This is because of fact that BA has combined advantages of existing optimization techniques.
- The concept of Bat Algorithm is very concise and clear in understanding for layman, which has also increased its acceptance for solving single objective or multi-objective problems.
- Bat Algorithm offers good exploitation capability, which helps in obtaining global optimal solution.

- Bat Algorithm diversify solutions obtained among population, rather than sticking to previously obtained global optimal solution.
- Due to automatic parameter tuning and automatic zooming, BA offers faster convergence rate among candidate/feasible solutions of bat population.
- It is good at obtaining global optimal solution in comparison to local optimal solution.

7.2 Weaknesses of Bat Algorithm

- Bat Algorithm lacks of performing exploration at a better rate in order to obtain local optimal solution.
- BA does parameter tuning, but still there is a scope of improvement by introducing more parameters, which fine-tune the results obtained.
- Switching between exploration and exploitation phase is done when local solutions are obtained. But, decision regarding shift from exploration phase to exploitation phase, needs improvement. This shift must takes place at right moment, with improved control strategy over both phases.
- To enhance performance of Bat Algorithm, still there is a scope of improvement in convergence rate.

7.4 Future Scope

- Inclusion of bats' characteristics in existing variants of Bat Algorithm will improve performance while obtaining optimal solutions of the problem at hand. For example, Doppler Effect property of sound produced and echo received, incorporated in Standard Bat Algorithm, and has boosted performance of the said algorithm to a greater extent. The way bats jam the signal or sound produced by other bats to capture prey before any other bat does, can be used in military applications while receiving signals of opponents, without consuming own resources. Bats create three-dimensional picture of surrounding after receiving echo. There is always a difference in reception of echo by both ears of bats. How internal processing is carried out in bats' ears can also be studied, to generate more accurate results.

- Unravelling multi-objective problems, using existing variants of Bat Algorithms, specifically implemented for solving single-objective problems, is also extension of Bat Algorithm. Generally, multi-objective optimization problems are those problems which deals with more than two objectives. In this research work, three algorithms which are extension of Standard Bat Algorithm are developed and investigated their performance for solving single-objective optimization problem.
- Generally, optimization techniques are developed for unravelling continuous optimization problems. As one cannot apply any optimization technique to solve discrete versions. An extension of these optimization algorithms may target discrete optimization problems. Here, algorithms proposed have targeted only continuous optimization problems. Considering performance of these algorithms, it is expected that these algorithms, if extended further, will be suitable for solving discrete optimization problems.
- Extending applicability to these optimization techniques to solve more number of real world problems. Here, optimization algorithms proposed are applied to solve such optimization problems, which are considered for performance evaluation in existing research work done by other researchers. An extension to this work, is to prove applicability of proposed techniques to solve other engineering problems, or real world applications to model, to control and may be to optimize crane systems in civil engineering applications and many more other fields.
- In this research work, algorithms proposed are modified versions of Standard Bat Algorithm. These algorithms are not hybridized with other evolutionary or other swarm intelligence-based techniques. Empirical studies and observation suggest that integration of pros of one algorithm with pros of another algorithm, will definitely yield more promising results. This strategy of adopting advantages of both algorithms will overshadow shortcomings of both algorithms. Another aspect to this hybridization also suggests that if both algorithms coevolve, then chances of obtaining better results increases. Thus, futuristic work is to focus on how to hybridize both algorithms and to yield most promising results.

- Many SI optimization techniques are required to tune parameters for better functionality and to obtain diverse solutions. Parameter Tuning is one of the aspects which affects performance of Bat Algorithm. As Bat Algorithm consists of many parameters like, frequency, position, pulse emission rate, loudness and velocity. Few parameters have more impact on solution generated as compared to other parameters. Due to this reason, parameter tuning should be done in proper way and analysis is to be done in a sensitive fashion, to fine tune results obtained to select best solution. It is also a fact that tuning of such parameters is merely related to hit and trial method of adjustments, along with some empirical observations. But answer to the question-how to fine tune these parameters is still open for future research.
- Even though, multiple swarm intelligence-based algorithms are proposed and implemented for solving diverse kinds of optimization problems, but still mathematical evaluations of all such algorithms are rare to find. To get in-depth knowledge and understanding of all these algorithms, there is a need for theoretical framework, which analyses performance of these algorithms, mathematically. For illustration, it is difficult to grasp that how defining local rules will help in obtaining self-organized behavior among population of swarm. Moreover, it sometimes lack the key phenomenon behind existence and popularity of swarm intelligence, existence of multiple agents in swarm and their characteristics.
- Almost all the swarm intelligence inspired optimization algorithms are used for solving diverse applications and in most of the cases, range of design variables lies from small scale to medium scale. But, in real world, to solve applications, design variables may lie beyond medium scale. In such cases, there is a need of extension in existing optimization techniques to expand with respect to design variables and to solve large-scale problems. Till now, SI techniques are applied to solve various types of problems, like: traveling salesperson problem, scheduling jobs problems, and many more, and also provided promising results. As these types of problems are NP-hard (i.e. non-deterministic polynomial problems), which are very much difficult to solve for larger number of sizes. Researchers should suggest ways to deal with these types of problems.

- Parameters of swarm intelligence-based optimization techniques are fine-tuned as per underlying problem. These parameters do adapt to different situations, without any human intervention. These algorithms are adaptive in nature and in order to bring variations in parameters, pseudorandom generators or some random variables are used. These algorithms must evolve over a period of time, by learning from choices made in past and their past performances. The main goal of researchers should be to propose, develop and implement wide variety of self-adaptive, self-learning, self-tuning and self-evolved optimization techniques, to solve wide variety of application problems.

The future extensions to Bat Algorithm can be done in three ways as depicted in Figure 7.1. Details of the same are mentioned below:

- Solving Combinatorial Problems
  - Vehicle Routing Problem
  - Scheduling Problem
  - Location and Allocation Problem
  - Assignment Problem
- Hybridization with existing metaheuristic approaches
  - PSO
  - ACO
  - GA and many more
- Inclusion of Bats' biological features

