Sigma J Eng Nat Sci, Vol. 42, No. 3, pp. 919–944, June, 2024

Sigma Journal of Engineering and Natural Sciences
Web page info: https://sigma.yildiz.edu.tr
DOI: 10.14744/sigma.2024.00000

Review Article

A systematic comparative study of popular biomimetic intelligence techniques

SHIVANI[1], Satinder Bal GUPTA[1,*]

[1]*Department of Computer Science & Engineering, Indira Gandhi University, Meerpur, Rewari, Haryana, 122502, India*

ARTICLE INFO

Article history
Received: 21 October 2022
Revised: 17 December 2022
Accepted: 16 January 2023

Keywords:
Biomimetics; Biomimetics Intelligence (BI); Evolutionary Algorithms (EA); Neural Network; Swarm Intelligence (SI)

ABSTRACT

Biomimetics is an emerging field that allows mimicry of living organisms in nature to develop different techniques so as to solve hard and complex problems related to optimization. The different techniques developed in this field takes inspiration from biology or nature. Biology acts as a powerful tool for imitating, copying, learning, understanding and inspiring the development of new systems and models. The different techniques discussed in this paper include techniques based on evolutionary algorithms, neural network and swarm intelligence. All these techniques are biologically inspired and provide good accuracy. The accuracy of all these algorithms can be increased by using them in hybrid form with other techniques and using different datasets. The comparative analysis of these techniques is done using advantages, disadvantages and applications of these techniques.

Cite this article as: Shivani, Gupta SB. A systematic comparative study of popular biomimetic intelligence techniques. Sigma J Eng Nat Sci 2024;42(3):919–944.

INTRODUCTION

Biomimetics refers to the process of designing models and systems by getting inspiration from the nature. So, it is also called as biomimicry, bionics, biognosis or bioinspiration. The word Biomimetics comprises two Ancient words of Greek origin – "Bios" which refers to life and "mimesis" which refers to imitate. This field of science comprises imitating and learning from nature and life [1,2]. It can be considered as a new science that is based on developing systems by getting inspiration from nature. The different scientific approaches used by humans help them to understand the principles associated with the novel devices which in turn2 help to improve the capability of the devices. Biomimetic Intelligence (BI) techniques are used to develop systems for solving complex problems. These techniques help to develop algorithms based on biological working mechanism [3,4]. Biology acts as a powerful tool for imitating, copying, learning, understanding and inspiring the development of new systems and models. Some of the examples include: (i) the design and function of fins which is inspired from the legs of creatures such as seal, goose that live in water (ii) the birds give inspiration for flying (iii) the development of tweezers and tong was inspired by the beak of birds. In biomimetic intelligence, the word intelligence describes emulating the biological and nature's capabilities for the purpose of decision making. The different techniques used are based on evolutionary algorithms, neural network and swarm intelligence

***Corresponding author.**
*E-mail address: satinderbal@gmail.com

This paper was recommended for publication in revised form by Editor in-Chief Ahmet Selim Dalkilic

Published by Yıldız Technical University Press, İstanbul, Turkey

[5,6]. These techniques are used successfully in different applications. For example, neural network algorithm for imitating the working of human brain, ant colony optimization algorithm is a SI technique inspired by the foraging nature of ants in real environment; genetic algorithms are inspired from Darwin's theory based on natural selection. In this paper, the different techniques have been discussed that are biologically inspired along with their accuracy, advantages, disadvantages and applications.

Literature Review

Biomimetics is a very popular field and a lot of research has already been done in this area. It refers to the transference of various strategies inspired from biological structures to technological field and is an emerging field in research which has led to remarkable concepts in the last few years. The number of research papers available on the biomimetic topic as well as the different techniques reviewed in this paper is very large. As the number of papers is very large, the authors in this paper have reviewed the latest papers written on these techniques. The authors in this paper have reviewed some latest literature about the popular techniques used by the researchers, which are biologically inspired. The review is shown with the help of Table 1.

Table 1. Related work

Sr. No.	Reference	Technique used	Objective	Conclusion
1.	Chiesa et al. (2020)[7]	Genetic Algorithm	To propose an implementation of GARS in order to identify important features in datasets of high dimensionality.	GARS was found to be a powerful tool for selecting features in high dimensional datasets.
2.	Drachal et al. (2021)[8]	Genetic Algorithm	To review the application of GA in order to forecast the price of commodities.	The authors focussed on the hybrid approach for analysing three classes of commodities which includes energy, metals and agricultural products. They used operations such as reproduction, mutation, crossover in order to create probabilistic models. The method was found to be efficient and fit for modelling the economic behaviour of human.
3.	Albadr et al. (2020)[9]	Genetic Algorithm	To present a new GA called GABONST (Genetic Algorithm Based On Natural Selection Theory) for improving the control of exploration and exploitation.	It was concluded that the GABONST algorithm was capable to produce solutions of good quality. The accuracy of the GABONST-ELM (GABONST with Extreme Learning Machine) was found to be 99.38%.
4.	Georgioudakis et al. (2020)[10]	Differential Evolution	To scrutinize and examine the performance shown by different variants of DE algorithm including Standard, Composite, Adaptive DE and Self-adaptive DE.	The COmposite DE (CODE) guaranteed robustness but the convergence speed was slow. Adaptive DE provided a compromise between convergence speed and optimum result.
5.	Du et al. (2019) [11]	Differential Evolution	To propose a multiscale cooperative DE algorithm for solving problems related to limited search space at the beginning stage and gradual convergence in the later stages.	The algorithm suggested by the authors was found to be better when compared to the other algorithms both in terms of accuracy as well as convergence speed.
6.	Nabil et al. (2020) [12]	Clonal Selection Algorithm	To explore the potential of binary form of CSA (Clonal Selection Algorithm) in solving the problem of feature selection.	The authors used three different datasets for comparing the Binary CSA and Optimum Path Forest Classifier with four other algorithms – Bat Algorithm, Flower Pollination Algorithm, Cuckoo Search Algorithm and Differential Evolution Algorithm. All these four algorithms were also in binary form. The proposed BCSA algorithm provided better classification accuracy and took less average time in execution.

Table 1. Related work (continued)

Sr. No.	Reference	Technique used	Objective	Conclusion
7.	Yu et al. (2020) [13]	Ant Colony Optimisation	To propose ACO algorithm for warehouse path planning.	It was concluded that the proposed algorithm had a faster rate of convergence and provided better stability in creating paths for different types of maps as compared to other algorithms. The smooth treatment of path resulted in reduction in the cardinality of paths and span of path in driving process.
8.	Kruekaew et al. (2020)[14]	Bee Colony Optimisation	To propose HABC (Heuristic task scheduling with ABC) algorithm.	It was found that the HABC-LJF (Largest Job First-ABC) outperformed all other algorithms such as ACO, PSO. It offered best performance in case of load balancing and scheduling.
9.	Hantash et al. (2020)[15]	Particle Swarm Optimisation	To propose improved PSO algorithm for improving voltage profile and reducing power loss.	The PSO algorithm suggested by authors reduced the number of active cases of power loss by 31.6%.
10.	Khan et al. (2021)[16]	Cuckoo search algorithm	To review Cuckoo Search Algorithm and its applications.	The authors focused on the feeding behavior of cuckoos and found that the algorithm could be used to solve many real-time problems including path planning, forecasting and so on.
11.	Zebari et al. (2020)[17]	Bat algorithm	To review bat algorithm.	The authors reviewed bat algorithm and its diverse applications including image processing, classification, feature selection etc.
12.	Odili et al. (2020) [18]	Flower pollination algorithm	To find out the best number of iteration and range of population for finding the best optimized solution in carrying out the diagnostic evaluation of algorithm.	It was concluded that better output can be obtained when appropriate population of flower and the iteration is used. Also, it was found to be a fast technique for solving optimisation problems.
13.	Yang et al. (2021) [19]	Flower pollination algorithm	To improve the flower pollination algorithm using chaotic mapping strategy.	The authors found that the improved FPA provided better results in testing the convergence of function and better accuracy in predicting geographic location.
14.	Altherwi et al. (2020)[20]	Firefly algorithm	To propose an effective method for the production companies for maximizing profits and minimising the cost of production.	It was concluded that the algorithm helped to achieve maximum profit and minimise the cost by producing a single product. The company was able to decide whether the available raw materials were sufficient to produce the product or not.
15.	Nayak et al. (2020)[21]	Firefly algorithm	To deeply analyse the variants, importance and applications of Firefly algorithm in different fields including biomedical engineering and healthcare.	Firefly algorithm was found to be an appropriate algorithm for solving biomedical and health care problems as compared to other algorithms. The algorithm was able to present the solution with less computational instances.
16.	EESA et al. (2021) [22]	Cuttlefish algorithm	Using Cuttlefish Optimisation algorithm for the task of classification in dealing with intrusion detection problem.	It was concluded that this algorithm provided good results when compared to other algorithms and produced 93.9% in precision, 92.2% in recall and 94 7% in area under curve respectively.

General Biomimetic Procedure

The biomimetic process can follow one of the following two approaches: (a) Bottom-up approach: In the first step of this approach the biological model is considered and the fundamental morphology of that biological system is analysed. In the next step, quantitative analysis involving the detailed understanding of the system is done. After this, the abstraction helps to separate the principles from the biological model. Then, the technical implementation of the system takes place and new product is introduced. (b) Top-down approach: This approach starts with the work of an engineer. In this, the technical products that already exist are considered for innovations. The two approaches can be shown with the help of diagram as shown below:

The Figure 1 shows the steps involved in the top-down as well as bottom-up approach of the biomimetic process. In top-down approach, firstly, the boundaries of the technical problem are clearly defined. Then, the sources of natural inspiration are found so that they can be used to solve the problem. Then, selection is done among the different solutions. After that, abstraction is performed which refers to separating the solutions from their natural examples. Then, technical implementation is checked and the prototypes are obtained. The new improved product is then introduced in the market. The bottom-up approach relies on the prior knowledge obtained from biological solutions and applying the concept on the existing design problems to obtain a new product. In this firstly, the biological system is analysed so as to develop an understanding of its shape, function and structure. After this, abstraction is performed which separates the principles from the existing biological model. These biological insights are then used to build a successful technical model by making use of laboratory and engineering scale. At last, a new product is obtained.

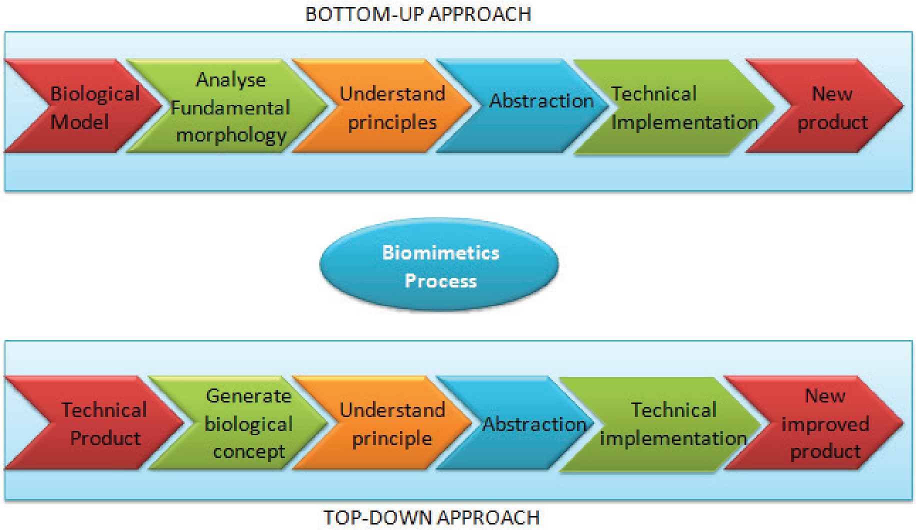

Figure 1. Biomimetic process approaches

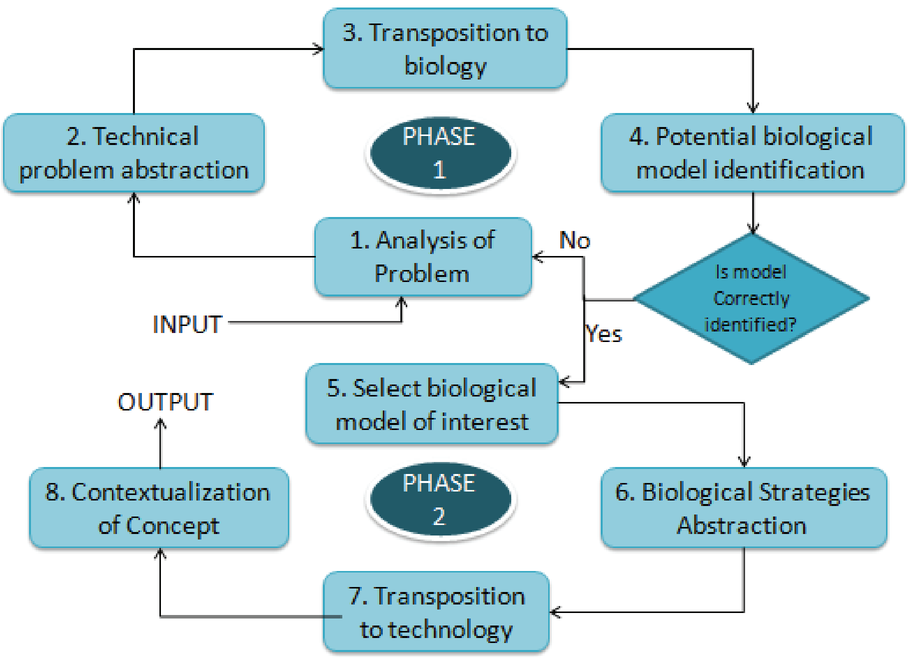

Figure 2. Phases of biomimetic process [4].

