Master of Science in Computer Science Engineering

June 2022

A Comparative Study on Optimization Algorithms and its efficiency

Kareem Ahmed Sheik

This thesis is submitted to the Faculty of Computer Science at Blekinge Institute of Technology in partial fulfilment of the requirements for the degree of Master of Science in Computer Science and Engineering. The thesis is equivalent to 20 weeks of full-time studies.

The authors declare that they are the sole authors of this thesis and that they have not used any sources other than those listed in the bibliography and identified as references. They further declare that they have not submitted this thesis at any other institution to obtain a degree.

Contact Information:

Author(s):
Kareem Ahmed Sheik
E-mail: kash17@student.bth.se

University advisor:
Huseyin Kusetogullari
Department of Computer Science and Engineering

Faculty of Computer Science	Internet : www.bth.se
Blekinge Institute of Technology	Phone : +46 455 38 58 76
SE-371 79 Karlskrona, Sweden	Fax : +46 455 38 58 76

ABSTRACT

Background: In computer science, optimization can be defined as finding the most cost-effective or notable achievable performance under certain circumstances, maximizing desired factors, and minimizing undesirable results. Many problems in the real world are continuous, and it isn't easy to find global solutions. However, computer technological development increases the speed of computations [1]. The optimization method, an efficient numerical simulator, and a realistic depiction of physical operations that we intend to describe and optimize for any optimization issue are all interconnected components of the optimization process [2].

Objectives: A literature review on existing optimization algorithms is performed. Ten different benchmark functions are considered and are implemented on the existing chosen algorithms like PSO (Particle Swarm Optimization) Method, GA (Genetic Algorithm), ACO (Ant Colony Optimization) Method and PIBO (Plant Intelligence Behaviour optimization algorithm) to measure the efficiency of these approaches based on the factors or metrics like CPU Time, Optimality, Accuracy and Mean Best Standard Deviation.

Methods: In this research work, a mixed-method approach is used. A literature review is performed on the existing optimization algorithms. On the other hand, an experiment is conducted by using ten different benchmark functions on the current optimization algorithms like PSO algorithm, ACO algorithm, GA and PIBO to measure their efficiency based on the four different factors like CPU Time, Optimality, Accuracy, Mean Best Standard Deviation. This tells us which optimization algorithms perform better.

Results: The experiment findings are represented within this section. Using the standard functions on the suggested method and other methods, the various metrics like CPU Time, Optimality, Accuracy, and Mean Best Standard Deviation are considered, and the results are tabulated. Graphs are made using the data obtained.

Analysis and Discussion: The research questions are addressed based on the literature review and experiment's results that have been conducted.

Conclusion: We finally conclude the research work by analyzing the existing optimization methods and the algorithms performance. The PSO performs much better and can be depicted from the results of the optimal metrics, best mean, standard deviation, accuracy and CPU Time. Other algorithms performed better than PSO when certain benchmark functions are tested.

Keywords: Optimization, Nature-based optimization algorithms, Heuristic Search Algorithms, Benchmark Optimization Problems, Systematic literature review, Benchmark functions, Genetic Algorithm, Plant intelligence based optimization algorithm

ACKNOWLEDGEMENTS

This thesis is presented to the Department of Computer Science & Engineering at Blekinge Institute of Technology (BTH) to partially complete requirements for a Master of Science in Computer Science & Engineering degree.

I sincerely thank my supervisor, Professor Huseyin Kusetogullari, for my thesis's continuous support and understanding.

Secondly, I would like to thank my parents, family, and friends for supporting me throughout my academic journey.

Table of Contents

List of Figures

List of Tables:

1 INTRODUCTION

Many issues are persistent in the actual world, and discovering global solutions becomes tough. Although advances in computer technology boost the speed of calculations, this is frequently insufficient, especially when the size of the issue's instance is big. Using a correct approach to such situations necessitates their linearization [1]. To overcome these problems, optimization algorithms are used and have played a vital role in solving the issues by giving the optimal solutions. There are many optimization algorithms, but the genetic algorithm is the most popular one. Even though there are several optimization algorithms available in the current literature only few of them can perform efficiently on low power edge devices[1][3][4][28]. In a genetic algorithm, "the optimization procedure begins with generating a collection of random solutions to serve as possible solutions for a specific optimization issue [3]". But every algorithm has its pros and cons. Although the Genetic Algorithm is amongst the most widely used algorithms, its fundamental disadvantage is its stochastic character, which results in different solutions in each run and a massive number of objective functions for each run [2]. These genetic algorithm & heuristic approaches played a crucial role in the development of other algorithms like PSO [4], [5], ACO [3], [6] GSO methods [7], [8], and NSGA II [9].

Ant-colony optimization has become a meta-heuristic method motivated by the behavior of actual ants, in which they seek optimum solutions by taking into account both local heuristics and prior knowledge, as measured by pheromone fluctuations [9]. Particle Swarm Optimization has become a hybrid method that uses the diversification approach particle swarm optimization for acquiring global optima & the intensification strategy for finding the optimal solution on local level, as well as the Modified Corner List method [10]. The Gravitational search algorithm was implemented on gravity law & the mass interactions concept. It employs Newtonian physics theory & searcher agents to gather masses [5]. The evolution technique is utilized as a solid baseline technique against that more sophisticated techniques can be evaluated. It may also be helpful in specific real-world situations where local search appears to be superior to or comparable with population-based techniques [7]. This NSGA-II overcomes the computational issue, non-elitism method, & parameter sharing [8].

As previously said, they are all algorithms of heuristic search that handle distinct issues. Because some algorithms provide better results than others, the quest for heuristic methods has become an open topic [11]. Nature often inspires metaheuristic algorithms, which are currently among the most extensively used optimization techniques. They offer several benefits over traditional algorithms, as shown by the many examples provided. Simulated annealing, differential evolution, genetic algorithms, particle swarm optimization, ant & bee algorithms, firefly algorithm, cuckoo search, harmony search, and others are examples of metaheuristic algorithms [2]. The paper is heavily influenced by nature based optimization algorithms and its adaptation to prevailing circumstances and its capacity to withstand environmental influences. With today's technological advances, an optimization method that is adaptable & adapts to changing situations is required. The comparative study in the experiment determines the better optimization algorithm based on the bench mark functions.

1.1 Aim

This thesis aims to do a comparative study of existing optimization algorithms and determine the better algorithm that could solve optimization problems and measure the efficiency of the algorithms with existing benchmark functions.

1.2 Objectives

Key points of the current study are:

- To do a systematic literature review on existing optimization algorithms
- To measure the efficiency of the optimization algorithms based on the factors and using some benchmark functions.

1.3 Research Questions

Following we formulated these research questions to address the objectives:

Q1 What are the optimization algorithms available in the current literature?

Motivation: This research question is formed to know all the nature based algorithms that we have chosen for the research in our literature review and comparative study

Q2. Which optimization algorithm gives optimal performance of CPU Time?

Motivation: This research question is formed to know the optimal performance of the optimization algorithms we have chosen for the experiment.

1.4 Outline

The entire thesis work is outlined as below:

Chapter1: In this thesis work, the Introduction and motivation regarding the thesis topic are clearly determined and further supported with the problem statement, methodology, & research questions.

Chapter 2: In this chapter 2, The Related work for the thesis has been clearly presented. Various existing optimization algorithms have been represented.

Chapter 3: In this chapter 3, we represented the background.

Chapter 4: The methods that we have used in our research are explained and are divided into two sections. The first section gives a brief explanation of Literature Review, and the second section gives an overview of the experiment that has been conducted.

Chapter 5: The results of the optimization algorithm and its comparative study are described.

Chapter 6: This chapter presents the analysis and discussion.

Chapter 7: In chapter 7, the threats to validity have been represented.

Chapter 8: In this chapter 8, the conclusion & future work for presented thesis work have been represented.

2 RELATED WORK

Hjelmfelt et al. [12] suggetsed the chemical implementation and connections among neurons. They postulated that metabolic processes in plants work similarly to Boolean computer logic gates like Or, AND, & NOR, and chemical neurons. The natural behavior of plants inspires a variety of optimization strategies. Yang et al. [13] presented the flower pollination algorithm, which is influenced by the cross-pollination of flowers. According to their simulation findings, the flower method outperforms both the Genetic Algorithm & particle swarm optimization. They also employed the flower approach for solving a nonlinear design criterion, demonstrating that the convergence rate has been almost exponential.

Cui et al. [14] developed a novel method called the Artificial plant optimization algorithm, driven by the tree's development process in a new evolutionary approach. Two apps on ANN training & a toy version of protein folding validate the suggested approach. The simulation results demonstrate that the novel optimization technique is sound and has many applications in different disciplines. Bayat et al. [6] introduced a novel numerical optimization approach for handling complex engineering issues motivated by a strawberry plant. This approach is applied to standard test functions, & the outcomes are compared with the Genetic Algorithm & particle swarm optimization. The simulations demonstrated that the suggested approach could successfully tackle complex optimization issues. S. Zahra Mirjalili et al[1] introduced a grass hopper optimization algorithm for multi-objective optimization problems. Strawberry plant optimization algorithm[6] has been introduced as a numerical based optimization algorithm based on the strawberry plant. Flower Pollination Algorithm[13] has been introduced to solve the global optimization problems. Artificial Plant optimization algorithm[14] has been introduced in the swarm intelligence and bio-inspired computation.

Our work is mainly inspired by the success of the above optimization algorithms. The systematic literature review is done on the nature based optimization algorithms and then 4 algorithms have been chosen to do the experiment based on the 10 different benchmark functions resulting in efficiency.

3 BACKGROUND

Before we get into the following sections, it is important to discuss the existing optimization algorithms and their importance in various fields.

3.1 Optimization Algorithm [2], [15]

Optimization is all over the place and is, in this manner, a significant paradigm itself with a wide range of uses. In almost every application in engineering & industry, we are always seeking to streamline something - whether to reduce cost and vitality use or improve benefit, production, execution, and effectiveness. Although most real-world applications involve intricate parts and characteristics that impact how the framework operates, the optimal exploitation of available resources of any sort consists of a shift of perspective in logical reasoning.