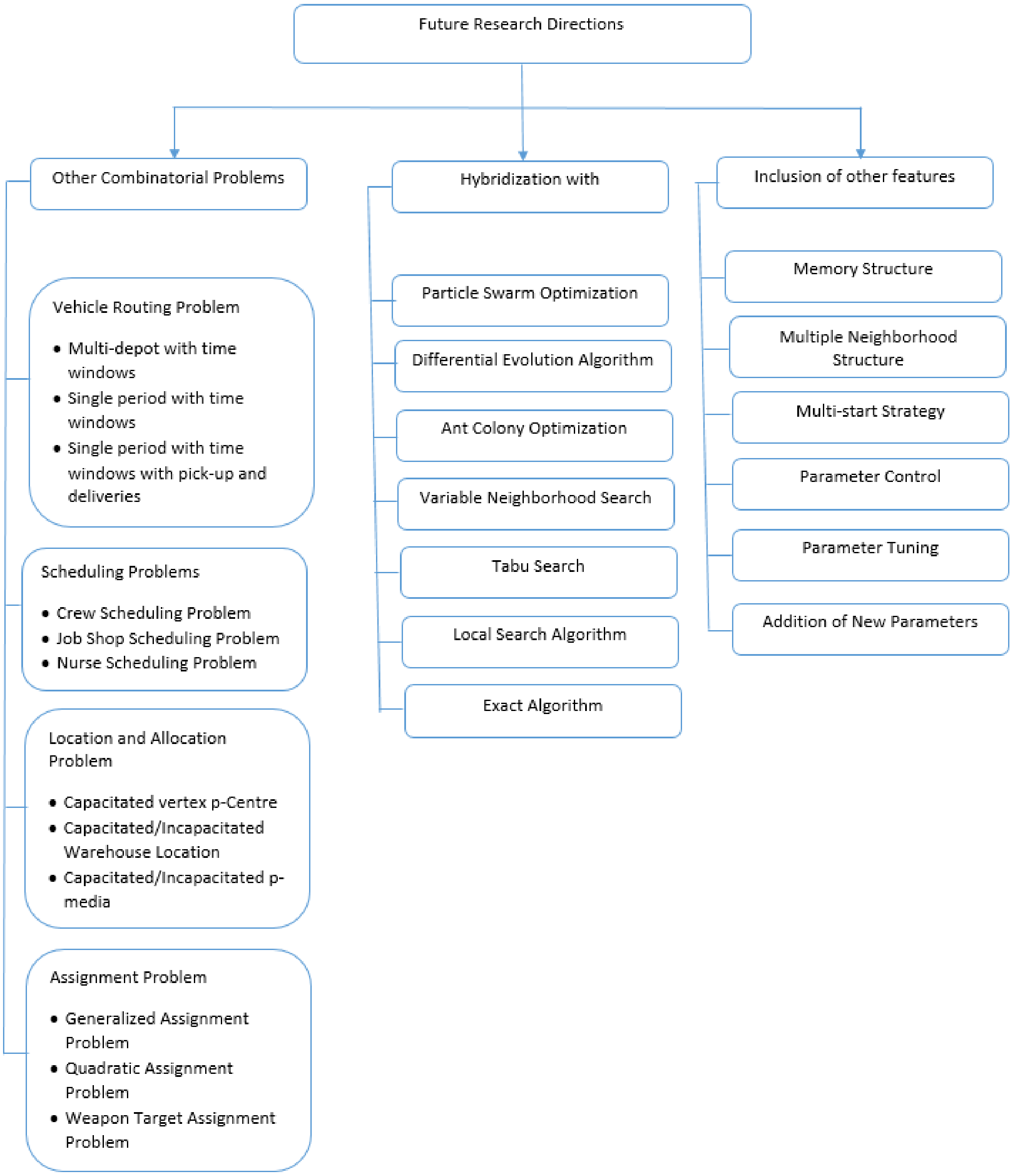


Figure 7.1: Future Research Scope

REFERENCES

[1] A. Banks, J. Vincent, C. Anyakoha, "A review of particle swarm optimization. Part i: background and development," Natural Computing, vol. 6, no. 4, p. 467–484, 2007.

[2] A.C. Enache V. Sgarciu, A. Petrescu-Nita, "Intelligent feature selection method rooted in Binary Bat Algorithm for intrusion detection," in 2015 IEEE 10th Jubilee International Symposium on Applied Computational Intelligence and Informatics (SACI), Timisoara, 2015.

[3] A. Afshar, O. Haddad, M.A.Mariño, B.J.Adams, "Honey-bee mating optimization (hbmo) algorithm for optimal reservoir operation," Journal of the Franklin Institute, vol. 344, no. 5, pp. 452-462, 2007.

[4] A. Alihodzic, M. Tuba, "Improved Bat Algorithm Applied to Multilevel Image Thresholding," The Scientific World Journal, pp. 1-16.

[5] A. Amirjanov, "The development of a changing range genetic algorithm," Computer Methods in Applied Mechanics and Engineering, vol. 195, no. 19-22, pp. 2495-2508, 2006.

[6] A. Antoniou, W. S. Lu, Practical optimization. Algorithms and engineering applications, Springer, 2007.

[7] A. Askarzadeh, "Bird mating optimizer: an optimization algorithm inspired by bird mating strategies," Communication in Nonlinear Science and Numerical Simulation, vol. 19, no. 4, pp. 1213-1228, 2014.

[8] A. Hashmi, N. Goel, S. Goel, D. Gupta, "Firefly algorithm for unconstrained optimization," Journal of Computer Engineering, vol. 11, no. 1, pp. 75-78, 2013.

[9] A. H. Gandomi, X. S. Yang, A. H. Alavi, "Bat Algorithm for constrained optimization tasks," Neural Computing and Applications, vol. 22, no. 6, pp. 1239-1255, 2013.

[10] A. H. Gandomi, A. H. Alavi, "Krill herd: A new bio-inspired optimization algorithm," Communications in Nonlinear Science and Numerical Simulation, vol. 17, no. 12, pp. 4831-4845, 2012.

[11] A. R. Jordehi, "Chaotic bat swarm optimisation (CBSO)," Applied Soft Computing, vol. 26, no. C, pp. 523-530, 2015.

[12] A. R. Conn, K. Scheinberg, P. L. Toint, "Recent progress in unconstrained nonlinear optimization without derivatives," Mathematical Programming, vol. 79, no. 1, pp. 397-414, 1997.

[13] A. Nebro, J. Durillo, J. Garcia-Nieto, C. C. Coello, F. Luna, E. Alba, "Smpso: A new PSO-based metaheuristic for multi-objective optimization," in Proceedings of the IEEE Symposium on Computational Intelligence in Multi-Criteria Decision-Making (MCDM'09), Tennessee, USA, 2009.

[14] A. O. Topal and O. Altun, "Dynamic Virtual Bats Algorithm (DVBA) for Global Numerical Optimization," in 2014 International Conference on Intelligent Networking and Collaborative Systems, Salerno, Italy , 2014.

[15] A. R. Yıldız, "A novel particle swarm optimization approach for product design and manufacturing," International Journal of Advanced Manufacturing Technology, vol. 40, no. 5-6, pp. 617-628, 2009.

[16] A. Sadollah, A. Bahreininejad, H. Eskandar, M. Hamdi, "Mine blast algorithm: A new population based algorithm for solving constrained engineering optimization problems," Applied Soft Computing, vol. 13, no. 5, pp. 2592-2612, 2013.

[17] A. Zhou, B.-Y. Qu, H. Li, S.-Z. Zhao, P. N. Suganthan, Q. Zhang, "Multiobjective evolutionary algorithms: A survey of the state of the art," Swarm and Evolutionary Computation, vol. 1, no. 1, pp. 32-49, 2011.

[18] A. Tharwat, E. H. Houssein, M. M. Ahmed, A. E. Hassanien, "MOGOA algorithm for constrained and unconstrained multi-objective optimization problems," Applied Intelligence, vol. 48, no. 8, pp. 2268-2283, 2018.

[19] A. Novick, "Echolocation in bats: Some aspects of pulse design: During insect pursuits, landings, and obstacle evasions, bats alter the design of their orientation pulses in ways which help us uncover the nature of their sonar," American Scientist, vol. 59, no. 2, pp. 198-209, 1971.