The biomimetic process [4] in simplified form is shown in Figure 2. The process comprises eight steps and these steps are subdivided into 2 phases. The first phase involves steps 1 to 4 and second phase involves steps 5 to 8. The major focus of the first phase is the transference of strategy from technology to biology. The second phase focuses on transposition from biology to technology. In this model, top-down approach is followed as it is easy to identify the boundaries of the technical problem to be solved. Also, the sources of natural inspiration can be found easily so as to solve the problem. The different steps are as follows:

Analysis of Problem

This step involves assessing the situation or the given problem. In this step, the problem is clearly identified and described so as to avoid the problem of complications in the solution.

Technical Problem Abstraction

This step helps in obtaining a functional model which explains the context and constraints related to the given problem and also give clarity about the function that needs to be achieved.

Tranposition to Biology

The functional model obtained from previous step helps in transposing the problem into biological form. It allows exploring the role of nature in managing the functions.

Potential Biological Model Identification

The process of identifying the biological model helps in developing a deep understanding of the given primary problem. This identification can be done by doing a research in the literature or by using search engines or databases. It helps in reformulating the problem and the biological analogy, if required. If the model is identified correctly, it moves to step 5, otherwise the above four steps are repeated.

Selecting the Biological Model of Interest

The biological models identified in the previous step are compared in this step so as to find the best model among them with respect to the initial technical problem.

Biological Strategies Abstraction

In this step, the biological strategies that are implemented by the selected biological models are understood and abstracted. This step helps to obtain a functional model of the biological system and is very important to find the feasibility of the perfect technology-biology correspondence.

Transposition to Technology

In this step, the biological strategies are transposed. The detailed description of the design principles and others is prepared so that the biological model could be technologically emulated.

Contextualization of Concept

It is the final step that involves implementing the technological process in the context of initial problem and then evaluating it. The cycle of the biomimetic process is

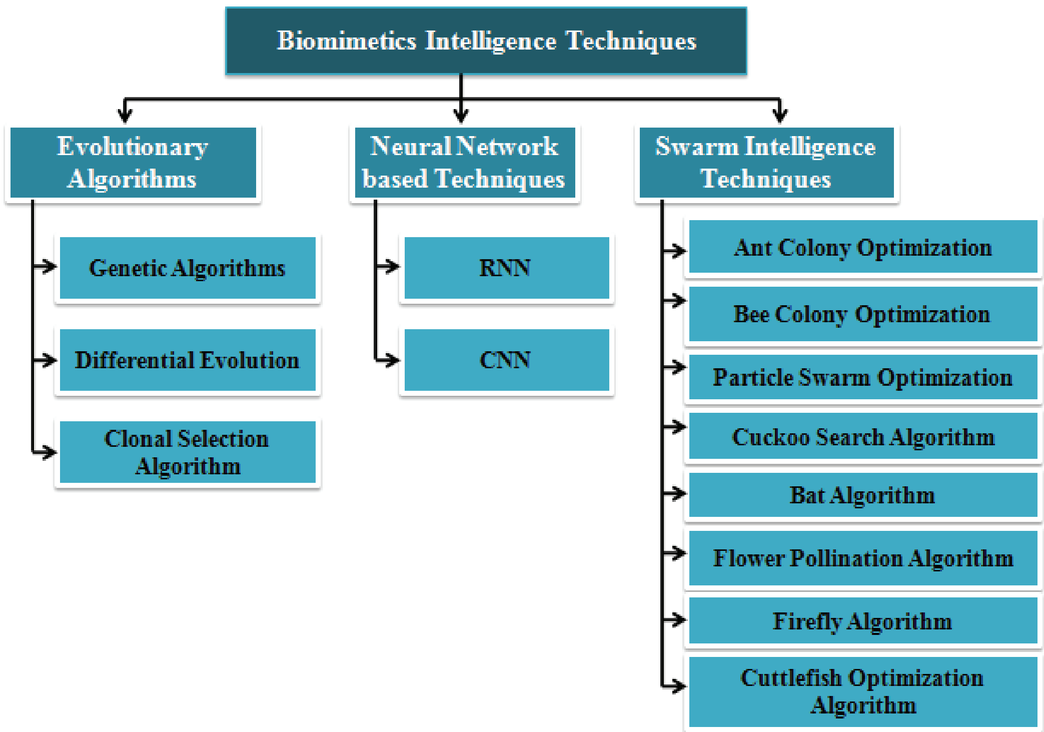

Figure 3. Popular biomimetic intelligence techniques.

successfully completed at this point. Now, the process is fully re-initiated if the outcome does not meet the expectations or instead of starting from the initial point, only phase 2 can be iterated by selecting the different interest model for the problem.

Popular Biomimetic Intelligence Techniques

Biomimetics includes technologies that are inspired from nature. The models and systems developed using this process helps in solving complex problems. It provides support that helps biologist and engineers to share their perspectives. It not only mimics biological systems but do a deeper study of living organisms to understand their living nature and environment in which the organisms operate. So, different techniques are used that help to develop algorithms that operates on the working of these organisms. Some of them are shown in Figure 3 below:

Evolutionary Algorithms

Evolutionary algorithms include all those algorithms that are inspired by nature and mimics the behavior of living things. It makes use of heuristic based approach to solve complex problems. These algorithms aim at maximizing or minimizing the objective function in order to determine the best optimal solution for an optimization problem. These algorithms work on the concept of evolution of biological processes. An initial 'population' comprising possible solutions of the given problem is created at first and then, a fitness function is allocated to each possible solution that indicates how good the solution is. The population evolves with time and better solutions are identified. Some EA algorithms specifically Genetic Algorithm, Differential Evolution Algorithm and Clonal Selection Algorithm are discussed below:

Genetic Algorithms (GA)

It refers to an algorithm which works on natural selection theory and helps in finding solutions for various optimization and search problems by making use of operators including crossover, mutation and selection. These operators are biologically inspired. The Darwin's theory of evolution which states "the survival of the fittest" works as a basis for GA. This algorithm is based on population, chromosome and gene. The fittest gene is found using the objective function known as fitness function which helps to find out the fitness of all the individuals present in the population [7-9,23]. The process of GA comprises combining different solutions using genetic crossover operator, using local moves with mutation operator and renewing the individuals in population for finding the best solutions using natural selection operator.

The Figure 4 shows the procedure of GA which involves the following steps:

Define Initial Population

It defines the possible values of the solution set, defined by $S = \{S_1, S_2, S_3,\ldots\ldots\ldots\ldots S_p\}$

Evaluation of Fitness Value

The g(S) = fitness value, where g represents the fitness function that helps in calculating the fitness of each one of the chromosomes, s present in the population.

Apply Operators

The different operators are applied on the chromosomes which include:

Selection

It can be considered as one of the important operators in the algorithm as it helps in increasing the chances of the survival of the individuals that are more fit. There are various selection methods that work well for different problems. So, an appropriate method needs to be chosen for the given specific problem for increasing the chances of getting best solution. The selection operator helps in selecting the chromosome on the basis of their fitness value. Based on the fitness values, more suitable chromosomes are selected and they are allowed to survive while unsuitable chromosomes are discarded.

Crossover

After selecting the chromosomes, the new offspring is created by applying crossover operator. This operator helps to exchange information between two selected parents. This operator plays a vital role in searching the best optimal solution from many solutions.

Mutation

After the application of crossover operator, mutation is applied in which genes of the offspring are changed. The main objective of this operator is to maintain diverseness in the population in order to reach the best likely solutions and overcoming the problem of local minima.

After this, the new population is transferred to the population (S). This process keeps on reiterating till the termination condition gets satisfied i.e. when there is no improvement in the newly found population or the newly obtained fitness value of the offspring is not preferable to the initial population.

Differential Evolution (DE)

Differential evolution (DE) is a biomimetic intelligence method that works on the theory of the survival of the fittest. It is considered as one of the most compelling algorithms for solving optimization problems among the various evolutionary algorithms. This algorithm involves use of mutation, selection and crossover operators for the purpose of imitation during biological evolution [10,11]. The key operator of DE is the differential mutation operator in which the individuals are randomly selected as parents from the possible solutions present in the population. It finds out the optimal solution by retaining the most fitted or the highly adaptable individual from the population [24,25]. It is a greedy algorithm that involves following steps:

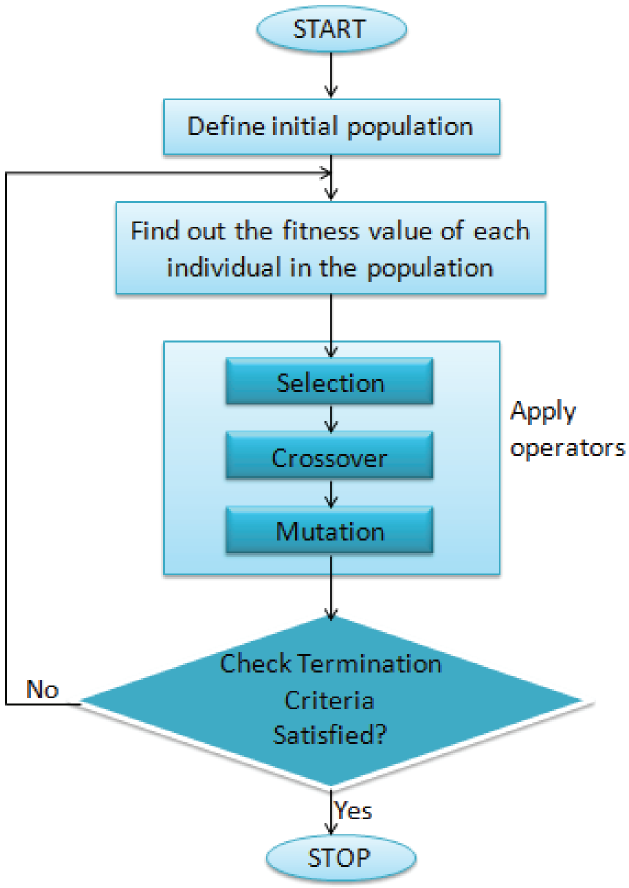

Figure 4. Steps of genetic algorithm [9].