Computer simulations are heavily used in modern engineering configurations. This complicates the optimization process. The growing interest in accuracy and the ever-expanding multidimensional character of structures and frameworks make the recreating technique more time-consuming. Evaluating a single design might take many weeks or even months in many technical professions. Any solutions that will shorten recreation time & improve the process will therefore save time and money. The coordinated pieces of the streamlining technique for each optimization problem are also the optimization technique, a productive numerical simulator, and a sensitive representation of the physical operations we desire to exhibit and improve. This is generally a time-consuming technique, and computational costs are typically relatively high. When we have a better model, the generic calculation costs are defined by the search optimization techniques and the numerical solver used for simulation.

3.2 Swarm Intelligence [4], [16]

A swarm is a collection of several homogeneous, primary agents that associate individually to themselves & their environment, with no centralization to allow for exciting global evolution. Swarm-based methods have recently emerged as a nature-inspired class, population-based techniques capable of providing simple, rapid, and resilient solutions to a few complicated problems.

Swarm Intelligence may therefore be defined as a relatively new branch of AI that is used to illustrate the social swarm's collective behavior in nature, such as bird flocks, colonies of ant, and honeybees. Whereas these entities are typically unsophisticated with limited capabilities on their own, they link with certain personal behavior rules to enable them achieve essential tasks for their survival. Direct or indirect social relationships might exist among swarm persons.

Swarm Intelligence ideas have been successfully used to a variety of problems, such as function optimization, scheduling, optimum route finding, structure optimization, plus picture and data processing. A swarm is a group of numerous homogenous, primary agents who interact locally with one another & with their surroundings, without centralized control, to produce globally intriguing behavior. Swarm-based methods have lately evolved as a nature-inspired class and population-based approaches capable of delivering simple, quick, & robust solutions to some of those complex issues.

Swarm intelligence is now a new field of artificial intelligence that depicts the collective behavior of social swarms in environment, for instance, bird flocks, ant colonies, and honeybees. Whereas these entities (swarm members or insects) are usually naive with limited capabilities on their own, they are collaborating with personal conduct rules to assist them in achieving essential tasks for their survival. Direct or indirect social relationships might exist among swarm individuals. Swarm Intelligence concepts have been effectively used in a variety of problem fields, like function optimization, scheduling, optimal route finding, structural optimization, & image and data processing.

3.3 Genetic Algorithm [17]

GA is a search & optimization method relying on concepts of natural evolution, which John Holland initially introduced around 1970. Genetic algorithms also realize optimization techniques by reenacting species evolution via common choices. In general, the genetic algorithm has been comprised of two steps. The first procedure would be the choice of the individual for next-generation products, and the 2nd stage is the manipulation of a chosen person to generate the next generation from mutation and crossover processes [3]. The mechanism of selection decided that which individuals have been selected for reproduction & also how many children every selected person produces. A primary guideline of the determination technique is that the better someone is, the greater their chances of becoming a parent are.

GA view the problem area as a individuals' population and aim to find the healthiest person, repeatedly generating ages. GA transforms a populace of newcomers into a populace of experts, with each person speaking to a solution of issue at hand. A fitness function estimates the nature of each benchmark as quantitative expression of every adaptation of standard to a given context. This approach begins with a randomly generated population of individuals. At every age, three primary hereditary administrators, namely selection, crossover, and mutation, are implemented in a sequence to every individual with respective estimations.

The GAs would be a computer software that mimics the evolution and heredity of live creatures. Because GAs are multi-point search techniques, an optimum solution for the multi-modular target capabilities is conceivable in any case. Similarly, GAs is relevant to discontinuous search space concerns. As a result, GA is both easy to use and an exceptionally effective improvement tool. Within GA, search space has been formed of strings, in which every string refers to a competing problem solution and is referred to chromosomes. A fitness value is the target function value of every chromosome's estimate. The population is defined as a collection of chromosomes and their related health. Generations are represented as populations that emerge throughout a GA cycle.

4 METHODOLOGY

The methods used in our research are divided into two sections. The first section provided a brief explanation of Literature Review, and second section provides an overview of the experiment that has been conducted.

4.1 *Literature review of Optimization Algorithms*

In this section, we address all the major existing Optimization algorithms. In this paper, various articles have been considered for the research. These articles consist of different heuristic algorithms and genetic algorithms. The primary interest in reviewing these articles is that they are inspired by nature and the developed optimization algorithm. We have utilized many available digital libraries like Science Direct, Springer, Google Scholar, etc.

4.2 *The Algorithm of Whale Optimization (Seyedali Mirjalili, & Andrew Lewis) [18]*

This is a brand-new meta-heuristic optimization strategy based on nature. This method was inspired by creature known as "humpback whales." This method is motivated by social behavior & lifestyle of the humpback whales. This method is relied on humpback whales' social behavior & hunting habits. So, in this method, the modeled hunting strategy of humpback whales is explored, with the best or random search agent chasing their prey and usage of the spiral to imitate bubble-net attacking technique of humpback whales. These whales like to chase a swarm of fish and krill near to surface, and they do it by blowing characteristic bubbles or following a '9' shaped course, as seen in the image below:

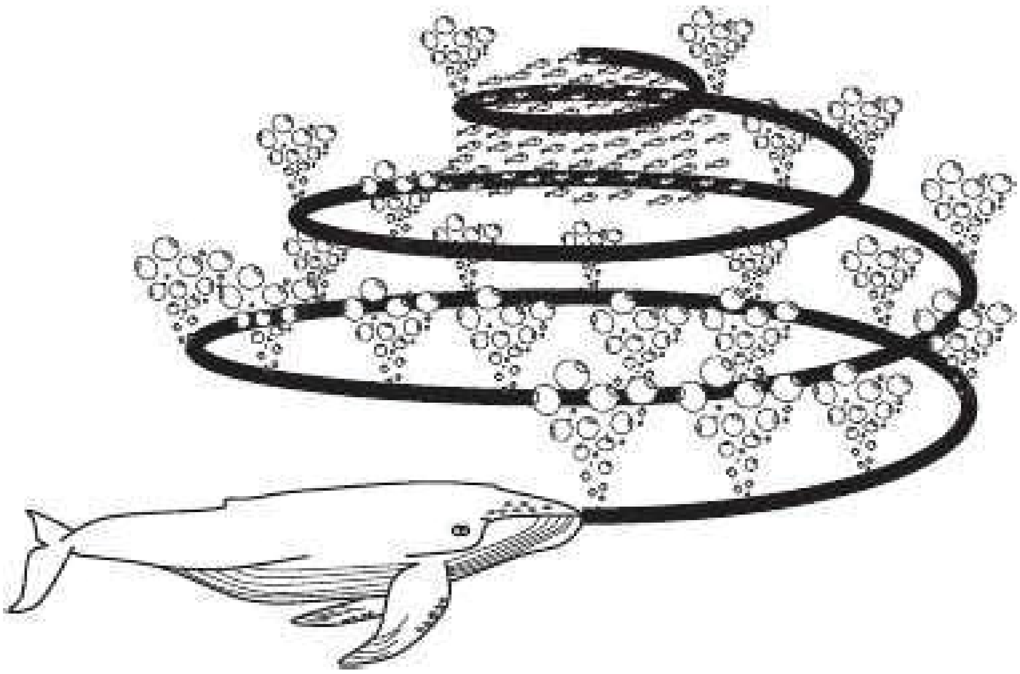

Figure 1: Behavior of Bubble-net feeding of humpback whales (Seyedali Mirjalili, Andrew Lewis) [18]

The surrounding of prey and the spiral bubble net feeding technique may be deduced from the illustration above. The optimal search agent is determined while surrounding the prey, & the other search agents would update their positions appropriately, as shown by the formula below:

$$\vec{D} = |\,\vec{H}\,\vec{X}^{*t} - \vec{X}^{(t)}| \tag{1}$$

$$\vec{X}^{(t+1)} = |\,\vec{X}^{*t} - \vec{G} * \vec{D}\,| \tag{2}$$

Here, the current iteration is denoted by t, coefficient vectors are represented by $\vec{G}$ & $\vec{H}$, X* is showed the best solution's position vector, position vector is declared by X, absolute value is described by | |, & is a multiplication of element − by − element. The bubble net attacking method is given by shrinking encircling technique and is shown in figure given below:

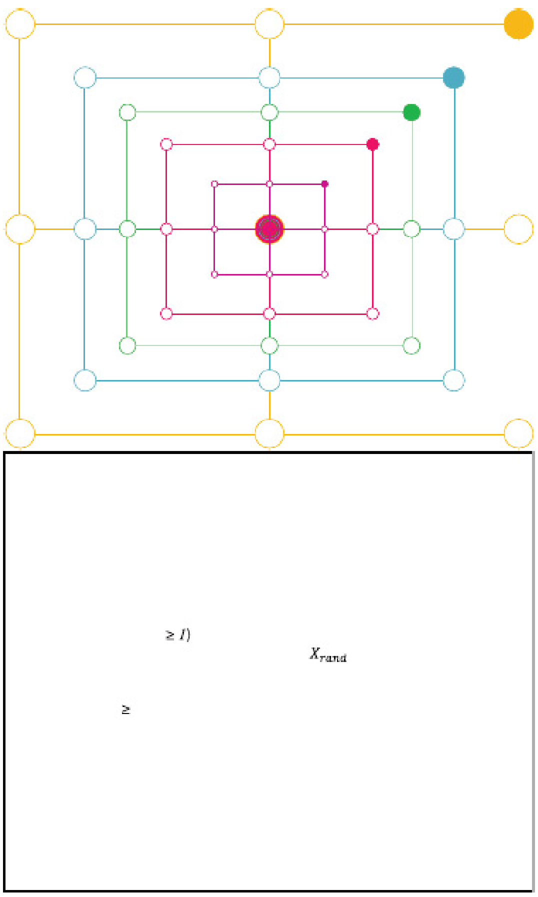

Figure 2: Bubble net attacking method [18]

This is how the exploration mechanism is implemented in the Whale Optimization algorithm.