[20] B. Aldeeb, N. M. Norwawi, M. A. Al-Beter, "A Survey on Intelligent Water Drop Algorithm," International Journal of Computers and Technology, vol. 13, no. 10, pp. 5075-5084, 2014.

[21] B. Bansal, A Sahoo, "Bat Algorithm for Full Model Selection in Classification: A Multi-objective Approach," International Journal of Innovations & Advancement in Computer Science, vol. 4, 2015.

[22] B. Mohl, "Target Detection by Echolocating Bats," Animal Sonar: Processes and Performance, pp. 435-450, 1988.

[23] B. Ramesh, V. C. J. Mohan, V. C. V. Reddy, "Application of Bat Algorithm for combined economic load and emission dispatch," Int. J. of Electricl Engineering and Telecommunications, vol. 2, no. 1, pp. 1-9, 2013.

[24] B. Tessema, G. Yen, "A self-adaptive penalty function based algorithm for constrained optimization," in Proceedings of the 2006 IEEE Congress on Evolutionary Computation (CEC'06), 2006.

[25] B. Vogler, G. Neuweiler, "Echolocation in the noctule (nyctalus noctula) and horseshoe bat (rhinolophus ferrumequinum)," Journal of comparative physiology, vol. 152, no. 3, pp. 421-432, 1983.

[26] B. Yang, Y. Chen, Z. Zhao, Q. Han, "A Master-Slave Particle Swarm Optimization Algorithm for Solving Constrained Optimization Problems," in Proceedings of the 2006 Sixth World Congress on Intelligent Control and Automation (WCICA), 2006.

[27] C. A.Coello, "Use of a self-adaptive penalty approach for engineering optimization problems," Computers in Industry, vol. 41, no. 2, pp. 113-127, 2000.

[28] C. A. C. Coello, N. C. Cortés, "Solving multiobjective optimization problems using an artificial Immune System," Genetic Programming and Evolvable Machines, vol. 6, no. 2, pp. 163-190, 2005.

[29] C. A. Coello, E. M. Montes, "Constraint-handling in genetic algorithms through the use of dominance-based tournament selection," Advanced Engineering Informatics, vol. 16, no. 3, pp. 193-203, 2002.

[30] C. A. C. Coello, M. S. Lechuga, "Mopso: A proposal for multiple objective particle swarm optimization," in Proceedings of the 2002 Congress on Evolutionary Computation (CEC'02), Hawaii, USA, 2002.

[31] C. Blum, A. Roli, "Metaheuristics in Combinatorial Optimization: Overview and Conceptual Comparison," ACM Computing Surveys, vol. 35, no. 3, pp. 268-308, 2003.

[32] C. Blum, X. Li, "Swarm Intelligence in Optimization," in Swarm Intelligence: Introduction and Applications, Springer, Heidelberg, 2008, pp. 43-85.

[33] C. Chandrasekar, "An optimized approach of modified Bat Algorithm to record deduplication," International Journal of Computer Applications, vol. 62, no. 1, 2013.

[34] C. Chiu, W. Xian, C. F. Moss, "Flying in silence: Echolocating bats cease vocalizing to avoid sonar jamming," in Proceedings of National Academy of Sciences of the United States of America, 2008.

[35] C. Chiu, P. V. Reddy, W. Xian, "Effects of competitive prey capture on flight behavior and sonar beam pattern in paired big brown bats, Eptesicus fuscus," Journal of Experimental Biology, vol. 213, no. 19, pp. 3348-3356, 2010.

[36] C. F. Tsai, T. K. Dao, W. J. Yang, T. S. Pan, "Parallelized Bat Algorithm with a Communication Strategy," in International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems, 2014.

[37] C. Sur, A. Shukla, "Network, Adaptive and Discrete Real Bat Algorithms for Route Search Optimization of Graph Based Road," in 2013 International Conference on Machine Intelligence and Research Advancement, 2013.

[38] C. Yammani, S. Maheswarapu, S. K. Matam, "A Multi-objective Shuffled Bat Algorithm for optimal placement and sizing of multi distributed generations with different load models," International Journal of Electrical Power & Energy Systems, vol. 79, no. 1, pp. 120-131, 2016.

[39] C. Yang, D. Simon, "A New Particle Swarm Optimization Technique," in ICSENG '05 Proceedings of the 18th International Conference on Systems Engineering, IEEE Computer Society, Washington, DC, USA, 2005.

[40] D. A. Amon, "A Modified Bat Algorithm for Power Loss Reduction in Electrical Distribution System," TELKOMNIKA Indonesian Journal of Electrical Engineering , vol. 14, no. 1, pp. 55-61, 2015.

[41] D. A. Waters, "Bats," The Mammal Society, 2003.

[42] D. Bertsimas, J. Tsitsiklis, "Simulated Annealing," Statistical Science, vol. 8, no. 1, pp. 10-15, 1993.

[43] D. Cvetkovic, I. Parmee, "Evolutionary design and multi–objective optimisation," in Proceedings of the Sixth European Congress on Intelligent Techniques and Soft Computing (EUFIT), Aachen, Germany, 1998.

[44] D. E. Goldberg, “Genetic Algorithms in Search, Optimization and Machine Learning”, Boston, MA, USA: ACM Digital Library, 1989.

[45] D. K. Bahriye Akay, "Artificial bee colony algorithm for large-scale problems and engineering design optimization," Journal of Intelligent Manufacturing, vol. 23, no. 4, pp. 1001-1014, 2012.

[46] D. Karaboga, B. Akay, "Artificial Bee Colony (ABC), Harmony Search and Bees Algorithms on Numerical Optimization," Innovative production machines and systems virtual conference, pp. 1-6, 2009.

[47] D. Karaboga, B. Basturk, "A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm," Journal of Global Optimization, vol. 39, no. 3, pp. 459-471, 2007a.

[48] D. Karaboga, B. Bastruk, "Artificial bee colony (ABC) optimization algorithm for solving constrained Optimization Problems," International Fuzzy Systems Association World Congress: IFSA 2007: Foundations of Fuzzy Logic and Soft Computing, p. 789–798, 2007b.

[49] D. Jones, S. Mirrazavi, M. Tamiz, "Multi-objective meta-heuristics: An overview of the current state-of-the-art," European Journal of Operational Research, vol. 137, no. 1, pp. 1-9, 2002.

[50] D. Merkle, M. Middendorf, H. Schmeck, "Ant Colony Optimization for Resource-Constrained Project Scheduling," IEEE Transactions on Evolutionary Computation , vol. 6, no. 4, 2001.

[51] D. R. Griffin, J. H. Friend, F. A. Webster, "Target discrimination by the echolocation of bats," Journal of Experimental Zoology, vol. 158, no. 2, pp. 155-168, 1965.

[52] D. Rodrigues, L. A. M. Pereira, R. Y. M. Nakamura, "A wrapper approach for feature selection based on Bat Algorithm and Optimum-Path Forest," Expert Systems with Applications: An International Journal, vol. 41, no. 5, pp. 2250-2258, 2014.

[53] E. Cuevas, M. Cienfuegos, "A new algorithm inspired in the behavior of the social-spider for for constrained optimization," Expert Systems with Applications, vol. 41, no. 2, pp. 412-425, 2014.