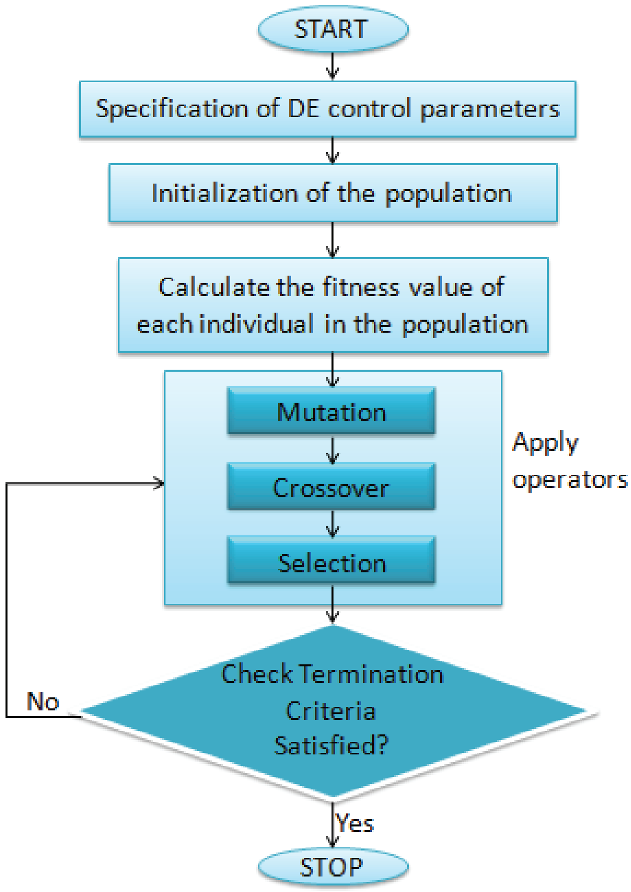

Figure 5. Steps of differential evolution algorithm.

The above Figure 5 shows the procedure of DE algorithm. Firstly, the control parameters – Mutation rate and crossover rate are specified. Then, the population is initialized. The fitness value is allotted to each individual. The operators are then applied and the whole process keeps on reiterating till the termination criterion is satisfied. The steps can be explained as:

Initialization

The population S_i^G is initialized as $S_i^G = \{S_{i1}^G, S_{i2}^G, S_{i3}^G \dots\dots\dots . S_{in}^G\}$ and i $\in \{1, 2, 3 \dots\dots\dots\dots, n(PS)\}$ where G represents the generation PS is the population size.

Mutation Operation

The chromosomes present in the current population, S_i^G creates a different chromosome, d_i^G by making use of mutation strategy

$$d_i^G = S_{r_1}^G + F(S_{r_2}^G - S_{r_3}^G),\ \ r_1, r_2, r_3 \in \{1,2,3 \dots\dots . PS\} \quad (1)$$

where r_1, r_2, r_3 are generated randomly and F represents the scaling factor whose value lies between [0,1]

Crossover Operation

New crossover individuals are generated by using crossover rate [C_r]. The operation is given as:

$$C_{i,j}^G = \begin{cases} d_{i,j}^G \ \ if\ random_j \le C_r\ or\ j = j_{rand} \\ S_{i,j}^G\ otherwise\ j = 1, 2, 3 \dots\dots\dots\dots . \end{cases} \quad (2)$$

where $random_j \in [0,1]$ $and\ j_{rand} \in \{1, 2, 3\dots \dots \dots .n\}$ $and\ C_r \in [0,1]$

Selection Operation

In this, the individual chromosomes are selected based on the allotted fitness value. If this assigned value of the new individual n_i is found to be better as compared to the previous one, then the new individual n_i is added to the previous population. Otherwise, n_i remains in the population of next generation only and performs mutation and crossover operations repeatedly. The selection operation is defined as:

$$S_i^{G+1} = \begin{cases} n_i^G\ if\ f(n_i^G) \le f(S_i^G) \\ S_i^G \quad otherwise \end{cases} \quad (3)$$

f(w) represents the objective function that need to be optimised.

Clonal Selection Algorithm

Clonal Selection Algorithm works on the principle of clonal selection. This algorithm is inspired by the body's defence mechanism i.e. immune system used for defining the characteristics of reaction of immune system to antigenic stimulus. The cells that are able to recognise the antigens are selected and these selected cells are allowed to proliferate. Natural immune system is used as an inspiration for developing an artificial immune system (AIS) that can be used to perform computational and pattern recognition tasks. The core components of this algorithm include operators such as cloning, mutation and selection. Cloning operator is responsible for generating new individuals having greater affinity values. Mutation operator produces disruption in clones in order to maintain the diverseness among the individuals present in the population. Selection operator helps in removing the lower affinity individuals from the population. This algorithm is based on the principle that only those cells are proliferated that are able to recognise antigens in the system. It means those cells which recognise antigens are selected and those which do not recognise are not selected. The selected cells are passed through similarity process which helps in increasing the similarity of antigens and cells [26,27]. Thus, it shows better performance in solving highly complex optimization problems and is used in various fields. The steps of clonal selection algorithm are shown in Figure 6.

Firstly, initialise the population randomly and also initialise different parameters such as number of iterations, mutation rate, cloning rate. Antibodies are generated randomly in population and find makespan value of each antibody. Sort antibodies on the basis of make span value/affinity and specify clone's number. The highest value antibodies are selected and the copies of the selected individuals are generated in proportion to their affinity with antigen. Then, mutation process is applied and the mutated individuals are then added to the population and the selected ones are deleted from the set of antibodies. Then the antibodies are evaluated again and then the entire population is updated. The best local and best Global makespan value is compared and the best one is selected and updated. If termination criteria get satisfied, then the results are displayed. Otherwise, the process is repeated [12,28,29].

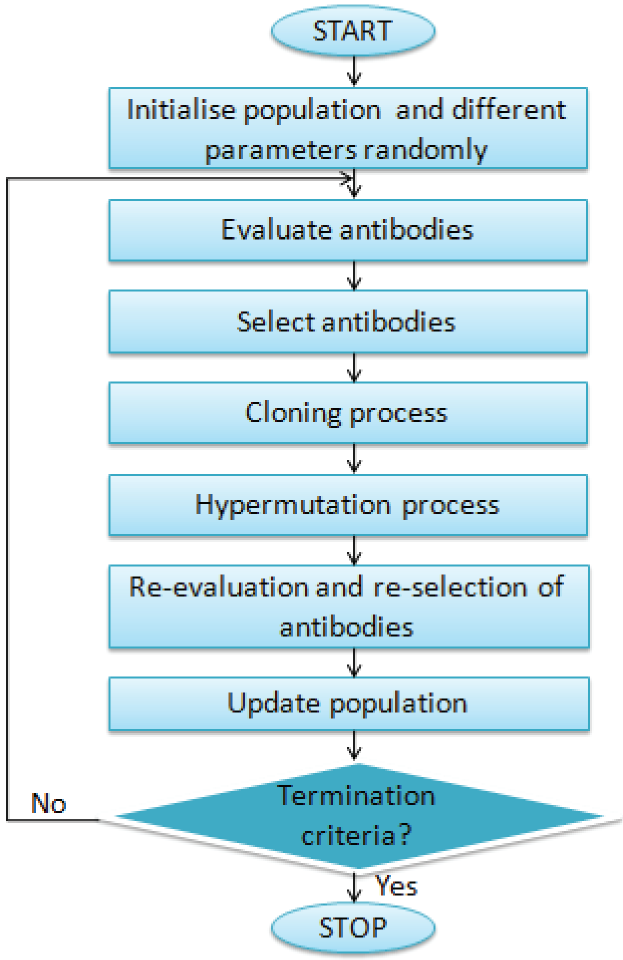

Figure 6. Clonal selection algorithm [28].

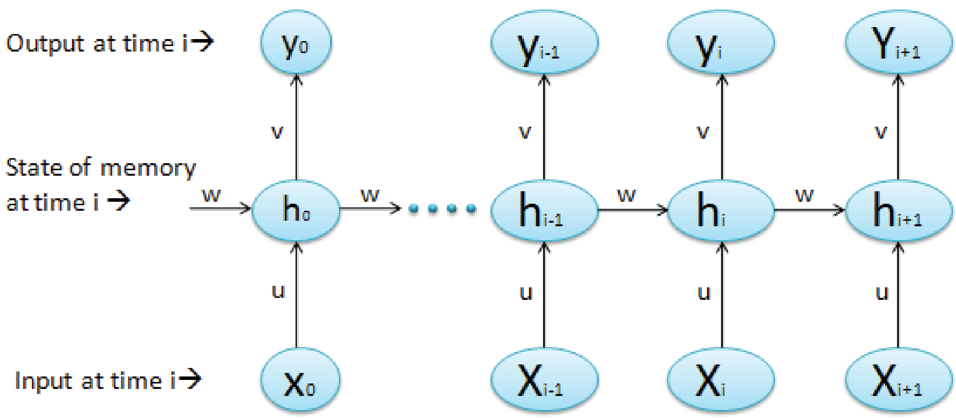

Figure 7. RNN with single layer of hidden memory.

Neural Network Based Techniques

The ANN (Artificial Neural Network) is inspired from the neurons working in the human brain. It consists of millions of processing nodes that are deeply interconnected. The ANN is adaptive in nature and can modify itself from the learning obtained during initial training. It can be feed forward network in which information flows in forward direction only and feedback network in which information flows in backward direction also. [30-32]. It includes following types:

RNN (Recurrent Neural Network)

It is a kind of ANN that uses consecutive data and is commonly used for solving temporal problems. It can handle sequential data of variable length. It uses training data for the purpose of learning. It uses generalized delta learning rule and energy minimization function for the purpose of training. It is most commonly used for data classification. It is a type of NN in which the model parameters and the operations performed are same throughout. For each element in a sequence, the same task is performed by the network. It is a feedback neural network where the operation is dependent on both the input and the previous state. The Figure 7 shows the RNN model in which the model parameters are u, v, w which are same throughout the architecture.

CNN (Convolutional Neural Network)

It is a type of machine learning ANN. The architecture of CNN involves following layers:

The Figure 8 shows the layers involved in CNN model. The different layers perform different functions as explained in Figure 9:

Convolution Layer

In this layer, pixel by pixel scanning of the input is done so as to create a feature map for defining future classification. It creates a new image by applying filters on the input image. These filters help to keep some information and ignore the rest of the information.

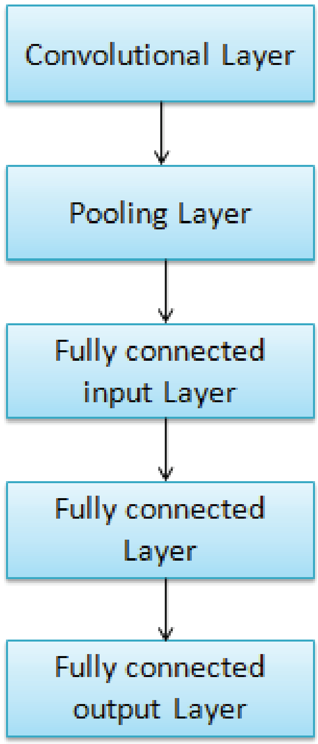

Figure 8. Layers of CNN.

Pooling Layer

This layer is present after the convolution layer. It is used for reducing the dimensions in the feature map. This layer helps in summarizing the features present in the feature map region created by the convolution layer.

Fully Connected Input Layer

The input to this layer is the outcome obtained from the pooling layer. It is used for flattening the image and converting the image into a single vector. The output is fed into the next layer called fully connected layer.

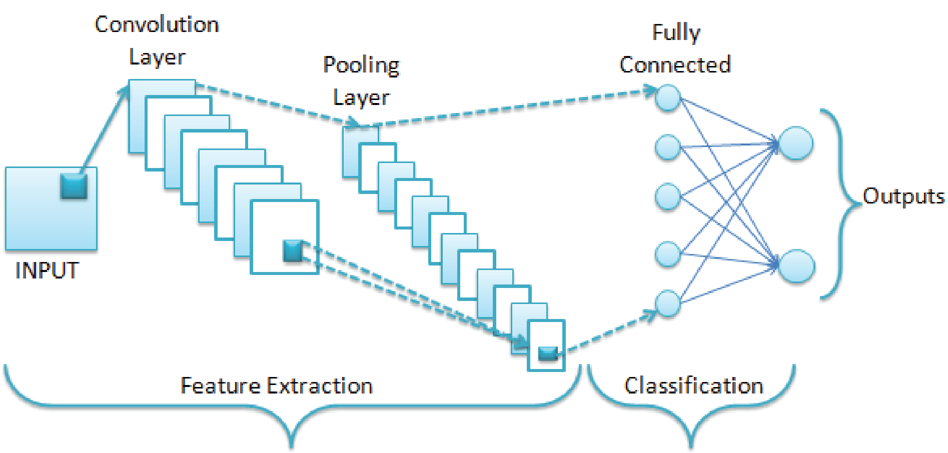

Figure 9. Working of CNN layers.