4.2.1 Cuckoo Optimization Algorithm (Ramin Rajabioun) [19]

This approach is appropriate for continuous nonlinear optimization problems. This algorithm was motivated by a bird known as the "Cuckoo." This is one of the algorithms influenced by nature, and the evolutionary method is used. This algorithm has been motivated by the lifestyle of the cuckoos, which includes characteristics like laying eggs and breeding. The basis of this algorithm is the effort to survive among cuckoos. So, these cuckoos usually search for a safe place to lay their eggs to have more chances of survival. Then, there is an egg-laying radius for these cuckoos where they sbegin to lay eggs within some other nests. The laying of egg procedure is repeated until the optimal location with the highest profit value is found, and the majority of the cuckoo swarm is concentrated around the same spot [14].

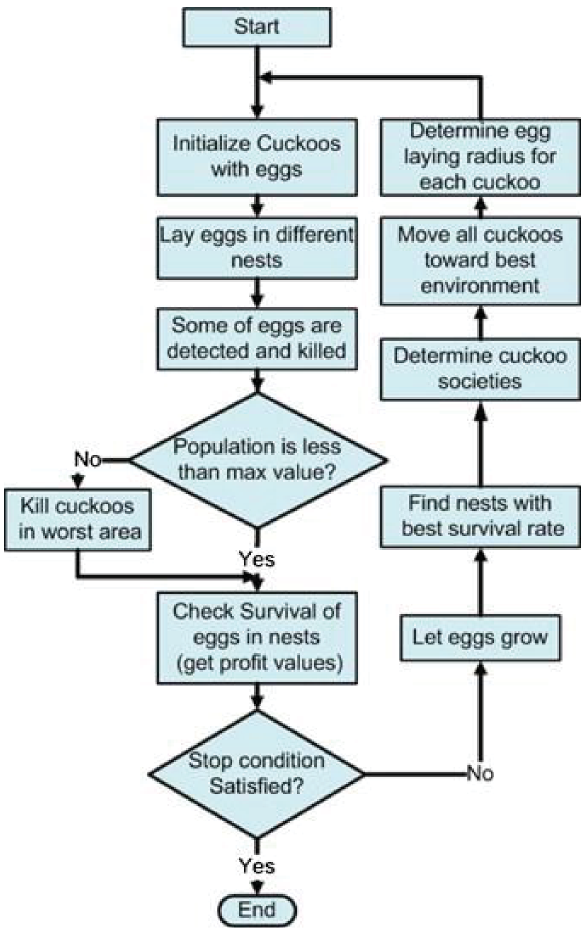

Figure 3: Cuckoo Optimization Algorithm Flowchart (Ramin Rajabioun) [19]

This algorithm is represented in the flowchart mentioned above. It is first seeded with cuckoo eggs, after which eggs are placed in various nests, and some eggs are found and destroyed. If the number is fewer than the maximum amount, the survival of the eggs within nests is tested. If it is no, then cuckoos in the worst area are killed, and then as the population becomes less, the condition is checked if it is satisfied or not. If it is satisfied, then the process is ended. If it is not happy, then another part of the tree is implemented where the eggs are grown, the best survival-rated nests are found. The cuckoo societies are determined, all the cuckoos have been moved to the best environment, the egg-laying radius has been determined, and then the part1 process of the flowchart is followed. This is what happens in the Cuckoo Optimization Algorithm.

The problem values variables are often represented as an array in optimization techniques. In the cuckoo optimization process, this is referred to as "habitat." In a $N_{\overline{var}}$ dimensional optimization process, a habitat array of $1 \times N_{var}$ represents the cuckoo's current dwelling position [14]. The following is the definition of this array:

$$Habitat = [x1, x2, x3, x4, \dots xNvar] \tag{3}$$

Each of the variables is a floating-point number. The profit function needs to be evaluated for the above habitat. It is given by,

$$Profit = fp(habitat) = fp\ (x1, x2, x3, x4, \dots xNvar) \tag{4}$$

In the cost minimizing problems, one can maximize the profit function as,

$$Profit = -cost(habitat) = -fc(\ x1, x2, x3, x4, \dots xNvar) \tag{5}$$

The egg laying radius (ELR) in this algorithm is given by,

$$ELR = \alpha \times \frac{Number\ of\ current\ cuckoo\ eggs}{Total\ number\ of\ eggs} \times (var_{hi} - var_{low}) \tag{6}$$

Where α is an integer,

V_{arlow} is the lower limit of variables

V_{arhi} is a higher limit of variables

4.2.2Algorithm of Forest Optimization (Manizheh Ghaemi Mohammad-Reza FeiziDerakhshi) [20]

The Forest Optimization Algorithm, based on a process employed in forests, is another attempt to address nonlinear optimization challenges. A few trees inside the backwoods because It can live for such a long period, but other trees could only live for a limited period of time Seed dispersion is a typical method that involves the transport of seeds over the whole countryside. The flight of the diaspora is managed via seed distribution (unit of a plant-like origin). When the seeding process begins, a few seeds fall immediately under the parent trees themselves, known as local dispersion (alluded to as local seeding in FOA). Large-distance seed dispersion is characterized by natural mechanisms such as critters and the wind to spread seeds over long distances (alluded to as global seeding in FOA). Likewise, there is always competition amongst surrounding trees to employ the existing principles. The winners are those plants with superior day-by-day environments, while other trees eventually age and perish [20].

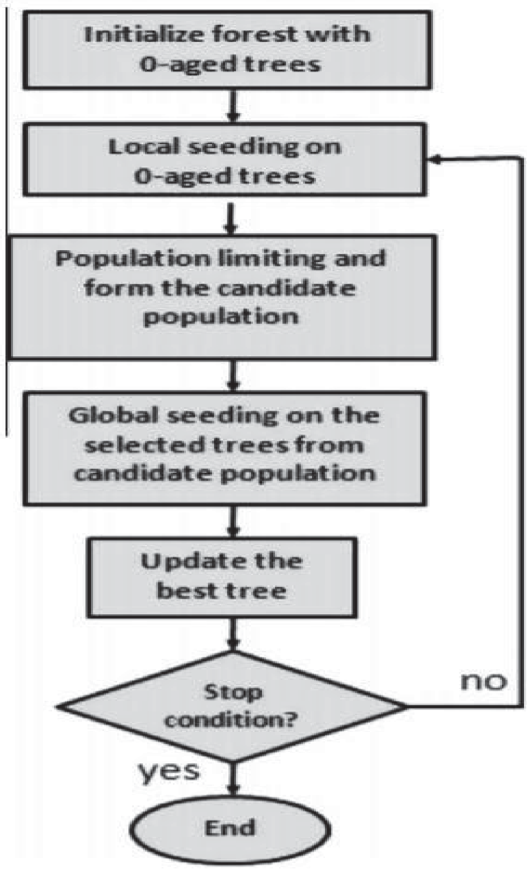

Figure 4: Flowchart of Forest Optimization Algorithm [21]

4.2.3 Prey Predator PSO Algorithm [21]

PSO is a common nature-inspired method of optimization. There have been several densely building "slothful particles" at slower speeds inside the mid-late cycles in the most current PSOs. A unique prey-predator PSO is presented influenced by the prey-predator interaction in nature, using the three capture, breeding, & escape approaches. In Prey Predator-PSO, slothful elements must be removed as well as adjusted, with the prior assisting with raising combination & measurement speed & the latter enhancing optimization efficiency. Whereas everyone in the population has been regarded as the particle representing the possible viable solution, a food place means the universal best solution. Initialization, population assessment, information updating, prey-predator interaction, and population breeding are the five steps of the PP-PSO approach. The flowchart shows the particle's sequence number, & MaxN is the maximum number of iterations.

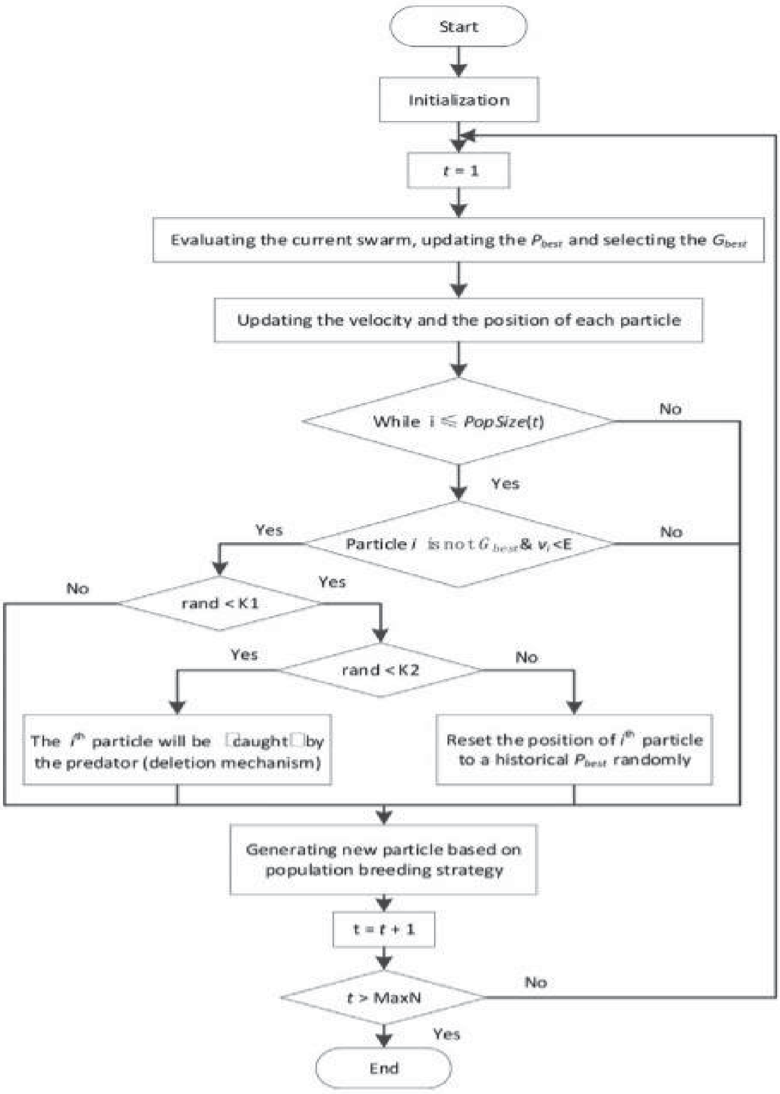

Figure 5: Prey Predator Particle swarm optimization algorithm [21]

Each particle has a designated random beginning place and initial velocity within the flowchart above during the startup step. Every particle's fitness function value is determined. Second, each particle's fitness function value is determined by its Pbest. In PP-PSO, the location of particle I within D dimensional space at Tth is given by assuming the size of the population is PopSize. Slothful particles move slowly and are less committed to seeking a better solution. PI control has grown enormously in popularity because of its stability and effectiveness in maintaining a boundary at a particular value or changing according to a specified rule. The PP-PSO must be dependent on a single population. Within a single swarm, particles interact with each other and gain info. Because of population turnover throughout the optimization phase, the algorithm maintains community variety and executes admirably. This discovery shows that multiswarm techniques may increase PSO performance, and we want to pursue them further in enhancing the PP-PSO.