[54] E. Hussein, A. Ibrahem-Ahmed, E. Hassanien, A. Fahmy, "Nature inspired algorithms for solving the community detection problem," Logic Journal of IGPL , vol. 25, no. 6, 2017.

[55] E. Mezura-Montes, C. Coello, "A simple multimembered evolution strategy to solve constrained optimization problems," IEEE Transactions on Evolutionary Computation, vol. 9, no. 1, pp. 1-17, 2005a.

[56] E. Mezura-Montes, C. A. C. Coello, "Useful infeasible solutions in engineering optimization with Evolutionary Algorithms," in Mexican International Conference on Artificial Intelligence: MICAI 2005: Advances in Artificial Intelligence, 2005b.

[57] E. Zahara, Y.-T. Kao, "Hybrid Nelder–Mead simplex search and particle swarm optimization for constrained engineering design problems," Expert Systems with Applications, vol. 36, no. 2, pp. 3880-3886, 2009.

[58] F. Kang, J. Li, Z. Ma, "Rosenbrock artificial bee colony algorithm for accurate global optimization of numerical functions," Information Sciences, vol. 181, no. 16, p. 3508–3531, 2011.

[59] F. Neri and V. Tirronen, "Recent advances in differential evolution: a survey and experimental analysis," Artificial Intelligence Review, vol. 33, no. 1-2, pp. 61-106, 2009.

[60] G. Ciuprina, D. Loan, I. Munteanu, "Use of intelligent-particle swarm optimization in electromagnetics," IEEE Transactions on Magnetics, vol. 38, no. 2, pp. 1037-1040, 2002.

[61] G. Komarasamy, A. Wahi, "An optimized K-means clustering technique using Bat Algorithm," European Journal of Scientific Research, vol. 84, no. 2, pp. 263-273, 2012.

[62] G.J. Diaz, F. Phillips, B. R. Fajen, "Intercepting moving targets: A little foresight helps a lot," Experimental Brain Research, vol. 195, no. 3, pp. 345-360, 2009.

[63] G. S.Wilkinson, "Reciprocal altruism in bats and other mammals," Ethology and Sociobiology, vol. 9, no. 2-4, pp. 85-100, 1988.

[64] G. Wang, L. Guo "A Novel Hybrid Bat Algorithm with Harmony Search for Global Numerical Optimization," Journal of Applied Mathematics, pp. 1-21, 2013.

[65] G. Wang, L Guo, H Duan, L Liu, H Wang, "A Bat Algorithm with mutation for UCAV path planning," The Scientific World Journal, vol. 6, 2012.

[66] H. Afrabandpey, M. Ghaffari, A. Mirzei, "A Novel Bat Algorithm Based on Chaos for Optimization Tasks," in Iranian Conference on Intelligent Systems, Kerman, Bam, 2014.

[67] H. Djelloul, S. Sabba and S. Chikhi, "Binary Bat Algorithm for graph coloring problem," in 2014 Second World Conference on Complex Systems (WCCS), Agadir, Morocco , 2014.

[68] H. Garg, "Solving structural engineering design optimization problems using an artificial bee colony," Journal of Industrial and Management Optimization, vol. 10, no. 3, pp. 777-794, 2014.

[69] H. Liu, Y. W. Zixing Cai, "Hybridizing particle swarm optimization with differential evolution for constrained numerical and engineering optimization," Applied Soft Computing, vol. 10, no. 2, pp. 629-640, 2010.

[70] H. T. Arita, M. B. Fenton, "Flight and echlocation in the ecology and evolution of bats," Trends in Ecology & Evolution, vol. 12, no. 2, pp. 53-58, 1997.

[71] H. U. Schnitzler, D. Menne, H. Hackbarth, "Range Determination by Measuring Time Delays in Echolocating Bats," Time Resolution in Auditory Systems, pp. 180-204, 1985.

[72] H. U. Schnitzler, E. K. Kalko, "Echolocation by Insect-Eating Bats: We define four distinct functional groups of bats and find differences in signal structure that correlate with the typical echolocation tasks faced by each group," AIBS Bulletin 51, vol. 51, no. 7, pp. 557-569, 2001.

[73] H. U. Schnitzler, E. Flieger, "Detection of oscillating target movements by echolocation in the Greater Horseshoe bat," Journal of comparative physiology, vol. 153, pp. 385-391, 1983.

[74] I Jr. Fister, D. Fister, I. Fister, "Differential evolution strategies with random forest regression in the Bat Algorithm," in Proceedings of the 15th annual conference companion on Genetic and evolutionary computation, Amsterdam, The Netherlands, 2013.

[75] I. Fister Jr., D. Fister, X. S. Yang, "A hybrid Bat Algorithm," Electrotehniški Vestnik, vol. 1, no. 2, pp. 1-7, 2013.

[76] I. Fister, X. S. Yang, I. Fister, J. Brest, "Memetic firefly algorithm for combinatorial optimization," Optimization and Control, 2012.

[77] I. Fister, S. Rauter, X. S. Yang, K. Ljubic, "Planning the sports training sessions with the Bat Algorithm," Neurocomputing , vol. 149, no. 2, pp. 99-1002, 2015.

[78] I. P. Stanimirovic, "Compendious Lexicographic Method for Multi-objective Optimization," Series Mathematics Informatics, vol. 27, p. 55–66, 2012.

[79] J. A. Ruiz-Vanoye, O. Diaz-Parra, F. Cocon, "Meta-Heuristics Algorithms based on the Grouping of Animals by Social Behavior for the Traveling Salesman Problem," ARGIS:International Information System for the Agricultural Science and Technology, 2012.

[80] J. A. Simmons, D. J. Howell, N. Suga, "Information content of bat sonar echoes: recent research on echolocation in bats identifies some of the kinds of information conveyed by echoes of their sonar sounds," American Scientist, vol. 63, no. 2, pp. 204-215, 1975.

[81] J. A. Simmons, R. A. Stein, "Acoustic imaging in bat sonar: Echolocation signals and the evolution of echolocation," Journal of comparative physiology, vol. 135, no. 1, pp. 61-84, 1980.

[82] J. A. Simmons, "A view of the world through the bat's ear: The formation of acoustic images in echolocation," Cognition, vol. 33, no. 1-2, pp. 155-199, 1989.

[83] J. Brest, S. Greiner, B. Boskovic, M. Mernik, V. Zumer, "Self-adapting control parameters in differential evolution: a comparative study on numerical benchmark problems," IEEE Transactions on Evolutionary Computation, vol. 10, no. 6, p. 646–657, 2006.

[84] J. Castro-Gutierrez, D. Landa-Silva, J. A. Moreno-Pérez, "CODEA-An Agent Based Multi-Objective Optimization Framework," in VII Congreso Espaol sobre Metaheursticas, Algoritmos Evolutivos y Bioinspirados (MAEB 2010), 2010.

[85] J. Castro-Gutierrez, D. Landa-Silva, J. M. Perez, "Improved dynamic lexicographic ordering for Multi-Objective Optimisation," in International Conference on Parallel Problem Solving from Nature:PPSN 2010: Parallel Problem Solving from Nature, PPSN XI, 2010.