Fully Connected Layer

It is considered to be a feed forward neural network in which computation is performed by assigning the random weights to the inputs.

Fully Connected Output Layer

This layer helps to do the classification of images by obtaining the probabilities of the input.

Swarm Intelligence Techniques

Swarm Intelligence (SI) is an AI based approach that gets inspiration from the nature for solving optimisation problems. These are optimization algorithms inspired from nature and are very powerful techniques used to solve NP-hard problems which are difficult to be solved for conventional algorithms. It works on the basis of the cumulative behaviour of natural organisms in self-organised systems. Some of the important SI techniques are discussed below:

Ant Colony Optimization (ACO)

This algorithm gets inspiration from the natural behaviour of ants. This technique is based on probability for finding solution of the computational problems. In ACO, artificial ants are used to find the solution to optimization problem. In this algorithm, the optimal path is found on the weighted graph. The artificial ants move on the graph in order to find the solution. When ants move on the graph from point to B, they mark the path by leaving a chemical called pheromone. This becomes helpful for the following ants to find the path by detecting pheromone and the path having greater concentration of pheromone gets selected. The pheromone acts as an indirect medium of communication between ants. [13,33,34]. So, this algorithm is based on probability. The ant-colony works well with a dynamic system and thus, the algorithm is used in graphs with changing topology.

The Figure 10 shows ACO algorithm stepwise. In this algorithm, firstly the parameters are initialized which involves: number of ants, number of iterations, and pheromone rate of evaporation. Initially, pheromone content in the environment is null as all the ants are in the nest. After that, iteration starts and the ants begin the search along each path with a probability of 0.5. Then ants move to another node and the ant following the shorter path reaches earlier to the food. An ant must be able to select the next node at each level and the selection probability to move from lth node to nth node is given by:

$$SP_{l.n} = \frac{(\tau_{l,n}^{\alpha})(\eta_{l,n}^{\beta})}{\sum_{m \in S}(\tau_{l,n}^{\alpha})(\eta_{l,n}^{\beta})} \tag{4}$$

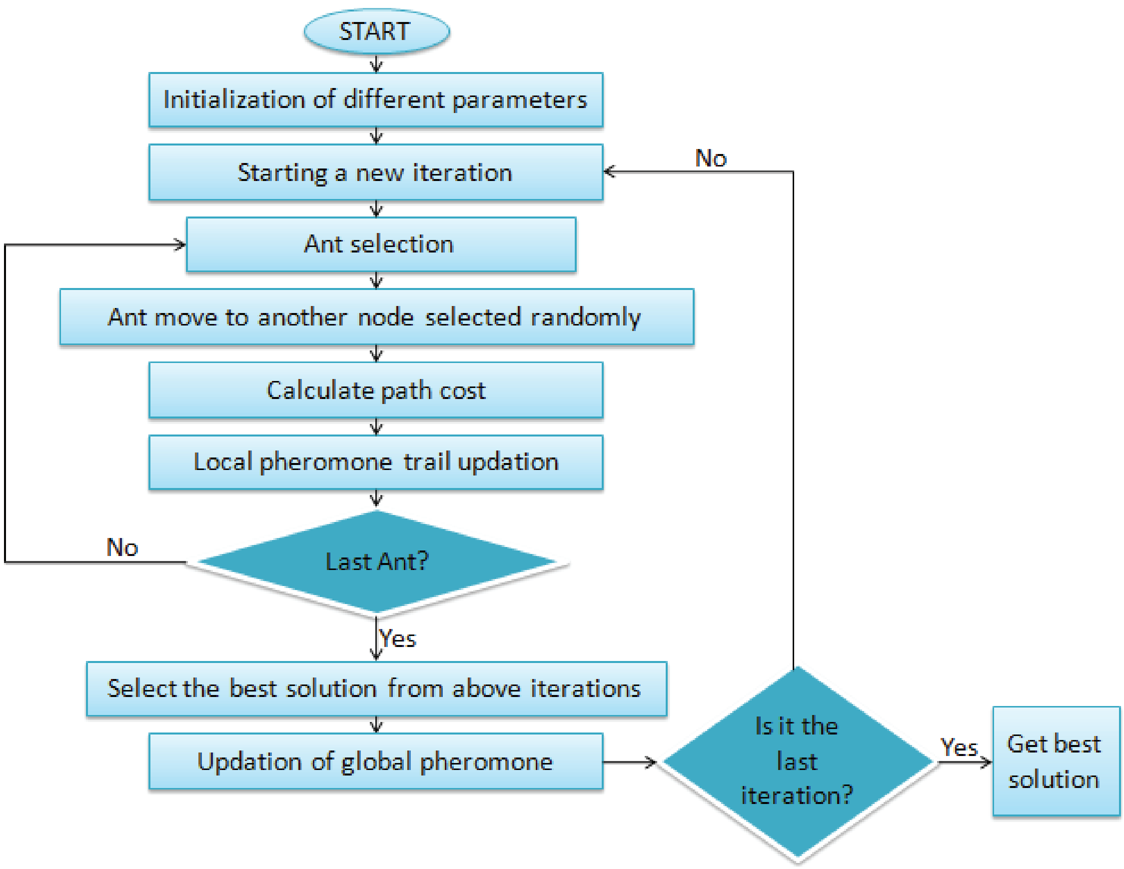

Figure 10. ACO algorithm [13].

where $\tau_{l,n}$ represents the pheromone concentration on path l to n

$\eta_{l,n}$ represents the heuristic function that helps in selecting nth node from lth node

α controls importance of pheromone

β controls importance of heuristic function

The selection of next node depends on $SP_{l,n}$ value of node. After completion of tour, the pheromone concentration is updated as:

$$\tau'_{l,n} = \tau_{l,n} + \tau_{l,n}.\left[\left(1 - \frac{1}{1+SP_{l,n}}\right)\right] \tag{5}$$

where $\tau'_{l,n}$ is the new updated value and $\tau_{l,n}$ is the preceding value of pheromone

The concentration of pheromone increases when more number of ants travel on the path. This concentration also decreases as it keeps on evaporating. So, the convergence of algorithm is not easy locally and is specified as:

$$\tau_{l,n} = (1-\rho).\tau'_{l,n} + \Delta\tau_{l,n} \tag{6}$$

where ρ represents percentage of pheromone that is evaporated and $\Delta\tau_{l,n}$ defines fitness of ant on a path. This refers to the global pheromone updation rule.

Bee Colony Optimization (BCO)

It is a kind of optimization algorithm that is based on the perspicacious behaviour of honey bees. A colony of bees can be considered as a swarm in which bees are individual agents. Each bee acts as a low-level component that works through swarm at global level to form a system. The local behaviour of individuals helps to determine the global behaviour of system and thus results in organised teamwork. The bee-colony comprises three classes- employed bees, onlooker bees and scout bees [14]. The colony i divided into two halves- the initial half consists of employed bees and the next half comprises onlooker bees. Only one employed bee is present for each food source. Scout bee has the responsibility of finding new food or nectar source. It is that employed bee whose food source has been forsaked by artificial bees and it needs to discover a new food source. Onlooker bees take the findings from the employed bees present in hive regarding different food sources and select one of those sources to collect nectar. Employed bee stays on food source [35].

The Figure 11 shows the BCO algorithm stepwise. Initially, the scout bees select a food source at random using equation:

$$S_{low_k}(j) = S_{min_k} + rand[0,1](S_{max_k} - S_{min_k}) \tag{7}$$

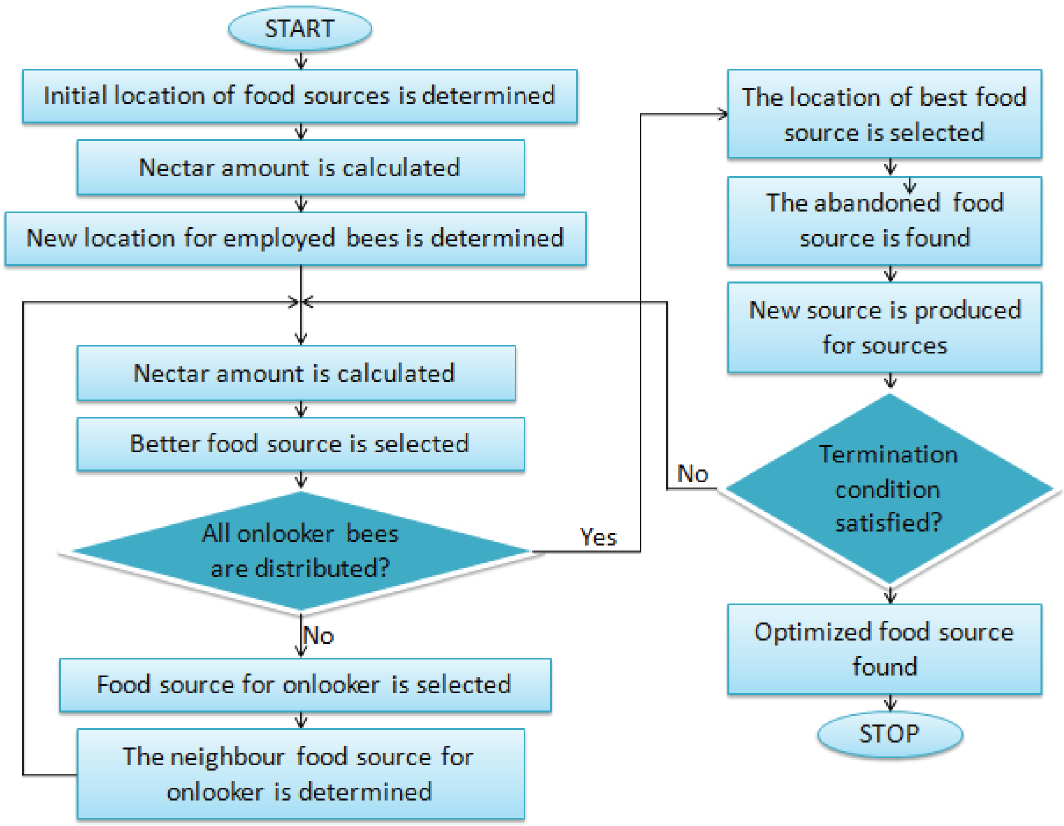

Figure 11. BCO algorithm [14].

$$S_{up_k}(j) = S_{min_k} + rand[0,1](S_{max_k} - S_{min_k}) \quad (8)$$

where $S_{low_k}(j)$ represents the lower bound of kth attribute related to bee j.

$S_{up_k}(j)$ the upper bound of kth attribute related to bee j.

rand[0,1] is a random number which is evenly distributed between interval [0,1].

The lower bound and upper bound are exchanged on the basis of greater value. Then, quality of sources is accessed and half of the sources are preserved as initial source. Then, each employee bee chooses a food source randomly in neighbourhood of current source with the help of equation:

$$newS_{low_k}(j) = S_{low_k}(j) + rand[-1,1]\left(S_{low_k}(j) - S_{low_k}(i)\right) \quad (9)$$

$$newS_{up_k}(j) = S_{up_k}(j) + rand[-1,1]\left(S_{up_k}(j) - S_{up_k}(i)\right) \quad (10)$$

where $newS_{low_k}(j)$ and $newS_{up_k}(j)$ represents the lower and upper bound of n_{th} feature which is related to the ith new food source of bee. Rand[-1,1] is a random number which is evenly distributed between interval [-1,1].

Now, the wiggle dance done by the employed bees is observed by the onlooker bees. Then, the probability of each food source is calculated by the onlooker in order to select the best source using equation:

$$P_k = \frac{fkt_k}{\sum_{k=1}^{N} fkt_k} \quad (11)$$

where fkt_k represents fitness of the source utilised by employed bee i.

If the better food source is not found by the employed bee, then it gets transformed into a scout bee and the other source is selected.