4.2.4 ACO Algorithm [3]

The various ant species' foraging behavior inspired the ACO. These ants deposit pheromones on the surface to determine how the colony's members should travel. To handle optimization challenges, Ant colony optimization employs a comparison method. A metaheuristic would be a collection of algorithmic notions utilized to describe heuristic techniques essential to a broad range of diverse issues. An appropriate model is required to implement ACO to the specific issue of combinatorial optimization. The acceptable method is necessary to implement ACO to the specific issue of combinatorial optimization. To describe the ACO pheromone model, an optimization problem model is used. The pheromone value has been associated with every solution part, with every potential job of assigning a variable value.

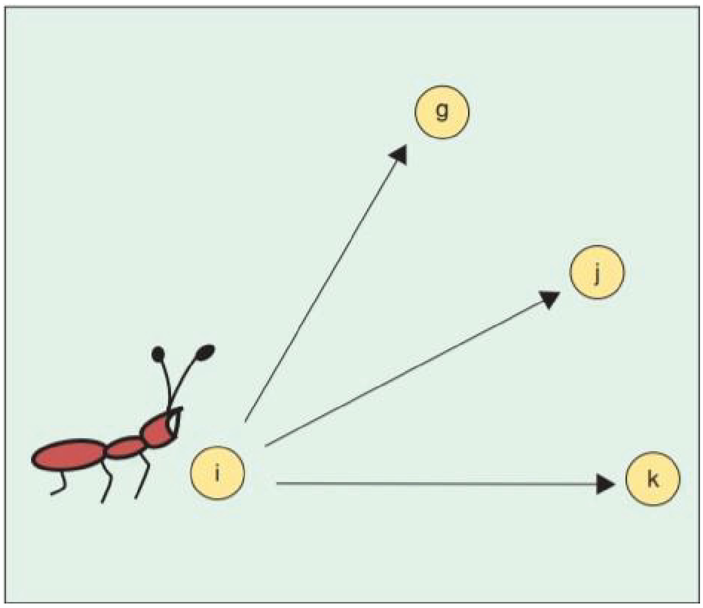

Figure 6: Ant colony optimization algorithm [3]

In the above figure, the ant into the city 'I' decide next city for visiting using the stochastic method.

The algorithm if ACO is as follows: Set parameters, initialize pheromone

while termination condition not met

do ConstructAntSolutions

ApplyLocalSearch(optional) UpdatePheromones

End while

4.2.5 Plant Intelligence Behavior Optimization Algorithm(PIBO)[28]

A lot of study has been done & is still being done to create optimization algorithms based on natural processes. An effort to design an optimization method relying on plants behavior is discussed in this study. Natural processes like how the plants resist competition & alter morphology in reaction to changing in environment make designing a plant intelligence-inspired algorithm [22]. Plant intelligence originates with molecular cell networking. A live creature is formed as a result of molecular interactions. The pattern recognition has been accomplished by assembling neutron-substance [12]. In-plant understanding, proteins are used as analytical components [24]. Plants may scavenge for food by altering their activities. This is accomplished via the evolution of biology, anatomy, & design [25]. Plants make decisions, inserting roots, leaves, and, shoots inappropriate locations or ranks based on the accessibility & evident benefits locations. This study report focuses on plant resistance to harsh conditions, capacity to avoid competition, and ability to perform well in adverse conditions. Using this plant expertise to build an improvement measurement will result in a better streamlining measurement tailored to react to the stated circumstances. Recent technology advancement necessitates an enhancing measure that's flexible and adjustable to diverse scenarios. In contrast to existing nature-driven estimations, the suggested equation is unique in that it may divert time assets to more realistic answers.

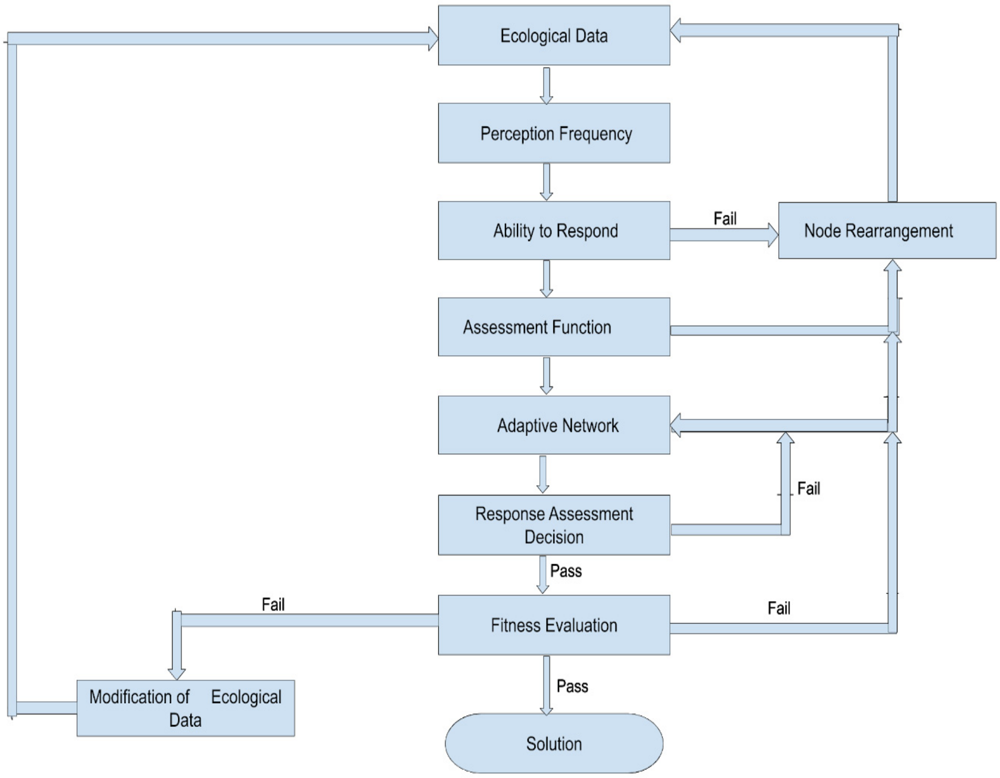

Figure 7: Flowchart of PIBO

Algorithm[28]

1. START
2. Objective Function $O(x_i) = (x_1, x_2, x_3, \ldots x_i)$
3. Generate Population 'N' which is includes x_1 to x_i. Perception Function is to be defined using equation 1.
4. While N < MaxGen do
5. initialize xmin,xmax,fxmin,fxmax
6. get a random node
7. Finding ability to respond using r_i by using equation 2.
8. Find the distance between the node to the stem
9. Assessment function is evaluated based on r_i, by using equation 3
10. Setup the initial adaptive network.
11. **Get** α, β, γ and η.
12. **if** ($\alpha x_i > 0$ and $\beta x_I > 0$ and $\gamma x_I > 0$) and ($\eta x_I > 0$ or $\eta x_i \leq 0$) then
13. find $\max(G(x_i)) = N(C,E)$ and
14. **Choose** $(\alpha, \beta, \gamma) > 0$ yields to max $G(x)$

15. **else if** $\alpha = \beta = \gamma \leq 0$ th_{en}

16. Set G(x) = 0

17. end if

18. Initialize the set Z

19. Find the best current solution O(x).

These are some of the existing optimization algorithms for which the literature review has been done. The literature review is considered in such a way that to understand and get a clear picture of the existing optimization algorithms and to understand what worked for researchers and how research is carried out throughout the paper. The systematic approach that the researcher considers while experimenting is observed, and the idea is formed on how to proceed with this research paper. The main idea behind the literature review is to note what worked for the author, what metrics are considered, and what didn't work so that I would be representing what is necessary for my research instead of overhead and the experiment is done with the utmost care and preciseness.

4.2.6 Selection of Optimization Algorithms

The selection of optimization algorithms is done using the systematic literature review of some of the existing optimization algorithms where its pros and cons are determined. We have decided to use the existing benchmark functions that could tell the efficiency of the algorithms. The metrics which are used to determine the usage of the optimization algorithm are Optimality, CPU Time, Mean and Standard deviation. Some of the benchmark functions that are used to determine the metrics of the algorithms are Rastrigin function(F1)[19], Rosenbrok function(F2)[21], Schwefel function(F3)[23], Ackleyfunction(F4)[24],Griewankfunction(F5)[25],LevyFunction(F6)[26],Sphere Function(F7)[27], Styblinski-Tang Function(F8)[28], Sum Sqaure's Function(F9)[29], Zakharov Function(F10)[30]. Our findings suggest that Optimality is one of the most important factors that could determine the efficiency of the optimization algorithms. These results can be changed if we have chosen the other benchmark functions but we wanted to determine the efficient algorithm by comparing it with the well-known benchmark functions in the optimization space. Upon the systematic literature review these optimization algorithms have their own pros and cons as follows:

Algorithm	Pros	Cons
Particle Swarm Optimization Algorithm[4]	Less CPU Time	More Standard deviation
Whale Optimization Algorithm[18]	Easy to implement	Takes more time than others
Cuckoo Optimization Algorithm[19]	Average CPU Time	Standard deviation is high
Ant Colony Optimization Algorithm[3]	High Accuracy	High CPU Time
Genetic Algorithm[17]	More Optimality when compared to ACO	High CPU Time when compared to PSO and PIBO.
PIBO[28]	Simple to implement, more optimality	Computation time is little more when compared to PSO and ACO.
GrassHopper Optimization Algorithm[1]	Average CPU Time	Low optimality
Strawberry Plant Optimization Algorithm[6]	Average Optimality	Low accuracy
Artificial Plant Optimization Algorithm[14]	Easy to implement	High standard deviation
Flower Pollination Algorithm[13]	Average standard deviation	Low optimality

Table-1: Selection of optimization algorithms

Based on the above systematic literature, we have chosen PSO Algorithm[4], GA[17], ACO[3] and PIBO[28] as the optimization algorithms and the benchmark functions have been used to determine the metrics and the comparative study has been done.