[86] J. D. Altringham, "Bats: biology and behaviour," The Oxford University Press, 1996.

[87] J. D. Pye, "A theory of echolocation by bats," The Journal of Laryngology & Otology, vol. 74, no. 10, pp. 718-729, 1960.

[88] J. J. Liang, T. P. Runarsson, E. Mezura-Montes, M. Clerc, P. N. Suganthan, C. C. Coello, K. Deb, "Problem definitions and evaluation criteria for the CEC2006 special session on constrained real parameter," Nanyang Technological University, Singapore, 2006.

[89] J.H. Lin, C. W. Chou, C. H. Yang, H. L. Tsai, "A bio-inspired optimization algorithm for modeling the dynamics of biological systems," in Proceedings of the 2012 Third International Conference on Innovations in Bio-Inspired Computing and Applications (IBICA), 2012a.

[90] J.H. Lin, C. W. Chou, C. H. Yang, "A chaotic levy flight Bat Algorithm for parameter estimation in nonlinear dynamic biological systems," Computer and Information Technology, vol. 2, no. 2, pp. 56-63, 2012b.

[91] J. H. Holland, "Genetic Algorithms and the Optimal Allocation of Trials," SIAM Journal on Computing, vol. 2, no. 2, pp. 88-105, 1973.

[92] J. Kennedy and R. C. Eberhart, "Particle swarm optimization," in Proceedings of ICNN'95 - International Conference on Neural Networks, Perth, WA, Australia, 1995.

[93] J. Moore, R. Chapman, "Application of particle swarm to multiobjective optimization," Department of Computer Science and Software Engineering, Auburn University, 1999.

[94] J. Pérez, F. Valdez, O. Castillo, "A New Bat Algorithm with Fuzzy Logic for Dynamical Parameter Adaptation and Its Applicability to Fuzzy Control Design," Fuzzy Logic Augmentation of Nature-Inspired Optimization Metaheuristics, vol. 574, pp. 65-79, 2014.

[95] J. Pugh, A. Martinoli, Y. Zhang, "Particle swarm optimization for unsupervised robotic learning," in Proceedings 2005 IEEE Swarm Intelligence Symposium, 2005. SIS 2005, 2005.

[96] J. T. Robinson, Y. Rahmat-Samii, "Particle Swarm Optimization in Electromagnetics," IEEE Trans. on Antennas and Propagation, vol. 52, no. 2, pp. 397-407, 2004.

[97] J. Xie, Y. Zhou, H. Chen,"A Novel Bat Algorithm Based on Differential Operator and Lévy Flights Trajectory," Computational Intelligence and Neuroscience, pp. 1-13, 2013.

[98] J. W. Zhang, G. G. Wang, "Image Matching Using a Bat Algorithm with Mutation," Applied Mechanics and Materials, vol. 203, no. 1, pp. 88-93, 2012.

[99] J. Zhou, M. Wang, "Cloud Manufacturing Service Paradigm for Group Manufacturing Companies," Advances in Mechanical Engineering, 2015.

[100] K. E. Parsopoulos, M. N. Vrahatis, "Unified particle swarm optimization for solving constrained Engineering Optimization Problems," in International Conference on Natural Computation:ICNC 2005: Advances in Natural Computation, 2005.

[101] K. Ghose, T. K. Horiuchi, P. S. Krishnaprasad, "Echolocating Bats Use a Nearly Time-Optimal Strategy to Intercept Prey," PLoS Biology, vol. 4, no. 5, pp. 865-873, 2006.

[102] K. Khan, A. Nikov, A. Sahai, "A Fuzzy Bat Clustering Method for Ergonomic Screening of Office Workplaces," in Third International Conference on Software, Services and Semantic Technologies S3T 2011.

[103] K. Khan, A Sahai, A Campus, "A fuzzy c-means bi-sonar-based metaheuristic optomization algorithm," IJAMI, vol. 7, pp. 26-32, 2012a.

[104] K. Khan, A Sahai , "A Comparison of BA, GA, PSO, BP and LM for Training Feed forward Neural Networks in e-Learning Context," International Journal of Intelligent Systems and Applications, vol. 7, no. 1, pp. 23-29, 2012b.

[105] K. Parsopoulos, M. N. Vrahatis, "Particle swarm optimization method in multiobjective problems," in Proceedings of ACM Symposium on Applied Computing, Madrid, Spain, 2002.

[106] K. Prajna, G. S. Rao, K. V. V. S. Reddy, R. U. Maheswari, "Application of Bat Algorithm in dual channel speech enhancement," in 2014 International Conference on Communications and Signal Processing (ICCSP), 2014.

[107] K. S. Lee, Z. W. Geem, "A new meta-heuristic algorithm for continuous engineering optimization: harmony search theory and practice," Computer Methods in Applied Mechanics and Engineering, vol. 194, no. 36-38, pp. 3902-3933, 2005.

[108] L. C. Cagnina, S. C. Esquivel, C. A. Coello, "Solving Engineering Optimization Problems with the Simple Constrained Particle Swarm Optimizer," Informatica, vol. 32, no. 1, pp. 319-326, 2008.

[109] L. D. S. Coelho, V. C. Mariani, "Use of chaotic sequences in a biologically inspired algorithm for engineering design optimization," Expert Systems with Applications, vol. 13, no. 3, pp. 1905-1913, 2008.

[110] L. Gao, D. Zou, Y. Ge, W. Jin, "Solving pressure vessel design problems by an effective global harmony search algorithm," in Proceedings of the 2010 Chinese on Control and Decision Conference, Xuzhou, Chinese, 2010.

[111] L. Hansheng, K. Lishan, "Balance between exploration and exploitation in genetic search," Wuhan University Journal of Natural Sciences, vol. 4, no. 1, pp. 28-32, 1999.

[112] L. Li, Y. Zhou, "A novel complex-valued Bat Algorithm," Neural Computing and Applications, vol. 25, pp. 1369-1381, 2014.

[113] L. Lovász, "Discrete and Continuous: Two Sides of the Same?," Visions in Mathematics, pp. 359-382, 2010.

[114] L. Jiao, L. Li, R. Shang, F. Liu, R. Stolkin, "A novel selection evolutionary strategy for constrained optimization," Information Sciences, vol. 239, pp. 122-141, 2013.

[115] L. Jun, L. Liheng, W. Xianyi, "A double-subpopulation variant of the Bat Algorithm," Applied Mathematics and Computation, vol. 263, pp. 361-377, 2015.

[116] L. K. Denault, D.A. McFarlane, "Reciprocal altruism between male vampire bats, Desmodus rotundus," Animal Behaviour, vol. 49, no. 3, pp. 855-866, 1995.

[117] L. L. Laudis, S. Shyam, C. Jemila, V. Suresh, "MOBA:Multi Objective Bat Algorithm for Combinatorial Optimization in VLSI," in 6th International Conference on Smart Computing and Communications, ICSCC 2017, Kurukshetra, India, 2017.

[118] L. Mohamed Amine, K. Jadjet, "A Multi-objective Binary Bat Algorithm," in Proceedings of the International Conference on Intelligent Information Processing, Security and Advanced Communication, Batna, Algeria, 2015.