Particle Swarm Optimization (PSO)

The PSO algorithm gets inspiration from a flock of birds called swarm. As all the birds are starving, they are in search of food. The birds are associated with different jobs in the computing system starving for resources. Also, only one food particle is present in the locality of birds. This food particle is associated with a resource. There are many jobs, but the resources are limited. The birds do not know the whereabouts of the food particle as it is hidden. The algorithm is designed to discover the hidden food particle as havoc can be created if each bird tries to find out the food particle itself. The birds do not know the location of hidden particle but they know the distance of the particle from itself. So, the approach of algorithm is to go after the

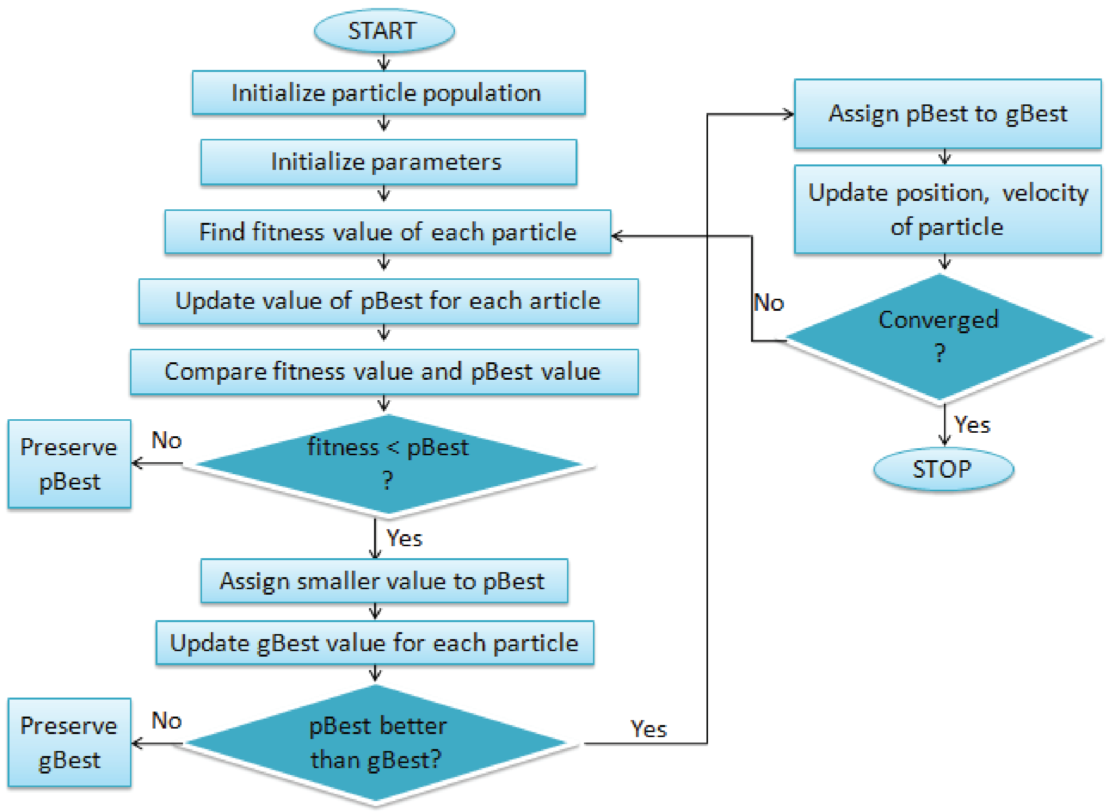

Figure 12. PSO algorithm.

birds which are present near the food particle [36,37]. It is a stochastic algorithm that is based on population.

The Figure 12 shows the PSO algorithm stepwise. Firstly, population of different particles is initialized randomly. The control parameters are initialized which includes number of iterations, position limit, velocity limit, social and cognitive factors. Initially, the best position of particle is its current position i.e.

$$bp_k^d(0) = x_k^d(0) \tag{12}$$

where $bp_k^d(0)$ denotes the best position of k[th] particle in d dimensional space and $x_k^d(0)$ represents the k[th] particle in the dimensional space, d.

Then, the fitness value of every particle is computed. This value is then compared with pBest value. The smaller value is assigned to pBest and then gBest value compared with the updated pBest and smaller value is assigned to gBest. After this, the position of particle and the velocity are updated. The velocity of k[th] particle at any time t is given by:

$$v_k^d(t+1) = w(t)v_k^d(t) + \alpha\left[pBest^d - x_k^d(t)\right] + \beta\left[gBest^d - x_k^d(t)\right] \tag{13}$$

where $v_k^d(t+1)$ represents the new velocity of k[th] particle in dimensional space, d and at time t+1, α and β are random vectors whose value lies in the interval [0,1], $pBest^d$ represents the best preceding position of k[th] particle in the dimensional space d, $gBest^d$ represents the best preceding position of k[th] particle among different particles present in the population.

The updation of the position of particle is done as follows:

$$x_k^d(t+1) = x_k^d(t) + v_k^d(t+1) \tag{14}$$

where $x_k^d(t+1)$ represents the next position [15,38].

Cuckoo Search Algorithm

This heuristic algorithm is used for solving problems related to optimization. This algorithm gets inspiration from nature and depends on the lifestyle of bird family named Cuckoo. It takes inspiration from the parasitic strategy of cuckoo bird and also takes into account the Levy flight mechanism of fruit flies and birds and their random walks. Cuckoo is a fascinating bird for two reasons: (a) Beautiful sound (b) Aggressive strategy of reproduction. Cuckoo Birds uses this strategy for laying their eggs in the nest of other species. In the nest, each egg denotes a solution and on the other hand, cuckoo egg denotes a new solution. The egg laying and breeding of cuckoo birds form the basis of this optimization algorithm. The cuckoo species named *Guira* and *Ani* cuckoos make use of communal nests for laying their eggs. In order to raise the probability of hatching of their own eggs, they remove other species eggs from the nest. If the host bird comes to know that the eggs are not theirs, they engage in direct conflict with the cuckoo birds and either throw the eggs of cuckoo birds or leave their own nest. While the *Tapera* cuckoo species includes female parasitic cuckoo that is specialized in doing mimicry of the color as well as pattern of eggs of some of the chosen species. It helps to increase the probability of reproductivity of their eggs. Animals randomly search for food in the nature and the path is a random walk as the current location and the probability of the next location acts as the basis for the next move of the animal. So, Levy Flight mechanism is used for this purpose. In this algorithm, the step-length is calculated on the basis of heavy-tailed probability distribution. [16,39]. The Cuckoo search algorithm follows the following three rules:

Rule 1. Each of the Cuckoo birds can lay only one egg at a time and the nest for this egg is randomly chosen.

Rule 2. The nest having the highest quality eggs is considered the best and they are carry-forward to the next generation.

Rule 3. The available host nests are predetermined and the alien egg is found by host with a probability, P and P $\in$ [0,1].

The Figure 13 shows the CSA algorithm stepwise. In this firstly, the initial population is generated which involves N host nests. A cuckoo is selected which lays an egg and a nest is chosen randomly for that egg. The chosen nest has the host egg and the fitness of the host egg and the Cuckoo egg is evaluated and compared and the better one stays in the nest. If the host bird notices this, then it abandons the nest and builds a new one and this process keeps on reiterating till the number of iterations is completed. [40-42].

Bat Algorithm

It is a novel meta-heuristic algorithm which is biologically inspired by the behaviour of bats. [17]. This algorithm depends on the echolocation of bats for performing global optimisation. The artificial bats are used for the searching process as search agents. These bats mimic the behaviour of real bats including pulse, loudness as well as emission rate. The bat uses echolocation for finding the food. [43]. The microbats have a highly sophisticated sense of hearing sounds. When bats hit any object in their path, a sound gets emitted by the bouncing. After some fraction of time, the echo of sound returns back to the ear of bats. This echo helps the bats to find out the distance and size of bats and their travelling speed. The bat algorithm follows the following three rules: (a) All the bats are able to find out and locate the prey by using echolocation (b) The bats uses frequency, f_{min} to fly from its current position, p_i and with a velocity of v_i. The loudness and frequency varies (c) The loudness varies from minimum to maximum value i.e. from L_{min} to L_{max}.

The Figure 14 shows BAT algorithm stepwise. In this, first of all, the population of bats, b_i is initialised. Each individual in the population has defines position $p_i(t)$ and velocity $v_i(t)$ where t denotes the given point of time. The position as well as the velocity is updated with the increment

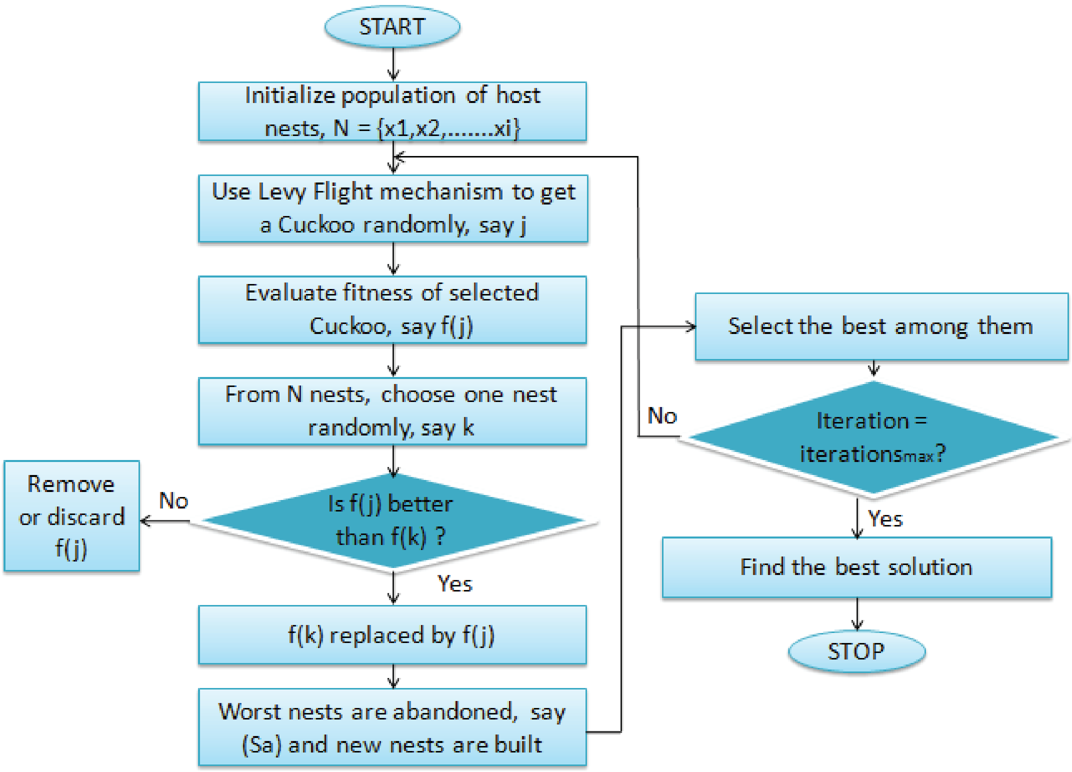

Figure 13. CSA algorithm [40,41].

in number of iterations. The positions and velocities are updated as follows:

$$p_i(t+1) = p_i(t) + v_i(t+1) \text{ and}$$
$$v_i(t+1) = v_i(t) + \left(p_i(t) - o(t)\right).X_i \qquad (15)$$
$$X_i = f_{min} + (f_{max} - f_{min}).\alpha$$

where α is a random vector whose value lie in the range of [0,1]. O(t) represents the current globally optimum solution and f_{min}=0 and f_{max}=1.

Depending upon the parameters, the algorithm has different search capabilities either local search or global search. So, a balance needs to be maintained by global and local search and this is done with the help of different parameters.

The local search strategy formula is:

$$p_i(t+1) = \vec{o}(t) + \varepsilon\vec{L}(t) \qquad (16)$$

where ε is any random number between [-1,1] and $\vec{L}(t)$ represents average loudness of the initial population.