4.2.7 Formulation of Search String for literature review

In literature review, formulation of search string plays a key role in retrieving appropriate and related research papers. The keywords that are used to search should be specific and totally related to the aim of our research. Various existing databases like Google Scholar, Scopus, Science Direct, IEEE Explorer, diva, etc. have been used for searching the related research papers for the systematic literature review. The keywords like “Optimization Algorithm” AND “Nature-based algorithms” are used as the keywords to search the related research papers and filtered thoroughly using Inclusion and Exclusion Criteria. After the results, the research papers are read thoroughly and unrelated papers are exempted from the literature review.

4.2.8 Inclusion and Exclusion Criteria

Inclusion Criteria

- The literature that has been published from the past 15 years will be considered for the systematic literature review.
- Only nature based optimization algorithms are chosen for the research.
- Literatures published in English are considered

Exclusion Criteria

- Only full text published literatures are considered.
- No other format than text is considered where audio etc. are avoided

4.3 Experiment

In this section, the experiment that has been performed is represented.

Part-1: The benchmark function that is used to measure the efficiency is represented

Part-2: The metrics that are used to represent the efficiency data are represented

Null Hypothesis(H_0): No difference in results of the metrics when the benchmark functions are applied on the four algorithms to compare.

Alternate Hypothesis(H_1): there is a difference in the metrics results when the benchmark functions are applied on the four algorithms to compare.

Independent Variables: This contains the PSO algorithm, the Genetic Method, the ACO algorithm, and the PIBO Algorithm

Dependent Variables: This includes the metrics like Optimality, Accuracy, CPU Time, and Mean best standard deviation.

In our experiment, we are comparing our chosen algorithms PSO[4], GA[17], ACO[3] and PIBO[28] with the ten benchmark functions and will determine the efficiency based on metrics like CPU Time, Optimality, Standard Deviation, Mean.

4.3.1 Benchmark Functions:

There are roughly ten benchmark functions in the research paper we wrote in which we compare to various existing algorithms such as PSO, GA, PIBO and ACO techniques. Following is a list of the benchmark functions that are used:

4.3.1.1 Rastrigin Function

Rastrigin function is used in our research paper. The formula gives this

$$f(a) = 10d + \sum_{i=1}^{d}[\{a_1^2 l - 10\cos(2\pi a)\}] \quad (19)$$

$$Subject\ to\ -5.12 \leq x\% \leq 5.12, the\ global\ minima\ is\ f(a) = 0\ at\ a = (0,0,\ldots.,0) \quad (20)$$

4.3.1.2 Rosenbrok Function

The Rosenbrok function is a valley-shaped non-convex and unimodal function implemented as a efficiency test issue for the optimization techniques. The Rosenbrock function may be effectively optimized by modifying the relevant coordinating system without using gradient information or local approximation models. The formula provides this information.

$$f(a) = \sum_{i=1}^{d-1}[(a_{i+1} - a_i^2)^2] \quad (21)$$

$$Subject\ to\ -2.3 \leq x\% \leq 2.3, the\ global\ minima\ is\ f(a) = 0, at\ a = (1,1,1,\ldots 1) \quad (22)$$

4.3.1.3 Schwefel function

In this study, we employ the Schewefel function to assess the efficiency of the suggested approach. This is provided by.

$$f(a) = 418.9829d - \sum_{i=1}^{d} x_i \sin(\sqrt{|a_i|}) \quad (23)$$

Subject to $-500 \leq a\% \leq 500$, the global minima are $f(a) = 0, at\ a = (420.97, \ldots, 420.97)$

4.3.1.4 Ackley function

Ackley function includes many local minimum points and is used in our research paper to calculate the efficiency of the suggested algorithm. This is given by.

$$\boldsymbol{f}(a) = -x\ exp\left(-b\sqrt{1/_{d}}\sum_{i=1}^{d} a_i^2\right) - exp(\tfrac{1}{d}\sum_{i=1}^{d} \cos(ca_i)) + x\ exp\,(1) \qquad (24)$$

Subject to $-35 \le x\% \le 35$, the $f(x) = 0\ at\ x = (0, 0, \ldots 0)$ is global minima, $b = 0.2$, $x = 20, c = 2\pi$

4.3.1.5 Griewank function

The Griewank function includes many local minimum points, and it is used in our research paper to calculate the efficiency of the suggested algorithm. This is given by;

$$\boldsymbol{f(a) = \sum_{i=1}^{d} \frac{a_i}{4000} - \prod_{i=1}^{d} \cos(\frac{a_i}{\sqrt{i}}) + 1} \qquad (25)$$

Subject to -100 ≤ $x\%$ ≤ 100 and f(x) = 0 at x=(0,0,…0) is global minima.

4.3.1.6 Levy function

The Levy function is used in our research paper to calculate the efficiency of the suggested algorithm. This is given by;

$$f(a) = sin^2(\pi w_1) + \sum_{i=1}^{d-1}(w_i - 1)^2[1 + 10sin^2(\pi w_1 + 1)] + (w_d - 1)^2[1 + sin^2(2\pi w_d)]$$
(26)

Where $w\_i = 1 + \ x\_(i-1)/4, i = 1, 2, \ldots .d$ for all

Subjected $to\ -10 \le a\$ \le 10$ and $f(a) = 0$ at $a = (1,1, \ldots 1)$ is a global minimum.

4.3.1.7 Sphere function

The sphere function would be unimodal with the local minimum point size utilized in our research study to evaluate the suggested algorithm's performance. This is provided by**;**

$$f(a) = \sum_{i=1}^{d} a_i^2 \qquad (27)$$

Subject to $-5.12 \le a\% \le 5.12$ and $f(a) = 0\ at\ a = (0,0, \ldots 0)$ is global minima.

4.3.1.8 Styblinski-Tang function

Styblinski-Tang function is used in our research paper to calculate the efficiency of the suggested algorithm. This is given by

$$f(a) = \frac{1}{2}\sum_{i=1}^{d}(a_i^4 - 16a_1^2 + 5a_i) \qquad (28)$$

Subject to $-5 \leq a\% \leq 5$ and $f(a) = -39.16$ at $a = (-2.9, -2.9, \ldots - 2.9)$ is global minima

4.3.1.9 Sum square's function

Sum squares would be a bowl-shaped function with just one global minimum point utilized in our research study to evaluate the efficiency of the suggested method. This is provided by

$$f(a) == \sum_{i=1}^{d} a_i^2 \tag{29}$$

Subject to $-10 \leq a\% \leq 10$ and $f(a) = 0$ at $a = (0, 0, \ldots 0)$ is global minima.

4.3.1.10 Zakharov function

Except for a global minimum, the function of Zakharov has almost no local minima and is employed in our research work to assess the performance of the suggested method. This is provided by

$$f(a) = \sum_{i=1}^{d} a_i^2 + \left(\frac{1}{2}\sum_{i=1}^{d} i a_i\right)^2 + \left(\frac{1}{2}\sum_{i=1}^{d} i a_i\right)^4 \tag{30}$$

Subject to $-5 \leq a_{\%} \leq 10$ and $f(a) = 0$ at $a = (0,0, \ldots 0)$ is global minima

Three additional algorithms, including the PSO [4], GA [3,] and ACO algorithm [3], are employed to measure the efficiency of the suggested innovative technique with current algorithms. A short discussion of these algorithms may be found towards the end of the literature review.

4.3.1.10.1 Metrics

Benchmark functions and metrics evaluate heuristic algorithms' performance in most cases. Optimality, CPU Time, Accuracy, and Best Mean Time are the measures we use in our experiment. As a result, our approach has been compared to existing methods like the PSO, the GA, and the ACO.

4.3.1.10.2 Optimality

This optimality is given by

$$1 - \frac{\|f_0 - \widetilde{f_0}\|}{\|\bar{f} - \underline{f}\|} \in [0,1] \tag{31}$$

4.3.1.10.3 Accuracy

Accuracy is given by

$$1 - \frac{\|x_0 - \breve{x}_0\|}{\|\bar{f} - \underline{f}\|} \in [0,1] \tag{32}$$

4.3.1.10.4 Mean

Mean is given by

$$\frac{1}{N}\sum_{i=0}^{n} \breve{f} \tag{32}$$

4.3.1.10.5 Standard Deviation

Standard deviation is given by

$$\sqrt{\frac{1}{N-1}\sum(\breve{f}_0 - Mean)^2} \tag{33}$$

Here, $\bar{f}$ $and\ \underline{f}$ are depicted as the upper & lower bound of the function f. $\bar{x}\ and\ \underline{x}$ are defined as the upper & lower bound of 'x,' a search space. And 'N' represents the number of repetitions.

Optimality measures how near a solution is to the fitness function. The closer you get to the answer, the more accurate you have to be with your guess. It's the mean that tells us how near we are to finding a solution on average.