[119] L. Wang, L.-P. Li, "An effective differential evolution with level comparison for constrained engineering design," Structural and Multidisciplinary Optimization, vol. 41, no. 6, pp. 947-963, 2010.

[120] M. Airas, "Echolocation in bats," in postgrad seminar course of HUT Acoustics Laboratory, 2003.

[121] M. A. Abido, "Multiobjective particle swarm optimization for environmental/economic dispatch problem," Electric Power Systems Research, vol. 79, no. 7, pp. 1105-1113, 2009.

[122] M. A. Al-Betar, A. T. Khader, "University Course Timetabling Using a Hybrid Harmony Search Metaheuristic Algorithm," IEEE Transactions on Systems Man and Cybernetics Part C (Applications and Reviews), vol. 42, no. 5, pp. 664-681, 2012.

[123] M. Airas, "Echolocation in bats," in Proceedings of spatial sound perception and reproduction, Helsinki, Finland, 2003.

[124] M. A. Tawfeeq, "Intelligent Algorithm for Optimum Solutions Based on the Principles of Bat Sonar," Neural and Evolutionary Computing, vol. 10, no. 10, pp. 1-9, 2012.

[125] M. B. Fenton, "Natural history and biosonar signals," In Hearing by bats, pp. 37-86, 1995.

[126] M. Campos, R. A. Krohling, I. Enriquez, "Bare bones particle swarm optimization with scale matrix adaptation," IEEE Transactions on Cybernetics, vol. 44, no. 9, pp. 1567-1578, 2014.

[127] M. D. Jolly, A. D. Lothes, L. S. Bryson, L. Ormsbee, "Research Database of Water Distribution System Models," Journal of Water Resources Planning and Management, vol. 140, no. 4, pp. 410-416, 2014.

[128] M. D. Tuttle, "Bats, artificial roosts, and mosquito control," Bat Conservation International, 2006.

[129] M. H. Afshar, "Extension of the constrained particle swarm optimization algorithm to optimal operation," International Journal of Electrical Power & Energy Systems, vol. 51, pp. 71-81, 2013.

[130] M. Gilli, P. Winker, "Review of Heuristic Optimization Methods in Econometrics," MRTN-CT-2006-034270 COMISEF, 2008.

[131] M. K. Marichelvam, T. Prabaharam, "A Bat Algorithm for realistic hybrid flowshop scheduling problems to minimize makespan and mean flow time," ICTACT Journal on Soft Computing, vol. 3, no. 1, pp. 428-433, 2012.

[132] M. R. Sierra, C. A. C. Coello, "Improving pso-based multi-objective optimization using crowding, Mutation and ∈-Dominance," in International Conference on Evolutionary Multi-Criterion Optimization:EMO 2005: Evolutionary Multi-Criterion Optimization, 2005.

[133] M. Reyes-Sierra, C. A. C. Coello, "Multi-objective particle swarm optimizers: A survey of the state-of the-art," International Journal of Computational Intelligence Research, vol. 2, no. 3, pp. 287-308, 2006.

[134] M. Sharawi, I. A. Saroit, H. EI-Mahdy, E. Emary "Routing Wireless Sensor Networks Based on Soft Computing Paradigms: Survey," International Journal on Soft Computing, Artificial Intelligence and Applications (IJSCAI, vol. 2, no. 4, pp. 1-15, 2015.

[135] M. Zhang, W. Luo, X. Wang, "Differential evolution with dynamic stochastic selection for constrained optimization," Information Sciences, vol. 178, no. 15, p. 3043–3074, 2008.

[136] M. E. Jensen, C. F. Moss, A. Surlykke, "Echolocating bats can use acoustic landmarks for spatial orientation," Journal of Experimental Biology, vol. 208, pp. 4399-4410, 2005.

[137] N. E.Y. Kouba, M. Menaa, M. Hasni, "Load Frequency Control in multi-area power system based on Fuzzy Logic-PID Controller," in 2015 IEEE International Conference on Smart Energy Grid Engineering (SEGE), 2015.

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

[139] N. M. Rivers, R. K. Butlin, J. D. Altringham, "Autumn swarming behaviour of natterer’s bats in the UK: Population size, catchment area and dispersal," Biological Conservation, vol. 127, no. 2, pp. 215-226, 2006.

[140] N. M. Nawi, M. Z. Rehman, A. Khan, "A New Bat Based Back-Propagation (BAT-BP) Algorithm," in Springer-Verlag Berlin Heidelberg, 2013.

[141] N. Suga, "Biosonar and neural computation in bats," Scientific American, vol. 262, no. 2, pp. 60-71, 1990.

[142] N. S. Jaddi, S. Abdullah, A. R. Hamdan, "Optimization of neural network model using modified bat-inspired algorithm," Applied Soft Computing, vol. 37, pp. 71-86, 2015.

[143] P. Musikapun, P. Pongcharoen, "Solving Multi-Stage Multi-Machine Multi-Product Scheduling Problem Using Bat Algorithm," in 2012 2nd International Conference on Management and Artificial Intelligence, Singapore, 2012.

[144] P. Ngatchou, A. Zarei , A. El-Sharkawi, "Pareto multi objective optimization," in Proceedings of the 13th International Conference on Intelligent Systems Application to Power Systems, Virginia, USA, 2005.

[145] Q. He, L. Wang, "An effective co-evolutionary particle swarm optimization for constrained engineering design problems," Engineering Applications of Artificial Intelligence, vol. 20, no. 1, pp. 89-99, 2007a.

[146] Q. He, L. Wang, "A hybrid particle swarm optimization with a feasibility-based rule for constrained optimization," Applied Mathematics and Computation, vol. 186, no. 2, pp. 1407-1422, 2007b.

[147] R. B. Statnikov, I. B. Matusov, "Optimization in engineering problems—Analysis and generalizations," Journal of Machinery Manufacture and Reliability, vol. 42, no. 4, pp. 289-292, 2013.

[148] R. Damodaram, M. L. Valamathi, "Phishing website detection and optimization using Modified Bat Algorithm," International Journal of Engineering Research and Applications, vol. 2, no. 1, pp. 870-876, 2012.

[149] R. E. Stebbings, D. W. Yalden, J. S. Herman, “Which bat is it? A guide to bat identification in Great Britain and Ireland”, The Mammal Society, 2007.

[150] R. Kumar, "Directed bee colony optimization algorithm," Swarm and Evolutionary Computation, vol. 17, pp. 60-73, 2014.

[151] R. L. Becerra, C. A. Coello, "Cultured differential evolution for constrained optimization," Computer Methods in Applied Mechanics and Engineering, vol. 195, no. 33-36, pp. 4303-4322, 2006.

[152] R. S. Parpinelli and H. S. Lopes, "New inspirations in swarm intelligence: A survey," International Journal of Bio-Inspired Computation , vol. 3, no. 1, pp. 1-16, 2011.

[153] R. S. Parpinelli, H. S. Lopes, A. A. Fretias, "Data Mining With an Ant Colony Optimization Algorithm," IEEE Transactions On Evolutionary Computing, vol. 6, no. 4, pp. 321-332, 2002.