The global search strategy formula is achieved by controlling the pulse rate $p_i(t+1)$ and loudness $L_i(t+1)$. The formula is:

$$L_i(t+1) = \beta L_i(0) \text{ and}$$
$$p_i(t+1) = p_i(0)[1 - exp(-\gamma t)] \qquad (17)$$

where β and γ are some constants and $\beta > 0, \gamma > 0$. $L_i(0)$ and $p_i(0)$ are primary values of loudness and pulse rate.

In this algorithm, for each bat some number ($0 < R_1 < 1$) is generated randomly and then fitness value for the corresponding bat is calculated. Now, the random number and the fitness value are compared and the best one is chosen. Now, a random number ($0 < R_2 < 1$) is generated for each bat. Now, the loudness and pulse rate are updated and loudness is compared with R_2 and new fitness is compared with the previous one. On the basis of allotted fitness value, every individual in the population is sorted and the best position is saved. Now, the termination criterion is checked and if it is satisfied, then algorithm stops. Otherwise, the above process is repeated [44,45].

Flower Pollination Algorithm

It is a heuristic algorithm which gets inspiration from the nature and is based on the behaviour of pollination in the flowering plants. Its main objective is the production of optimal reproduction of plants by finding the fittest flower among the flowering plants. It imitates the process of reproduction shown by flowering plant. The main goal of flower is reproduction and it is achieved with the help of pollination [18,19]. The process of flower pollination is accompanied with transferring the pollen by making use of pollinators such as bats, insects, and birds and so on. The two processes for transferring pollen includes: (a) biotic and cross pollination (b) abiotic and self pollination. Biotic

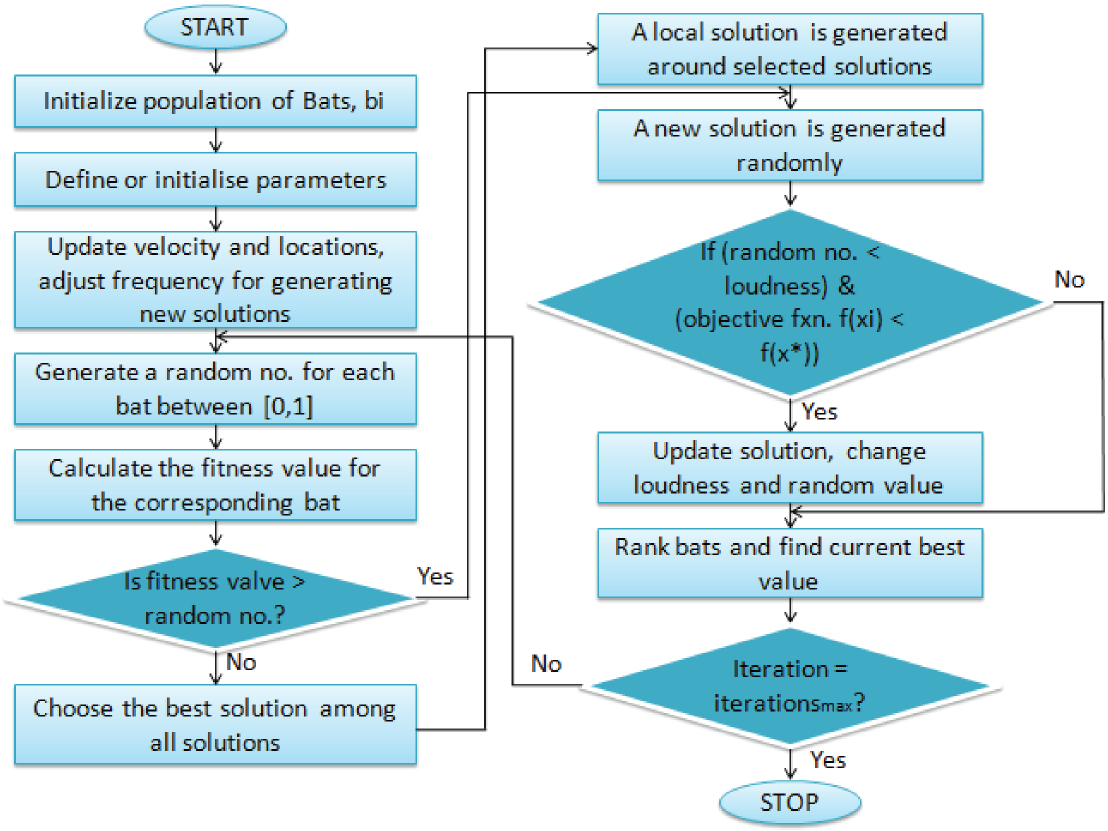

Figure 14. BAT algorithm [17].

pollination involves transfer of pollen from one flower to other in the different plant with the help of pollinator such as birds, insects etc. Cross pollination process can occur at long distance and thus, it is called Global pollination process in which pollinators perform levy flights. Abiotic or self pollination process involves fertilization of a flower from the pollen of that same flower to a different flower of same plant with the help of wind and water. It is called local pollination [46].

The Figure 15 shows the steps involved in FPA. In this firstly, the population p_i is initialised where i = 1, 2, 3.........n. Then, the parameters are initialised which includes size of population, n, switch probability s, where s ∈ [0,1] and number of iterations. A random solution is generated by finding out the fitness function of every solution present in the current population. This random solution is compared with the switching probability and if it is greater, then global pollination process starts, otherwise local pollination process starts. The new solution is generated with the help of levy flight distance as $-P_i^{t+1} = P_i^{t+1} + L(P_i^t - b^*)$ where t = iteration number, L = pollination strength derived from Levy distribution, b* is the best solution at iteration t.

Local pollination process is defined as $-P_i^{t+1} = P_i^t + \epsilon(P_j^t - P_k^t)$ where P_j^t and P_k^t are pollens from different flowers of same plant.

The solution is then updated and best solution b* is found after the iterations are completed [47,48].

Firefly Algorithm

It is a very important tool in different SI techniques as it can be applied in various areas related to optimization. It can be applied to solve NP-hard problems. It is a stochastic algorithm that uses randomisation to search from a set of available solutions. It is a meta-heuristic algorithm which gets inspiration from the behaviour of Fireflies. It takes into account their flashing behaviour and the process of food searching during the night. The Fireflies get attracted to the firefly which has more brightness [20,21]. The flashing light which is emitted by the Firefly acts as a mode of communication among the Fireflies. The flashing pattern depends on the type of species of Fireflies. The Firefly emits two different lights- one indicates food source availability and other one indicates the mating of Fireflies. It follows following rules – (a) The Fireflies are considered as unisexual and thus, one Firefly attracts the other one irrespective of the sex (b) Attractiveness which is directly proportional to the brightness which means a Firefly is more attracted to the brighter one (c) The brightness and distance are inversely proportional to each other which means as the distance increases, the brightness decreases (d) The brightness of

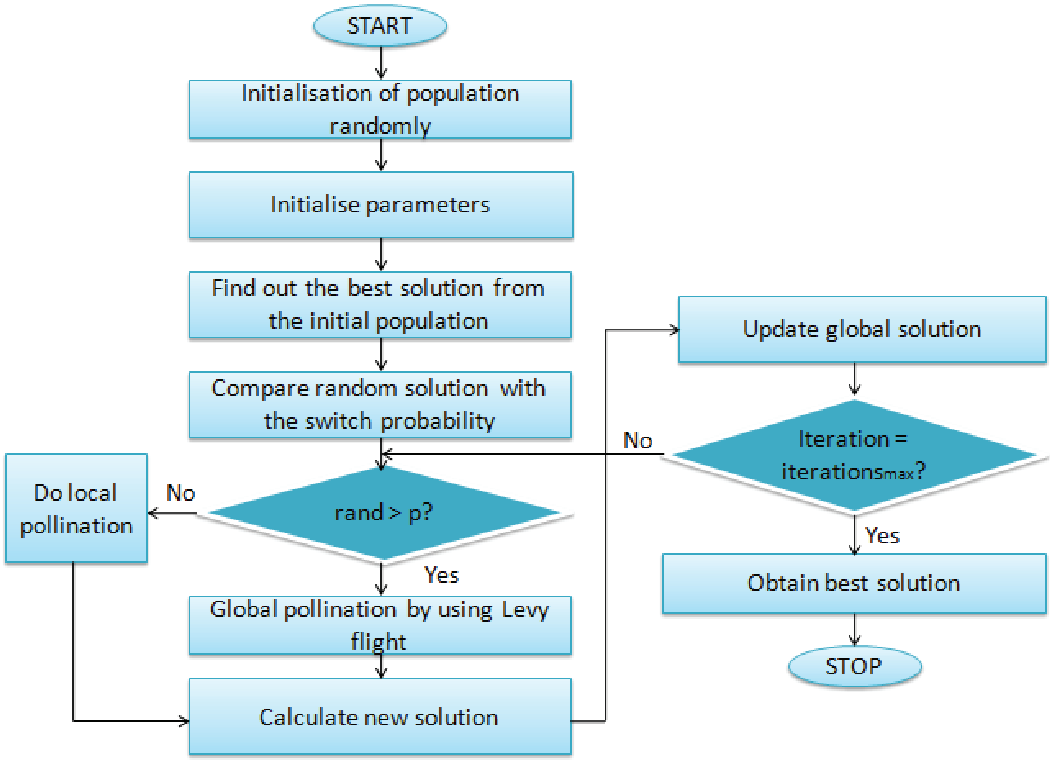

Figure 15. Flower pollination algorithm [46].

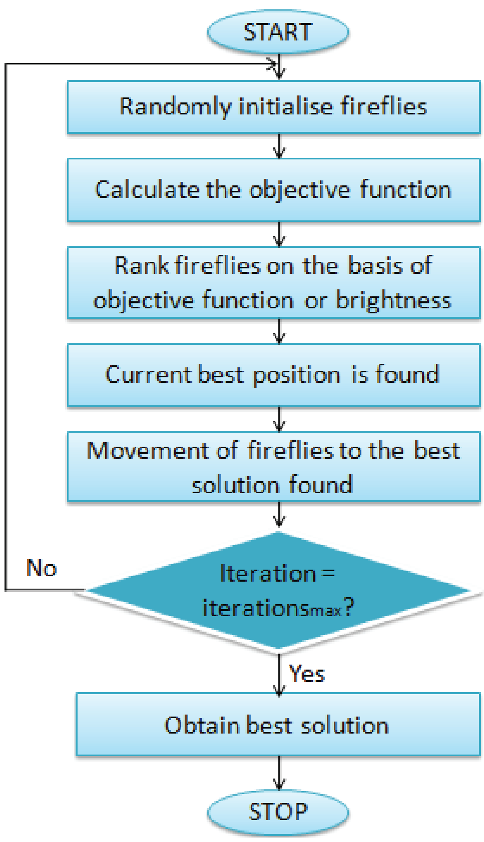

Figure 16. Firefly algorithm [50,51].

Firefly if found by objective function [49]. The brightness is directly proportional to the fitness of objective function.

The Figure 16 shows the steps involved in Firefly algorithm. In this, firstly the attractiveness of Firefly is found and then the movement of Firefly to the best solution is found. As we know that the attractiveness and brightness are directly proportional i.e. $A(d) = \frac{A_b}{d^2}$ where d represents the distance between fireflies and is inversely proportional to the attractiveness (A).

If there are n fireflies, and f_i represents the solution for the i^{th} firefly. The objective function $O(f_i)$ represents brightness of firefly i and is represented as $-A_i = O(f_i)$. Now, attractiveness value α is associated with each of the fireflies and is given as $-\ \alpha(d) = \alpha_0 e^{-\beta d^2}$ where α_0 represents attractiveness value of firefly at d=0 and β denotes the coefficient of light absorption. Now, the fireflies move to the brighter one and it can be represented as –

$$f_i(t+1) = f_i(t) + \alpha_0 e^{-\beta d^2}(f_i - f_j) + \beta\varepsilon_i \tag{18}$$

where $\alpha_0 e^{-\beta d^2}(f_i - f_j)$ represents attraction of j^{th} firefly to ith firefly. $\beta\varepsilon_i$ is the randomized parameter.

If the attractiveness value is not high, then the firefly moves randomly using equation –

$$f_i(t+1) = f_i(t) + \beta\varepsilon_i \tag{19}$$

The steps are repeated till the number of iterations is completed [50,51].