4.4 Experiment Setup

Specifications

- **CPU:** n1-standard-8 VCPUs, 16GB RAM(google cloud)
- **GPU:** NVIDIA T-4, 16GB DDR4(google cloud)
- **OS:** Windows 10
- **RAM:** 16 GB

All the algorithms are thoroughly tested using the 10 different benchmark functions and based on the metrics of Optimality, Accuracy, Mean and Standard Deviation the efficiency of the algorithm is determined in the results. Each algorithm has been run for about 1000 iterations with each benchmark function and the last iteration is considered as the result due to its accuracy. Some algorithms performed better in some benchmark functions and the others excelled in the others. All the results have been tabulated and graphical representation is done based on the given metrics.

5 RESULTS

Heuristic methods are often tested using benchmark functions, and their performance is assessed using metrics. In our research paper existing techniques like PSO, GA, ACO, and PIBO has been compared with the standard benchmark functions such as function Rastrigin function(F1)[19], Rosenbrok function(F2)[21], Schwefel function(F3)[23], Ackley function(F4)[24], Griewank function(F5)[25],LevyFunction(F6)[26],SphereFunction(F7)[27], Styblinski-Tang Function(F8)[28], Sum Sqaure's Function(F9)[29], Zakharov Function(F10)[30]. Mean Best (Std), CPU Time, Optimality, & Accuracy are the metrics employed to calculate & compare the methods' efficiency. These threshold functions have been employed within the algorithms above to test and compare their efficiency using the metrics mentioned above, with the results shown below:

Table 1. Measurements/Algorithms/Benchmark function

Measure	**Algo**	**F1**	**F2**	**F3**	**F4**	**F5**	**F6**	**F7**	**F8**	**F9**	**F10**
Mean Best (STD)	PSO	1.48E+01	1.08E+01	1.68E+03	4.21E+00	5.13E01	2.22E01	1.13E01	-3.38E+01	2.26E+00	6.86E +00
	GA	2.30E+01	5.92E+01	8.75E+02	1.57E+01	1.15E+00	2.34E+00	1.90E+00	-3.53E+01	3.39E+01	6.70E +01
	ACO	4.75E+01	2.93E+01	1.13E+03	1.51E+01	1.25E+00	1.38E+00	2.29E01	-3.01E+01	2.28E+00	2.01E +01
	PIBO	0.00E+00	6.81E05	4.13E01	0.00E+00	0.00E+00	5.04E06	1.93E35	-3.92E+01	0.00E+00	6.92E -14
CPU time	PSO	23,770.00	20,678.00	28,058.00	23,824.00	22,404.00	18,928.00	16,162.00	21,188	20,954.00	30,65 4.00
	GA	88,000.00	67,500.00	89,500.00	95,320.00	74,420.00	71,500.00	63,320.00	76,830.00	82,420.00	76,23 0.00
	ACO	100,000.00	100,000.00	100,000.00	100,000.00	100,000.00	100,000.00	100,000.00	72,678.00	100,000.00	96,77 4.00
	PIBO	74,482.00	88,128.00	97,900.00	96,158.00	76,824.00	72,242.00	56,814.00	78,756.00	82,386	77,66 6.00
Optimality	PSO	0.817	0.998	0.005	0.811	0.922	0.998	0.998	0.967	0.992	1
	GA	0.715	0.99	0.478	0.296	0.827	0.975	0.964	0.977	0.887	0.999
	ACO	0.411	0.995	0.327	0.324	0.812	0.986	0.996	0.945	0.992	1
	PIBO	0.923	0.994	0.573	0.697	0.934	0.991	0.983	0.958	0.898	0.997
Accuracy	PSO	0.932	0.797	0.671	0.989	0.973	0.989	0.992	0.867	0.99	0.961

	GA	0.907	0.84	0.705	0.916	0.963	0.955	0.969	0.907	0.967	0.861
	ACO	0.888	0.809	0.777	0.933	0.959	0.966	0.995	0.824	0.994	0.925
	PIBO	0.984	0.873	0.738	0.988	0.966	0.982	0.984	0.938	0.989	0.938

These are the results obtained. From these results, we can further deduce that when compared with all the other algorithms, PIBO gives the best result in mean best standard deviation, which showcases a low deviation from the error value. When optimality is considered, PIBO performs much optimal and outruns others in benchmark functions F1, F2, F3, F4, F5, F10, and other functions; it is almost as better as most of them are a good sign that this proposed algorithm gives a better result. On the other hand, when accuracy is taken into consideration, it outperforms different algorithms in the functions F1, F2, F4, F5, F8, F10, and in other functions, it is almost as close to the best one, which also proves that PIBO is good at giving accurate results. Lastly, when CPU Time is considered, it is not performing as well as PSO and GA but better than ACO; this can be viewed as a drawback, but while taking other positive results into account, CPU Time is a bit considerable.

It has been observed that as the iterations numbers of the algorithm increase, the value it outputs initially varies. Still, as it is run for 1000 iterations, the final result doesn't vary much and is almost the same as the results of the last few iterations in the total 1000 iterations. So, the values represented below are the values that are the result of the 1000^{th} iteration. So, it can be said that the optimality or the accuracy that is represented is achieved at the 1000^{th} iteration. This is considered to avoid confusion with varying outputs of the algorithms when tested with the considered benchmark functions.

The graphs of the individual metrics are represented below:

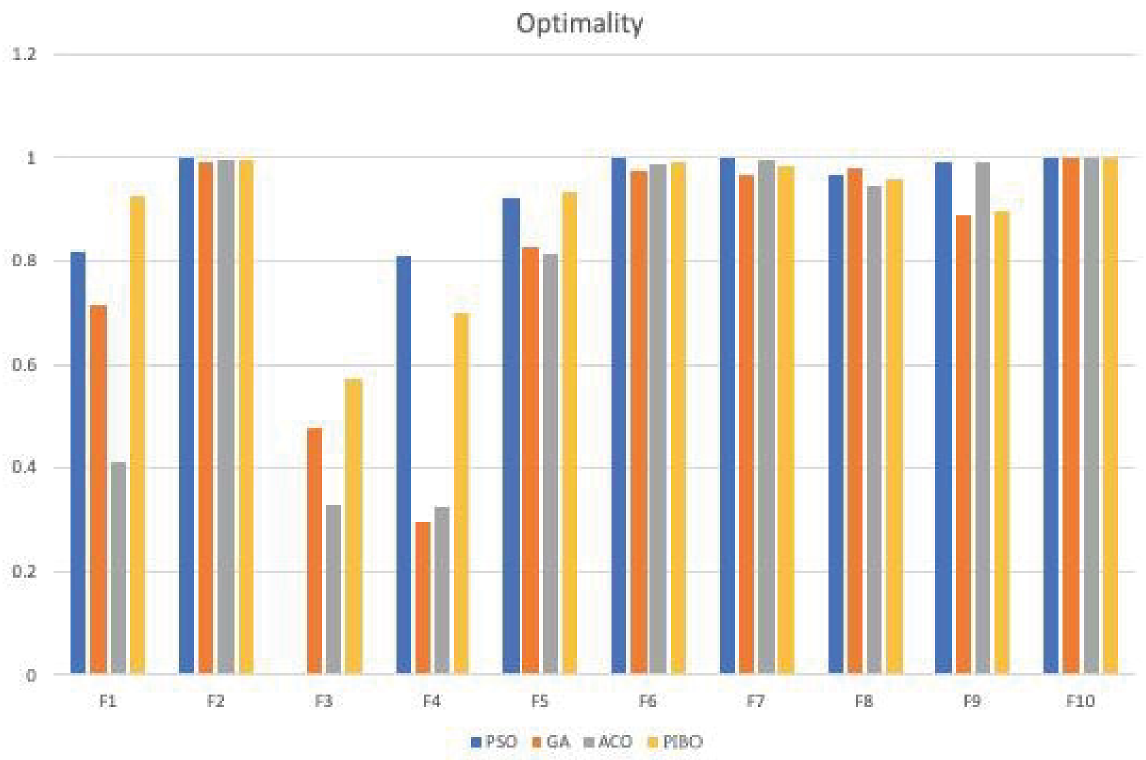

Figure 7: Optimality of Algorithms/benchmark functions

Optimality determines the relative closeness to the fitness of the solution. The figure clearly shows that it outperforms other algorithms in the benchmark functions F1, F2, F3, F5, F10 and so close to the other algorithms in the rest of the benchmark functions. As aforementioned, these are the results of the final iteration where the solution is almost the same as the last few iterations and is found to be much optimal when compared to the initial iterations when the algorithm is run. FIBO and PSO clearly depicts that they have more optimality when compared to other algorithms.