[154] R. Tang, S. Fong, X. S. Yang, S. Deb, "Wolf search algorithm with ephemeral memory," in 2012 Seventh International Conference on Digital Information Management (ICDIM), 2012.

[155] R. V. Kulkarni, A. Forster, G. K. Venayagamoorthy, "Computational Intelligence in Wireless Sensor Networks: A Survey," IEEE Communications Surveys & Tutorials, vol. 13, no. 1, pp. 68-96, 2011.

[156] R. V. Rao, V. J. Savsani, D. P. Vakharia, "Teaching–learning-based optimization: a novel method for constrained mechanical design optimization problems," Computer-Aided Design, vol. 43, no. 3, pp. 303-315, 2011.

[157] R. Y. Nakamura, L. A. Pereira, K. A. Costa, D. Rodrigues, J. P. Papa, X. S. Yang, "BBA: A Binary Bat Algorithm for Feature Selection," in 2012 25th SIBGRAPI Conference on Conference: Graphics, Patterns and Images (SIBGRAPI), 2012.

[158] S. Akhtar, A. Ahmad and E. Abdel-Rahman, "A Metaheuristic Bat-Inspired Algorithm for Full Body Human Pose Estimation," in 2012 Ninth Conference on Computer and Robot Vision, Toronto, ON, Canada, 2012.

[159] S. Biswal, A.K. Barisal, A. Behera, T. Prakash "Optimal power dispatch using Bat Algorithm," in 2013 International Conference on Energy Efficient Technologies for Sustainability (ICEETS), 2013.

[160] S. Goyal, M. S. Patterh, "Modified Bat Algorithm for Localization of Wireless Sensor Network," Wireless Personal Communications, 2015.

[161] S. B. Hamida and M. Schoenauer, "Aschea: new results using adaptive segregational constraint handling," in Proceedings of the 2002 Congress on Evolutionary Computation (CEC’02), Honolulu, USA, 2002.

[162] S. Koziel, Z. Michalewicz, "Evolutionary algorithms, homomorphous mappings, and constrained parameter optimization," Applied Mathematics and Computation, vol. 7, pp. 19-44, 1999.

[163] S. L. Voigt-Heucke, M. Taborsky, D. K. N. Dechmann, "A dual function of echolocation: bats use echolocation calls to identify familiar and unfamiliar individuals," Animal Behaviour, vol. 80, no. 1, pp. 59-67, 2010.

[164] S. Saha, S. Bandyopadhyay, "A generalized automatic clustering algorithm in a multiobjective framework," Applied Soft Computing, vol. 13, no. 1, pp. 89-108, 2013.

[165] S. S. Rao, “Engineering Optimization: Theory and Practice”, John Wiley and Sons, 2009.

[166] S. Mishra, K. Shaw, D. Mishra, "A New Meta-heuristic Bat Inspired Classification Approach for Microarray Data," Procedia Technology, vol. 4, pp. 802-806, 2012.

[167] S. Kirkpatrick, C. D. Gelatt, M. P. Vecchi, "Optimization by Simulated Annealing," Science, vol. 220, no. 4598, pp. 671-680, 1983.

[168] S. Mirjalili, S. M. Mirjalili, X. S. Yang, "Binary Bat Algorithm," Neural Computing and Applications , vol. 25, no. 3-4, pp. 663-681, 2014.

[169] S. Raghavan, P. Sarwesh, C. Marimuthu and K. Chandrasekaran, "Bat Algorithm for scheduling workflow applications in cloud," in 2015 International Conference on Electronic Design, Computer Networks & Automated Verification (EDCAV), 2015.

[170] S. Sagnika, S. Bilgaiyan, B. S. P. Mishra, "Workflow Scheduling in Cloud Computing Environment Using Bat Algorithm," in Proceedings of First International Conference on Smart System, Innovations and Computing, 2018.

[171] S. Sivakumar, R. Venkatesan, "Meta-heuristic approaches for minimizing error in localization of wireless sensor networks," Applied Soft Computing, vol. 3047, no. 1, pp. 1-13, 2015.

[172] S. Yilmaz, E. U. Kucuksille, "A new modification approach on Bat Algorithm for solving optimization problems," Applied Soft Computing, vol. 28, pp. 259-275, 2015.

[173] S. Yilmaz, E. U. Kucuksille, "Improved Bat Algorithm (IBA) on Continuous Optimization Problems," Suleyman Demirel University, Isparta, Turkey, 2013.

[174] T. A. Lemma, F. B. M. Hashim, "Use of Fuzzy Systems and Bat Algorithm for Exergy Modeling of an Industrial Gas Turbine," in IEEE Colloquium on Humanities, Science and Engineering Research (CHUSER2011), 2011.

[175] T. C. Bora, L. S. Coelho, "Bat-Inspired Optimization Approach for the Brushless DC Wheel Motor Problem," IEEE Transactions on Magnetics, vol. 48, no. 2, pp. 947-950, 2012.

[176] T. C. Havens, C. J. Spain, N. G. Salmon, J. M. Keller, "Roach infestation optimization," in Proceedings of the IEEE,Swarm Intelligence Symposium (SIS 2008), St. Louis, USA, 2008.

[177] T.-J. Hsieh, "A bacterial gene recombination algorithm for solving constrained optimization problems," Applied Mathematics and Computation, vol. 231, no. 1, p. 187–204, 2014.

[178] T. Navalertporn, N. Afzulpurkar, "Optimization of tile manufacturing process using particle swarm optimization," Swarm and Evolutionary Computation, vol. 1, no. 2, pp. 97-109, 2011.

[179] T. Ray, K. Liew, "Society and civilization: An optimization algorithm based on the simulation of social behavior," IEEE Transactions on Evolutionary Computation, vol. 7, no. 4, pp. 386-396, 2003.

[180] T. Runarsson, X. Yao, "Search biases in constrained evolutionary optimization," IEEE Transactions on Systems, Man, and Cybernetics, vol. 35, no. 2, pp. 233-243, 2005.

[181] T. S. Pan, T. K. Dao, S. C. Chu, "Hybrid Particle Swarm Optimization with Bat Algorithm," in Genetic and Evolutionary Computing: Proceeding of the Eighth International Conference on Genetic and Evolutionary Computing, 2015.

[182] T. Takahama and S. Sakai, "Constrained optimization by applying the a constrained method to the nonlinear simplex method with mutations," IEEE Transactions on Evolutionary Computation, vol. 9, no. 5, pp. 437-451, 2005.

[183] V. Cerny, "Thermodynamical Approach to the Traveling Salesman Problem: An Efficient Simulation Algorithm," Journal of Optimization Theory And Applications, vol. 45, no. 1, pp. 41-51, 1985.

[184] V. U. Reddy, A. Manoj, "Optimal Capacitor Placement for Loss Reduction in Distribution Systems Using Bat Algorithm," IOSR Journal of Engineering (IOSRJEN), vol. 2, no. 10, pp. 23-27, 2012.

[185] W. Gong, Z. Cai, D. Liang, "Engineering optimization by means of an improved constrained differential evolution," Computer Methods in Applied Mechanics and Engineering, vol. 268, no. 1, pp. 884-904, 2014.