Cuttlefish Optimisation Algorithm

Cuttlefish is a cephalopod which has the ability to change its colour either for disappearing into the environment or for producing displays that are stunning. The different patterns and colours produced by cephalopods are due to the light reflected by different layers of cells that are bundled together [52,53]. The different layers include: chromatophores, iridophores and leucophores. The function of these layers is explained below:

Chromatophores

It represents the collection of cells that involve an elastic saccule for holding pigment and about 15 to 25 muscles are attached to this saccule. These types of cells are present under the skin of Cuttlefish. While the muscles contract, the saccule gets stretched which allows the pigment to go inside and cover the large surface area. While muscles relax, the saccule gets shrinked which hides pigment.

Iridophores

It is the next layer present under the chromatophores. It is stacked with layer of platelets that is either chitin or protein based depending on the species. It helps to produce green, blue and gold looking metal in some species and silver colour around eyes. It works by reflection of light and is used for concealing organs and communication.

Leucophores

This layer has the responsibility of white spots that are present on Cuttlefish. These are flattened and branched cells that help in scattering and reflecting the incoming light. It means that their colour reflects the wavelength of light in environment. It helps the animal to adjust in the environment.

CFA depends on two different processes – one is reflection and other is visibility. Reflection is a mechanism that helps the Cuttlefish in reflecting the incoming light. Visibility helps to provide clarity in the matching pattern. The two mentioned processes are used for finding the Global optimum solution.

The Figure 17 shows Cuttlefish algorithm stepwise. In this, firstly the population P_i gets initialised where i = 1, 2,3 ... n. The best solution is found and kept in Best and the average of best solution is kept in Avg_{best}. Then, the population is dissected into four groups. The working of every group is different and the best solution is found in all groups. For Local Search, Group 1 and 4 work are used and for Global Search, Group 2 and 3 are used.

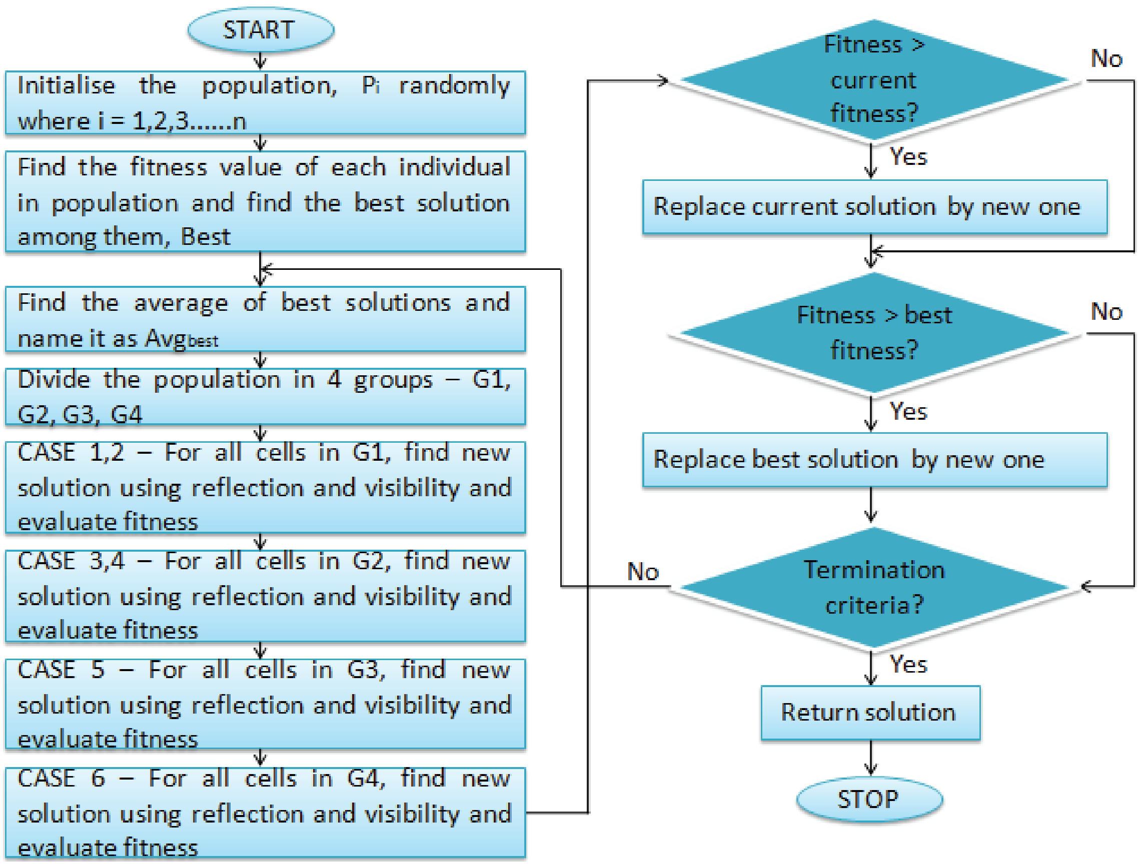

Figure 17. Cuttlefish optimisation algorithm [54].

Group 1

In this, reflected light is generated due to interaction between cells of chromatophores and iridophores. The reflection and visibility is found using equations –

$$reflection_k = T * G_1[j].Points[k]$$
$$visibility_k = B * (BestPoints[k] - G_1[j].Points[k]) \quad (20)$$

where G_1 is group 1 consisting of chromatophores cells, j is the jth cell of G1, Points[k] denotes the kth point of jth cell, BestPoints represent the best solution, T denotes reflection degree, B represents visibility degree.

The value of T and B are found using equations –

$$T = rand(\,) * (t_1 - t_2) + t_2 \text{ and}$$
$$B = rand(\,) * (b_1 - b_2) + b_2 \quad (21)$$

where rand ( ) represents the random function whose value lie between (0,1). t_1 *and* t_2 are fixed values for finding stretch interval. b_1 *and* b_2 are used to find visibility degree interval.

Group 2

The incoming light is reflected by the iridophores cells. The reflection is found using-

$$reflection_k = T * BestPoints[k] \quad (22)$$

Group 3

Leucophores cells are used in this group. The reflection and visibility are updated as-

$$reflection_k = T * BestPoints[k] \text{ and}$$
$$visibility_k = B * BestPoints[k] - Avg_{best} \quad (23)$$

where Avg_{best} represents the average value of best points.

Group 4

Leucophores cells are used for reflecting the light coming from the environment. The comparison of fitness value is then done with the new solution and then with the best solution and the result is updated accordingly. At last, the best solution is found [22,54].

Comparative Analysis of Biomimetic Intelligence Techniques

The different popular techniques used in Biomimetic Intelligence have been discussed by the authors in the previous section. In this section, comparison among the techniques has been shown with the help of a table. In this table, the advantages and disadvantages of each techniques has been mentioned. Also, the real world applications in which these techniques have been applied are mentioned so as to get a better understanding of the use of these techniques [7,8,10,11,18,20,23,26,28,30,31,37,39,42,43,55-74]. The Table 2 shows the advantages, disadvantages and applications of the techniques.

Table 2. Comparative analysis of BI techniques

Sr. No.	Techniques	Accuracy of Results	Merits	Demerits/Limitations	Applications
EVOLUTIONARY ALGORITHMS					
1.	Genetic Algorithm (GA) [7-9,23] Proposed by John Holland in 1975	The accuracy obtained by this algorithm ranges from 95% to 99% depending on the data sets and other techniques used with this algorithm	➢ Easy to understand the implementation process ➢ Works in Parallelism ➢ Easy to implement ➢ Used for solving engineering problems that are very complex ➢ The chances of getting optimal solution are more ➢ Require less computing resources	➢ Slow method ➢ Time consuming ➢ Suffers from degeneracy i.e. multiple chromosomes may represent the same solution and thus, can find inefficient solution. ➢ Optimization is difficult ➢ Difficulty in scaling with respect to complexity	➢ Feature selection ➢ Traffic and Shipment Routing (Travelling Salesman Problem) ➢ Engineering Design i.e. Designing of Aircrafts, Electronic Circuits, Antenna design ➢ Robotics ➢ Air flight Schedule Planning ➢ Bio informatics: Multiple Sequence Alignment ➢ Power system generator maintenance scheduling problem ➢ Drug generation for diagnosis of disease in body

Table 2. Comparative analysis of BI techniques (continued)

Sr. No.	Techniques	Accuracy of Results	Merits	Demerits/Limitations	Applications
2.	Differential Evolution (DE) [10,11,24,25] Proposed by K.V. Price and R. Storn in 1995	The accuracy of results lies between 94% and 99%. The accuracy can be further improved when it is used in conjunction with other techniques.	➢ Coding is simple ➢ Robust algorithm ➢ One of the fastest evolutionary algorithm	➢ Highly dependable on the parameters involved in algorithm ➢ Difficulty in the fine-tuning of control parameters ➢ Difficult to handle functions that are not linearly separable	➢ Industrial Control ➢ Designing of Antenna ➢ In the Power System ➢ Processing of Images ➢ 3D tracking of license plates from video ➢ Color image quantization ➢ Power plant control ➢ Optimization in Biological processes ➢ Gene regulatory networks ➢ Data analysis ➢ Protein folding
3.	Clonal Selection Algorithm (CSA) [12,26-29] Proposed by De Castro and Von Zuben in 2002	The accuracy obtained by this algorithm ranges from 65 to 91% depending on the data sets used	➢ Easy to perform and understand ➢ Less time taking	➢ Lack in adaptations ➢ New variant of the algorithm cannot be created, so no improvement can be made ➢ Premature convergence ➢ Unsatisfied accuracy	➢ Network security ➢ Job shop scheduling ➢ Travelling salesman problem ➢ Knapsack problem ➢ Pattern recognition in image processing ➢ Handwriting recognition
NEURAL NETWORK BASED TECHNIQUES					
1.	Recurrent Neural Network (RNN) [30-32] Proposed by Jordan in 1986	The accuracy obtained by this algorithm ranges from 87 to 97%. The accuracy differs on the basis of data sets and other techniques used with this algorithm	➢ Input of any length can be processed ➢ Powerful computation ➢ Dynamic in nature	➢ Computation is slow ➢ Training of algorithm is difficult ➢ Computational complexity is more	➢ Temporal processing models ➢ Voice Recognition ➢ NLP (Natural Language Processing) ➢ Speech Recognition ➢ Time Series Forecasting ➢ Machine Translation ➢ Data Classification ➢ Tracking Water Quality
2.	(Convolutional Neural Network (CNN) [30-32] Proposed byYannLeCun in 1980s		➢ Provides high accuracy ➢ Extract only relevant features from the input ➢ Reuse same parameters	➢ Lots of training is needed ➢ Need large database for computation purpose	➢ Computer Vision ➢ Image Recognition ➢ Image Classification ➢ Analysing Documents ➢ Understanding Climate

Table 2. Comparative analysis of BI techniques (continued)

Sr. No.	Techniques	Accuracy of Results	Merits	Demerits/Limitations	Applications
	SWARM INTELLIGENCE TECHNIQUES				
1.	Ant Colony Optimization (ACO) [13,33,34] Proposed by Marco Dorigo in 1992	The accuracy obtained by this algorithm is about 90% depending on the data sets and other techniques used with this algorithm	➢ Implementation is simple ➢ Can do parallel computation ➢ High probability of finding the optimal solution ➢ Convergence is guaranteed ➢ Short computation time	➢ Convergence time is uncertain ➢ Coding is difficult ➢ Probability distribution keeps on changing with iteration	➢ Travelling Salesman Problem (TSP) ➢ In Scheduling ➢ Problem of Network Model ➢ Routing of Vehicles ➢ Graph Coloring ➢ Quadratic Assignment Problem
2.	Bee Colony Optimization (BCO) [14,35] Proposed by Teodorovic and Dell in 2005	The accuracy obtained by this algorithm is about 72% but can be increased up to 99% when used with other techniques and using different datasets	➢ It makes use of a few control parameters ➢ Convergence is fast ➢ It is self-organizing ➢ Easy to implement ➢ The algorithm is highly flexible ➢ Global search ability is strong	➢ Poor local search ability ➢ The initial solution puts a limit on the search space ➢ Search speed slows down	➢ Benchmarking optimization ➢ Scheduling ➢ Bioinformatics ➢ Clustering and Mining ➢ Image processing ➢ Engineering designs ➢ General assignment problem ➢ Advisory system ➢ Software testing ➢ Structural Optimisation ➢ Numerical assignment problem ➢ Optimisation algorithm development ➢ Face poses estimation ➢ MR brain image classification
3.	Particle Swarm Optimization (PSO) [15,36-38] Proposed by Russell C. Eberhart and James Kennedy in 1995	The accuracy obtained by this algorithm ranges from 80 to 85%. But this accuracy can be increased upto 99% by using the algorithm in hybrid form with other techniques	➢ Simple to implement ➢ Uses only few parameters ➢ Parallel computations can be done ➢ Robust algorithm ➢ High probability for finding global optima ➢ Fast convergence ➢ Short computation time ➢ Efficient for solving problems	➢ Difficult to define the parameters initially ➢ Premature convergence possible and can be trapped in local optima ➢ Cannot be used to solve problems of scattering ➢ Local search ability is weak	➢ Detection and diagnosis of faults, Recovery from faults ➢ Sensors ➢ Metallurgy ➢ Optimization of engines and electrical motors ➢ Computer graphics and visualization ➢ Security and military applications ➢ Finance and economics ➢ Music generation and games