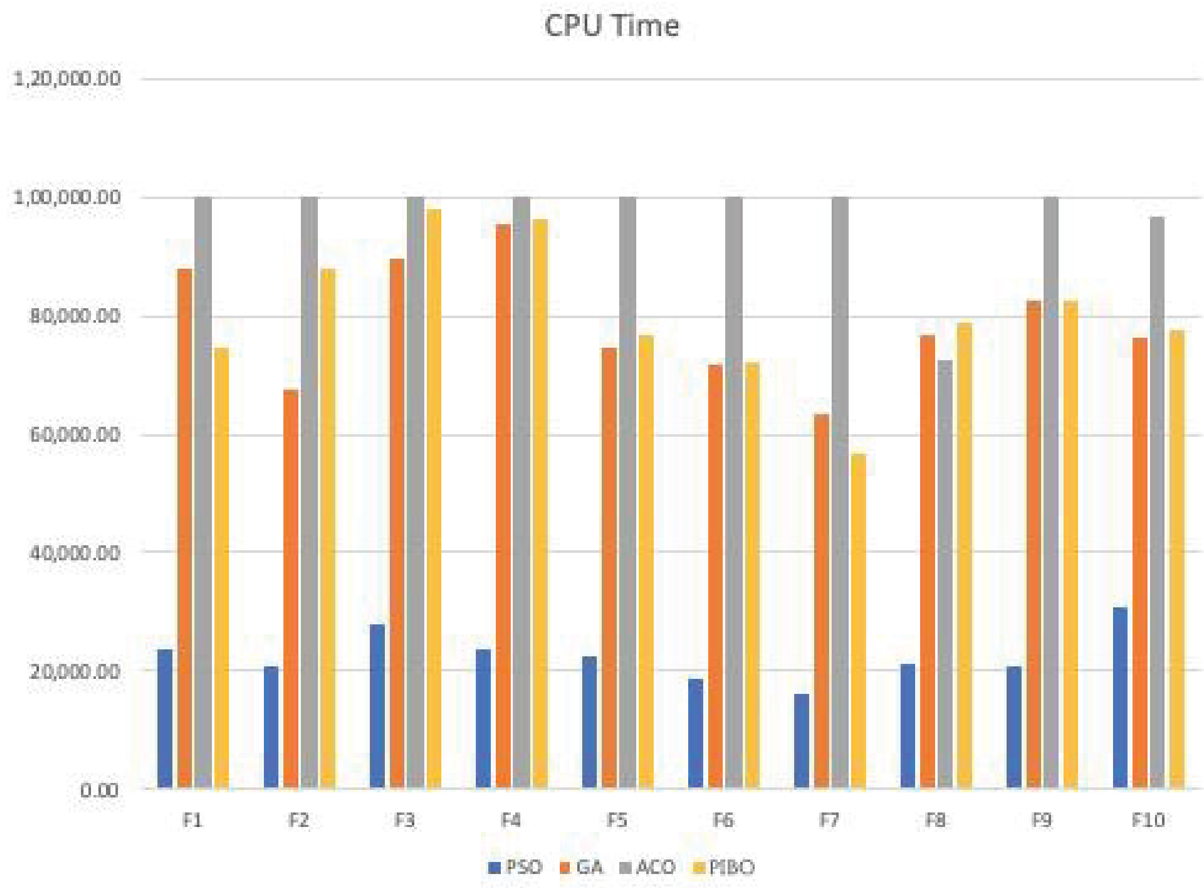

Figure 8: CPU Time of Algorithms/benchmark functions

CPU Time determines the total time it takes to run the algorithm's iterations using the benchmark functions. From the results obtained, it can be depicted that CPU Time is the major drawback for the algorithm. The other algorithms, like PSO, GA outperform the PIBO, but PIBO performs much better when compared to ACO algorithm and is a bit closer to GA. In F1, F6, and F7, it performs better than GA, while GA runs in other functions. Though it is a drawback, it can be considered when the other metrics are considered for efficiency of result. PSO clearly outperforms all the existing algorithms in CPU Time.

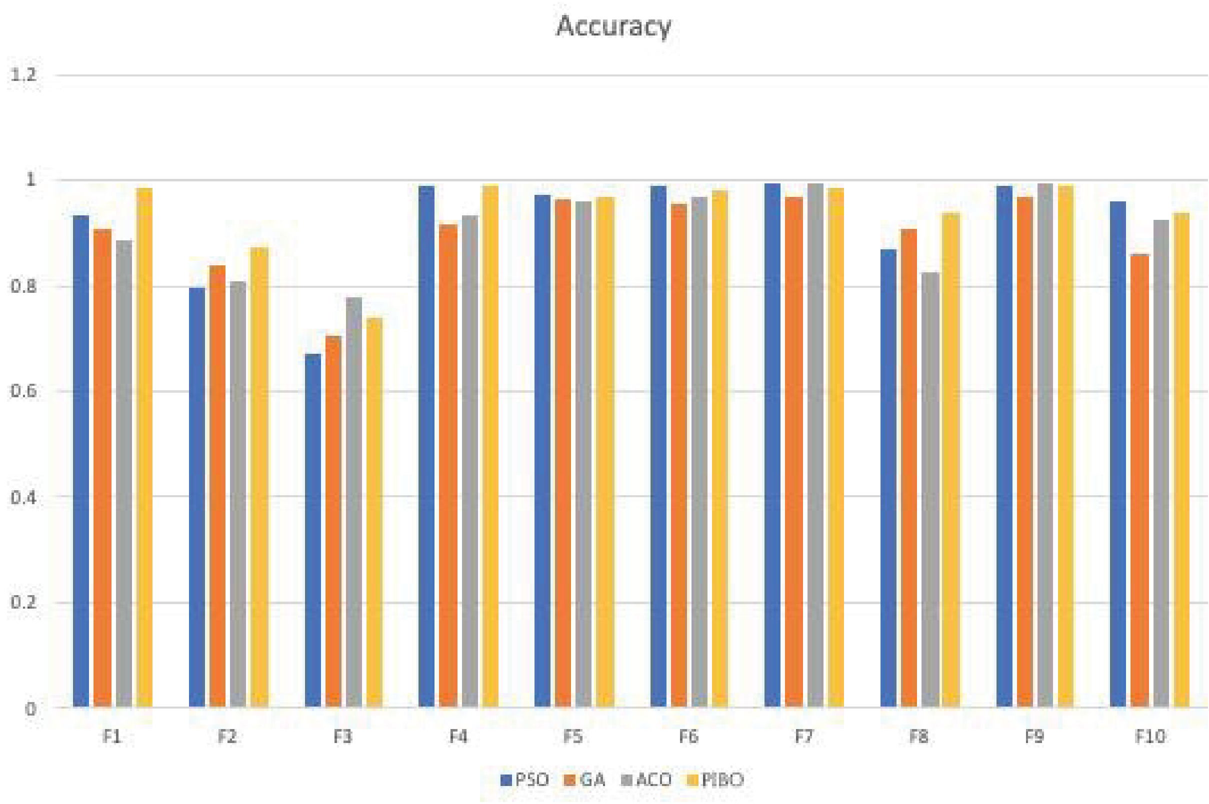

Figure 9: Accuracy of Algorithms/benchmark function

Accuracy determines the relative closeness to the solution. The results claimed that the suggested method PIBO outperformed existing algorithms in benchmark functions F1, F2, F8, F9 and is almost as close to the top-performing algorithm in the other benchmark functions. On the other hand, PSO is outperforming other existing algorithms.

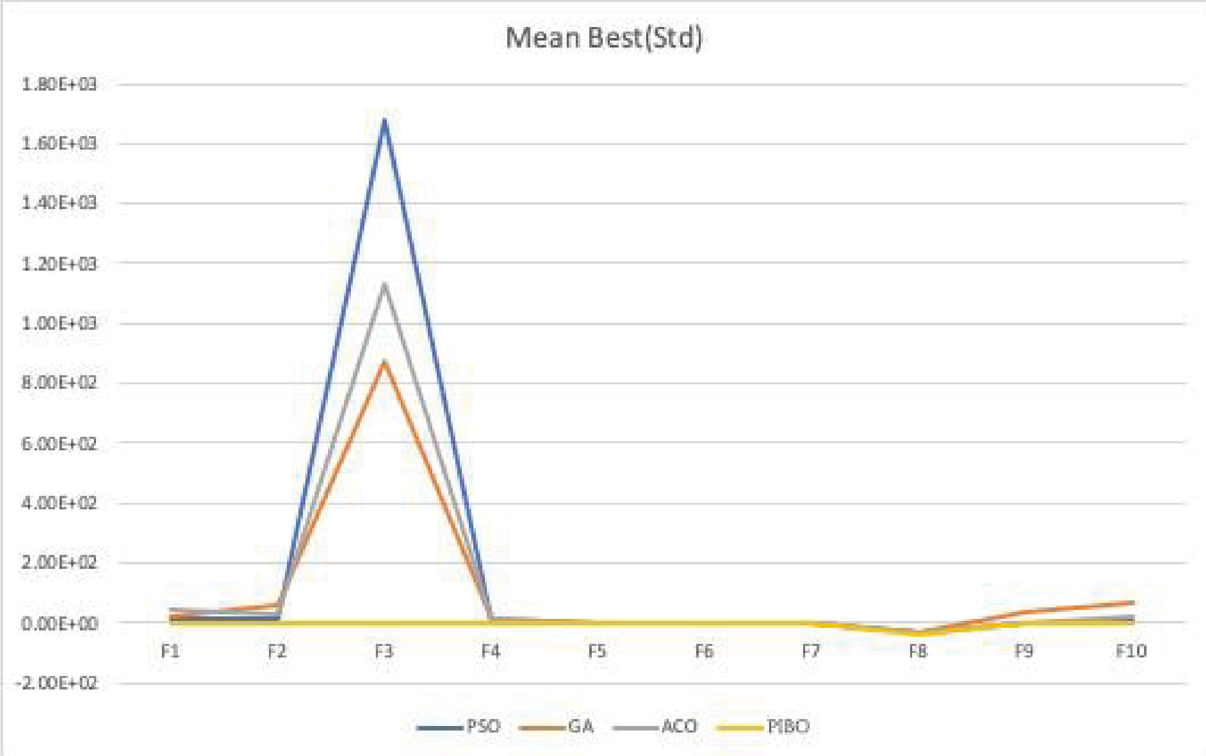

Figure 10: Mean best standard deviation of Algorithms/benchmark functions

Mean Best Standard deviation determines the average of the closeness to the solution or the average of the farness from the error of the solution. The figure represented above clearly demonstrated that the presented PIBO performed better than the other functions into the almost all benchmark functions. It outperforms the other algorithms in the functions F3, F9, and F10. This is a good sign as the mean square error is too low and gives the best results on more iterations.

6 ANALYSIS AND DISCUSSION

In this section, we mainly emphasize the research questions we previously mentioned in our research paper.

1. What are the optimization algorithms available in the current literature?

The Optimization algorithms that are chosen are nature-based optimization algorithms. In our literature review and the related work some algorithms are represented like Whale Optimization method based on humpback whales, their nature, and their motions. The cuckoo optimization technique depends on the cuckoo bird, its movement, & its nature. The ACO method is relied on the ants, biology, and mobility. Similarly, in a revolutionary evolutionary technique, an algorithm of artificial plant optimization is motivated by the tree's development process. Bayat et al. [6] introduced a novel numerical optimization approach for handling complex engineering issues inspired by the strawberry plant. The Forest Optimization Algorithm[20], based on a process employed in forests, is another attempt to address nonlinear optimization challenges. PSO[21] is a common nature-inspired method of optimization. There have been several densely building "slothful particles" at slower speeds inside the mid-late cycles in the most current PSOs. A unique prey-predator PSO is presented influenced by the prey-predator interaction in nature, using the three capture, breeding, & escape approaches. PIBO[28] is a an effort to design an optimization method relying on plants behavior is discussed in this study. Natural processes like how the plants resist competition & alter morphology in reaction to changing in environment make designing a plant intelligence-inspired algorithm [22]. Strawberry plant optimization algorithm[6] has been introduced as a numerical based optimization algorithm based on the strawberry plant. Flower Pollination Algorithm[13] has been introduced to solve the global optimization problems. Artificial Plant optimization algorithm[14] has been introduced in the swarm intelligence and bio-inspired computation.