[186] X. B. Meng, X. Z. Gao, Y. Liu, H. Zhang, "A novel Bat Algorithm with habitat selection and Doppler effect in echoes for optimization," Expert Systems with Applications, vol. 42, no. 17, pp. 6350-6364, 2015.

[187] X. Liang, W. Li, P. Liu, Y. Zhang, A. A. Agbo, "Social network-based swarm optimization," in Proceedings of the 2015: 12th International Conference on Networking, Sensing and Control, Taipei, Taiwan, 2015.

[188] X. S. Yang, "A New Metaheuristic Bat-Inspired Algorithm," Optimization and Control, vol. 284, pp. 65-74, 2010.

[189] X. S. Yang, S. Deb, "Cuckoo Search: Recent Advances and Applications," Neural Computing and Applications, vol. 24, no. 1, pp. 169-174, 2014.

[190] X. S. Yang, "Bat Algorithm for multi-objective optimisation," International Journal of Bio-Inspired Computation, vol. 3, no. 5, pp. 267-274, 2011.

[191] X.-S. Yang, "Engineering optimizations via nature-inspired virtual bee algorithms," in IWINAC 2005: Artificial Intelligence and Knowledge Engineering Applications: A Bioinspired Approach, 2005.

[192] X. S. Yang, "Bat Algorithm: Literature Review and Applications," International Journal of Bio-Inspired Computation, vol. 5, no. 3, 2013.

[193] X. S. Yang, M. Karamanoglu, "Flower Pollination Algorithm: A Novel Approach for Multiobjective Optimization," Optimization and Control, vol. 46, no. 9, pp. 1222-1237, 2014.

[194] X.-S. Yang, S. Deb, "Cuckoo search via lévy flights," in 2009 World Congress on Nature & Biologically Inspired Computing (NaBIC), 2009.

[195] X.-S. Yang, "Firefly algorithms for multimodal optimization," in SAGA 2009: Stochastic Algorithms: Foundations and Applications , 2009.

[196] X. S. Yang, M. Karamanoglu, "Bat Algorithm for Topology Optimization in Microelectronic Applications," in IEEE International Conference on Future Generation Communication Technology (FGCT 2012), 2012.

[197] X. Yang, A. H. Gandomi, "Bat Algorithm: a novel approach for global engineering optimization," Engineering Computations, vol. 29, no. 5, pp. 463-484, 2012.

[198] Y. Fei, H. Yu, X. Jiang, "An improved constrained optimization genetic algorithm," in IEEE International Conference on Intelligent Computing and Intelligent Systems, Xiamen, China, 2010.

[199] Y. Karpat, T. Ozel, "Multi-objective optimization for turning processes using neural network modeling and dynamic-neighborhood particle swarm optimization," The International Journal of Advanced Manufacturing Technology, vol. 35, no. 3-4, pp. 234-247, 2007.

[200] Y. T. Chen, C. S. Shieh, M. F. Horng, B. Y. Liao, J. S. Pan, "A guidable Bat Algorithm based on doppler effect to improve solving efficiency for optimization problems," in International Conference on Computational Collective Technologies and Applications, 2014.

[201] Y. Wang, Z. Cai, "Combining multiobjective optimization with differential evolution to solve constrained Constrained Optimization Problems," IEEE Transactions on Evolutionary Computation, vol. 16, no. 1, pp. 117-134, 2012.

[202] Y. Wang, Y. Z. F. Zixing Cai, "Constrained optimization based on hybrid evolutionary algorithm and adaptive constraint-handling technique," Structural and Multidisciplinary Optimization, vol. 37, no. 4, pp. 395-413, 2009.

[203] Y. Saji, M.E. Riffi, "A novel discrete Bat Algorithm for solving the travelling salesman problem," Neural Computing and Applications, vol. 27, no. 7, pp. 1853-1866, 2016.

[204] Y. Zhou, L. Li, M. Ma, "A Complex-valued Encoding Bat Algorithm for Solving 0–1 Knapsack Problem," Neural Processing Letters, vol. 44, no. 2, 2015.

[205] Y.-T. K. Erwie Zahara, "Hybrid nelder–mead simplex search and particle swarm optimization for constrained engineering design problems," Expert Systems with Applications, vol. 36, no. 2, pp. 3880-3886, 2009.

[206] Z.A.E.M. Dahi, C. Mezioud, A. Draa, "Binary Bat Algorithm: On the Efficiency of Mapping Functions When Handling Binary Problems Using Continuous-variable-based Metaheuristics," in International Conference on Computer Science and IFIP, 2018.

[207] Z. Cui, R. Alex, R. Akerkar, "Recent Advances on Bioinspired Computation," The Scientific World Journal, pp. 1-3, 2014.

[208] Z. Huang, M. Ma, C. Wang, "An archived differential evolution algorithm for constrained global Optimization," in ICSMA 2008, International Conference on Smart Manufacturing Application, Gyeonggi-do, South Korea, 2008.

[209] Z. Li, Z. Shang, J. Liang, B. Niu, "An improved differential evolution for constrained optimization with dynamic constraint-handling mechanism," in Proceedings of the Evolutionary Computation (CEC’12), Brisbane, Australia, 2012.

[210] Z. W. Geem, K. S. Lee, Y. Park, "Application of Harmony Search to Vehicle Routing," American Journal of Applied Sciences, vol. 2, no. 2, pp. 1552-1557, 2005.

LIST OF PUBLICATIONS

1. Shabnam Sharma, Ashish Kr Luhach, Kiran Jyoti. *Research & analysis of advancements in Bat Algorithm.* 3rd International Conference on Computing for Sustainable Global Development (INDIACom), 2016 (IEEE) [Published].
2. Shabnam Sharma, Sahil Verma, Kiran Jyoti. *A New Bat Algorithm with Distance Computation Capability and Its Applicability in Routing for WSN.* International Conference on Soft Computing & Signal Processing, 2018 (Springer) [Published].
3. Shabnam Sharma, Sahil Verma, Kiran Jyoti, Kavita. *Hybrid Bat Algorithm for Balancing Load in Cloud Computing.* Feynman100- 4^{th} International Conference on Computing Sciences and International Journal of Engineering and Technology (Scopus Indexed) [Published].
4. Shabnam Sharma, Sahil Verma, Kiran Jyoti. *RD-Bat: Bat Algorithm as Range Determiner.* Recent Patents on Computer Science (Scopus Indexed) [Accepted]
5. Shabnam Sharma, Sahil Verma, Kiran Jyoti. *A Novel Variant of Bat Algorithm Inspired from CATD-Pursuit Strategy & Its Performance Evaluations.* International Journal of Advanced Intelligent Paradigms (Scopus Indexed) [Accepted]
6. Shabnam Sharma, Sahil Verma, Kiran Jyoti. *Re-inspiring Bat Algorithm for Load Balancing in Cloud Computing Environment.* International Journal of Intelligent Systems and Applications (Scopus Indexed) [Communicated].
7. Shabnam Sharma, Sahil Verma, Kiran Jyoti. *Flight Behaviour Inspired Bat Algorithm & Its Performance Study.* International Journal of Computing and Digital Systems (Scopus Indexed) [Communicated].