These are some of the optimization algorithms that are used in the current literature and the systematic review has been done and further 4 algorithms have chosen for the comparative study based on the experiment done using the 10 benchmark functions resulting in the required metrics.

2. **Which optimization algorithm gives optimal performance in terms of CPU runtime?**

From the literature review we have done, we have chosen four different optimization algorithms for our experiment. They are PSO[4], GA[17], ACO[3] and PIBO[28]. These optimization algorithms are run against 10 different benchmark functions Rastrigin function(F1)[19], Rosenbrok function(F2)[21], Schwefel function(F3)[23], Ackley function(F4)[24], Griewank function(F5)[25],Levy Function(F6)[26],Sphere Function(F7)[27], Styblinski-Tang Function(F8)[28], Sum Sqaure's Function(F9)[29], Zakharov Function(F10)[30]. There are four metrics that are considered to measure the efficiency of the optimization algorithms. They are Mean Best (Std), CPU Time, Optimality, & Accuracy. Based on the metrics and functions, each algorithm has performed different in each benchmark function. CPU Time is considered as the efficient way to measure the optimal performance of the algorithms. In CPU Time, PSO outperforms again as it is having lowest CPU Time and ACO is not efficient when CPU Time is compared with other algorithms. PSO has the highest Mean standard deviation whereas FIBO has the lowest. As per our research, PSO is the efficient algorithm in terms of CPU Time.

6.1 Threats to Validity

- Because particular algorithms like PSO, GA, ACO, and PIBO techniques are utilized in the study paper to assess the algorithm's effectiveness. Each algorithm is checked against benchmark functions. If the criteria is changed then the efficiency would vary.
- There are specific 10 different benchmark functions considered which are applied to the algorithms to test the algorithm's efficiency. If different benchmark functions are used, then the result might vary.
- The constraints are fixed for the benchmark functions used, so if there are different benchmark functions, the constraints will be changed, and results will vary.

7 CONCLUSION & FUTURE WORK

In this section, we provide findings of our study and potential future directions.

7.1 Conclusion

In this study, there are four different optimization algorithms are chosen. A systematic literature review is done and then an experiment to showcase the efficiency of the algorithm against benchmark functions is performed. These algorithms are run through 10 distinct benchmark functions and algorithm's efficiency is assessed using accuracy, optimality, mean best standard deviation, & CPU time. When all four algorithms are considered then PSO methodology performs much better and can be illustrated by the results of the metrics of optimality, accuracy, mean best standard deviation, and on the other hand other algorithms performed better than PSO when certain benchmark functions are tested

7.2 Future work

The systematic literature review has been done and comparative study is performed through the experiment. This experiment has showcased that the algorithms that have chosen are nature-based optimization algorithms where it created a scope to conduct further experiments using the various datasets. This study is also helpful in trying out various other existing benchmark functions to check the efficiency and performance of the optimization algorithms.

Future studies could focus on using other benchmark functions on the optimization algorithms and check its efficiency based on the metrics. Also different types of metrics can be considered to see the varying result in optimality and efficiency based on CPU runtime.

8 REFERENCES

[1] S. Zahra Mirjalili *et al.*, "Grasshopper optimization algorithm for multi-objective optimization problems," *Springer*, vol. 48, no. 4, pp. 805–820, Apr. 2018, doi: 10.1007/s10489-017-1019-8.

[2] X. S. Yang, "Optimization and Metaheuristic Algorithms in Engineering," in *Metaheuristics in Water, Geotechnical and Transport Engineering*, 2013, pp. 1–23.

[3] M. Dorigo, M. Birattari, … T. S. intelligence, and undefined 2006, "Ant colony optimization," *ieeexplore.ieee.org*, pp. 28–39, Nov. 2006.

[4] J. Kennedy, R. E.-P. of ICNN'95-international, and undefined 1995, "Particle swarm optimization," *ieeexplore.ieee.org*.

[5] S. M. Mahmoudi, M. Aghaie, M. Bahonar, and N. Poursalehi, "A novel optimization method, Gravitational Search Algorithm (GSA), for PWR core optimization," *Ann. Nucl. Energy*, vol. 95, pp. 23–34, 2016, doi: 10.1016/j.anucene.2016.04.035.

[6] F. Merrikh-Bayat, "A Numerical Optimization Algorithm Inspired by the Strawberry Plant," Jul. 2014.

[7] J. Knowles and D. Corne, "The Pareto archived evolution strategy: A new baseline algorithm for Pareto multiobjective optimisation," in *Proceedings of the 1999 Congress on Evolutionary Computation, CEC 1999*, 1999, vol. 1, pp. 98–105, doi: 10.1109/CEC.1999.781913.

[8] K. Deb, A. Pratap, … S. A.-I. transactions on, and undefined 2002, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *ieeexplore.ieee.org*, vol. 6, no. 2, 2002.

[9] U. Boryczka and J. Kozak, "Ant colony decision trees - A new method for constructing decision trees based on ant colony optimization," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2010, vol. 6421 LNAI, no. PART 1, pp. 373–382, doi: 10.1007/978-3-642-16693-8_39.

[10] S. Paramasivam, S. Athappan, E. D. Natrajan, and M. Shanmugam, “Optimization of Thermal Aware VLSI Non-Slicing Floorplanning Using Hybrid Particle Swarm Optimization Algorithm-Harmony Search Algorithm,” *Circuits Syst.*, vol. 07, no. 05, pp. 562–573, 2016, doi: 10.4236/cs.2016.75048.

[11] D. H. Wolpert and W. G. Macready, “No free lunch theorems for optimization,” *IEEE Trans. Evol. Comput.*, vol. 1, no. 1, pp. 67–82, 1997, doi: 10.1109/4235.585893.

[12] A. Hjelmfelt, J. R. the N. A. of Sciences, and undefined 1992, “Chemical implementation and thermodynamics of collective neural networks,” *Natl. Acad Sci.*, vol. 89, pp. 388–391, 1992.

[13] X. S. Yang, “Flower pollination algorithm for global optimization,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 7445 LNCS, pp. 240–249, 2012, doi: 10.1007/978-3-642-32894-7_27.

[14] Z. Cui and X. Cai, “Artificial Plant Optimization Algorithm,” in *Swarm Intelligence and Bio-Inspired Computation*, 2013, pp. 351–365.

[15] C. R. . Gaidhani, Vedashree M. Deshpande, and Vrushali N. Bora., “"Image Steganography for Message Hiding Using Genetic... - Google Scholar,” vol. 3, Mar. 2014.

[16] H. R. Ahmed and J. I. Glasgow, “The Agile particle swarm optimizer applied to proteomic pattern matching and discovery,” *Soft Comput.*, vol. 20, no. 12, pp. 4791–4811, Dec. 2016, doi: 10.1007/S00500-015-1769-3.

[17] N. M. Razali and J. Geraghty, “Genetic algorithm performance with different selection strategiesin solving TSP,” in *Proceedings of the World Congress on Engineering 2011, WCE 2011*, 2011, vol. 2, pp. 1134–1139.

[18] S. Mirjalili and A. Lewis, “The Whale Optimization Algorithm,” *Adv. Eng. Softw.*, vol. 95, pp. 51–67, 2016, doi: 10.1016/j.advengsoft.2016.01.008.

[19] R. R.-A. soft computing and undefined 2011, “Cuckoo optimization algorithm,” *Elsevier*, pp. 5508–5518, Dec. 2011.

[20] F. Azam, M. Sharif, M. Yasmin, S. M.-S. Int, and undefined 2014, “Artificial intelligence

based techniques for earthquake prediction: a review," *Citeseer*, vol. 26, no. 4, pp. 1495–1502, 2014.

[21] H. Zhang, M. Yuan, Y. Liang, Q. L.-A. S. Computing, and undefined 2018, "A novel particle swarm optimization based on prey–predator relationship," *Elsevier*.

[22] G. Chagwiza, "Parasitic Optimisation Algorithm: a Lesson from Striga Asiatica (L) Kuntze Weed," *Process Integr. Optim. Sustain.*, vol. 3, no. 2, pp. 213–226, Jun. 2019, doi: 10.1007/S41660-018-0064-Z.

[23] A. Arkin and J. Ross, "Computational functions in biochemical reaction networks," *Biophys. J.*, vol. 67, no. 2, pp. 560–578, 1994, doi: 10.1016/S0006-3495(94)80516-8.

[24] D. Bray, "Protein molecules as computational elements in living cells," *Nature*, vol. 376, no. 6538, pp. 307–312, 1995, doi: 10.1038/376307a0.

[25] J. P. Grime and J. M. L. Mackey, "The role of plasticity in resource capture by plants," *Evol. Ecol.*, vol. 16, no. 3, pp. 299–307, 2002, doi: 10.1023/A:1019640813676.

[26] A. Dutta, "A Class-book of botany 16th ed.," 1980.

[27] S. Isnard and W. K. Silk, "Moving with climbing plants from Charles Darwin's time into the 21st century," *Am. J. Bot.*, vol. 96, no. 7, pp. 1205–1221, Jul. 2009, doi: 10.3732/AJB.0900045.

[28] Godfrey Chagwiza, "A New Plant Intelligent Behaviour Optimisation Algorithm for Solving Vehicle RoutingProblem", *MathematicalProblemsinEngineering*, vol. 2018, Article ID 9874356, 10 pages, 2018. https://doi.org/10.1155/2018/9874356