Table 2. Comparative analysis of BI techniques (continued)

Sr. No.	Techniques	Accuracy of Results	Merits	Demerits/Limitations	Applications
4.	Cuckoo Search Algorithm (CSA) [16,39-42] Proposed by Xin-She Yang and Suash Deb in 2009.	The accuracy obtained by the algorithm ranges from 88 to 99% depending on the data sets and other techniques used with this algorithm	➢ The essential parameter is only one, Sa, but it is fixed. So, the algorithm is easy to implement ➢ Simple algorithm ➢ Its main aim is to increase the convergence speed ➢ Able to deal with multiple criteria problems of optimization ➢ Can be used in combination with other swarm based algorithms	➢ Slow rate of Convergence ➢ Can be trapped in local optima ➢ Not suitable for discrete problems	➢ Scheduling ➢ Travelling salesman problem ➢ Flood forecasting ➢ Face recognition ➢ Flow shop scheduling ➢ Surface roughness ➢ Supplier selection ➢ Image Processing ➢ Path planning of logistics vehicles
5.	BAT algorithm (BA) [17,43-45] Proposed by Xin-She Yang in 2010	The accuracy obtained by this algorithm ranges from 95 to 97%. But it can be increased upto 99% by making improvements and using other techniques used with this algorithm	➢ Simple and flexible ➢ Easy to implement ➢ Able to solve highly non-linear problems in an efficient manner ➢ Provides best solution quickly ➢ Can be used to solve complicated problems ➢ Robust algorithm ➢ Provides optimal solutions ➢ Superior ability to deal with a large number of complex issues	➢ Slow rate of convergence ➢ Optimisation Precision is low	➢ Biology and medical problems ➢ Image segmentation ➢ Feature selection ➢ Design of skeletal structures ➢ Path planning ➢ Data classification and clustering ➢ Data mining ➢ Wireless Sensor
6.	Flower Pollination Algorithm (FPA) [18,19,46-48] Proposed by Xin – She Yang in 2012	The accuracy obtained by this algorithm ranges from 78 to 99% and can be increased further depending on the data sets and other techniques used with this algorithm	➢ Used to solve Real-time problems ➢ Can be used for solving diverse problems ➢ Powerful algorithm ➢ It is a simple algorithm ➢ It provides high performance in terms of computations ➢ Outperforms other algorithms	➢ Low convergence rate ➢ Low accuracy	➢ Wireless sensor networking ➢ Graph colouring problems ➢ Image and signal processing ➢ Clustering ➢ Computer gaming ➢ Structural designs

Table 2. Comparative analysis of BI techniques (continued)

Sr. No.	Techniques	Accuracy of Results	Merits	Demerits/Limitations	Applications
7.	Firefly algorithm(FA) [20,21,49-51] Proposed by Xin – She Yang in 2007	The accuracy obtained by this algorithm ranges from 80 to 90% and it differs on the basis of data sets used	➢ Easy to understand ➢ Provide facility of auto segmentation ➢ Able to deal with highly nonlinear as well as multi-modal problems of optimisation in an efficient manner ➢ Very high speed of convergence in finding the globally optimised solution ➢ Can be used with other techniques in hybrid form ➢ It is a simple flexible and versatile algorithm	➢ Exploration capability of Fireflies is low ➢ Optimal solutions are difficult to achieve	➢ Travelling salesman problem (TSP) ➢ Compression of Digital Images ➢ In Image processing ➢ Feature selection ➢ Detection of Faults ➢ Structural design ➢ In Scheduling ➢ Chemical phase equilibrium
8.	Cuttlefish Optimization Algorithm (CFA) [22,52-54] Proposed by Adnan Mohsin Abdulazeez Brifcane in 2014	The accuracy obtained by this algorithm ranges from 92 to 95% depending on the data sets used and the value of parameters initially defined	➢ The feature subset which is obtained using algorithm provides higher detection rate and accuracy rate ➢ The optimal features subset help in reducing the number of features selected ➢ Most stable ➢ Computational time required for training the classifiers is less ➢ Intelligent model that can be used in classifying data under noisy conditions	The main problem of CFA is the random selection of instances and subset of features. This random selection increases the possibility of missing out a set of features as well as instances which may contribute some important information during the classification process	➢ Security ➢ Robotics ➢ Controlling systems ➢ Parallel processing ➢ Mining of Data ➢ Power systems ➢ Computer networks ➢ Production engineering ➢ Biomedical engineering

OBSERVATIONS

The authors have reviewed and analysed different techniques used in implementing Biomimetic Intelligence systems i.e. those systems or models that are biologically inspired. From the comparative analysis, the authors have observed that the *Genetic Algorithm* technique is used for solving TSP (Travelling Salesman Problem) in order to discover the optimized path from source to destination, in Robotics, in air flight schedule planning and so on. *Differential evolution* technique is used for processing images, 3D tracking of license plates, in power systems. These algorithms are simple to understand and implement. These are robust algorithms and makes use of biological operators like crossover, mutation and reproduction. The *Clonal Selection Algorithm* is easy to perform and understand but provides unsatisfied accuracy. *RNN* is used to recognize and classifying images. *CNN* is used in NLP (Natural Language Processing), Speech recognition. These algorithms provide high accuracy. *ACO* algorithm is used in vehicle routing, graph coloring, network model problems. It is simple to implement and parallel computations are possible in this algorithm. But the coding is difficult. *BCO* is used in clustering and mining, image processing, bioinformatics. It is highly flexible techniques and can adapt to changes. *PSO* is used in computer graphics, metallurgy, security and military applications. The computation time is short in this algorithm and the convergence is fast. *Cuckoo Search Algorithm (CSO)* is a simple algorithm and can be implemented easily. But it can be trapped in local optima. The *Bat algorithm* can be used for solving non-linear problems quickly. It has slow rate of convergence and can be used in solving problems such as image segmentation, path planning, data mining. The *FPA* algorithm is one of the powerful algorithms used for solving complex problems such as Communication, clustering. The *firefly algorithm* can also be used in hybrid form with other algorithms. It is used in solving Travelling Salesman Problem, fault detection. The *cuttlefish algorithm* is more stable and takes less time in computation. It can be used in security, robotics and so on.

CONCLUSION

It can be said that the nature has been an inspiration to develop new techniques for solving complex problems, especially optimization problems. The biomimetic process is considered successful if it takes into consideration a deep understanding of the natural system, development of model in simplified form, technical implementation of the final design. The different techniques discussed by the authors have their own disadvantages and advantages and can be used in different real-life applications.

AUTHORSHIP CONTRIBUTIONS

Authors equally contributed to this work.

DATA AVAILABILITY STATEMENT

The authors confirm that the data that supports the findings of this study are available within the article. Raw data that support the finding of this study are available from the corresponding author, upon reasonable request.

CONFLICT OF INTEREST

The author declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

ETHICS

There are no ethical issues with the publication of this manuscript.

REFERENCES

[1] Tan R, Liu W, Cao G, Shi Y. Creative design inspired by biological knowledge: Technologies and methods. Front Mech Eng 2019;14:1−14. [CrossRef]

[2] Rovalo E, McCardle J. Performance based abstraction of biomimicry design principles using prototyping. Designs 2019;3:38. [CrossRef]

[3] Graeff E, Maranzana N, Aoussat A. Biological practices and fields, missing pieces of the biomimetics' methodological puzzle. Biomimetics (Basel) 2020;5:62. [CrossRef]

[4] Hoornaert SG, Lassabe N, Finzinger C, Penalva M, Chirazi J, et al. Nature can inspire solutions for aeronautics and space sciences. Aeronaut Aerosp Open Access J 2020;4:69−79. [CrossRef]

[5] Brodrick D. Natural genius: Approaches and challenges to applying biomimetic design principles in architecture (doctoral thesis). Colorado: Colorado Boulder Univ; 2020.

[6] Wang J, Chen W, Xiao X, Xu Y, Li C, Jia X, et al. A survey of the development of biomimetic intelligence and robotics. Biomimetic Intell Robot 2021;1:100001. [CrossRef]

[7] Chiesa M, Maioli G, Colombo GI, Piacentini L. GARS: Genetic Algorithm for the identification of a Robust Subset of features in high-dimensional datasets. BMC Bioinformatics 2020;21:54. [CrossRef]

[8] Drachal K, Pawlowski M. A review of the applications of Genetic Algorithms to forecasting prices of commodities. Economies 2021;9:6. [CrossRef]

[9] Albadr MA, Tiun S, Ayob M, AL-Dhief F. Genetic Algorithm based on Natural Selection Theory for optimization problems. Symmetry 2020;12:1758. [CrossRef]

[10] Georgioudakis M, Plevris V. A comparative study of Differential Evolution Variants in constrained Structural Optimization. Front Built Environ 2020;6:102. [CrossRef]

[11] Du Y, Fan Y, Liu X, Luo Y, Tang J, Liu P. Multiscale cooperative differential evolution algorithm. Comput Intell Neurosci 2019;2019:5259129. [CrossRef]

[12] Nabil E, Sayed SAF, Hameed HA. An efficient binary clonal selection algorithm with optimum path forest for feature selection. Int J Adv Comput Sci Appl 2020;11. [CrossRef]

[13] Yu J, Li R, Feng Z, Zhao A. A novel parallel ant colony optimization algorithm for warehouse path planning. J Control Sci Eng 2020;2020:5287189. [CrossRef]

[14] Kruekaew B, Kimpan W. Enhancing of artificial bee colony algorithm for virtual machine scheduling and load balancing problem in cloud computing. Int J Comput Intell Syst 2020;13:496−510. [CrossRef]

[15] Hantash N, Khatib T, Khammash M. An improved particle swarm optimization algorithm for optimal allocation of distributed generation units in radial power systems. Appl Comput Intell Soft Comput 2020;2020:8824988. [CrossRef]

[16] Khan A, Khan A, Bangash JI, Subhan F, Khan A, Khan A, et al. Cuckoo Search-based SVM (CS-SVM) model for real time indoor position estimation in IoT networks. Secur Commun Netw 2021;2021:6654926. [CrossRef]

[17] Zebari AY, Almufti SM, Abdulrahman CM. Bat algorithm: Review applications and modifications. Int J Sci World 2020;8:1−7. [CrossRef]
[18] Odili JB, Noraziah A, Babalola AE. Flower pollination algorithm for data generation and analytics - A diagnostic analysis. Sci Afr 2020;8:e00440. [CrossRef]
[19] Yang W, Sun Z. GPS position prediction method based on chaotic map-based flower pollination algorithm. Complexity 2021;2021:9972701. [CrossRef]
[20] Altherwi A. Application of Firefly algorithm for optimal production and demand forecasting at selected industrial plant. Open J Bus Manag 2020;8:2451−2459. [CrossRef]
[21] Nayak J, Naik B, Dinesh P, Vakula K, Dash PB. Firefly Algorithm in biomedical and health care: Advances, issues and challenges. SN Comput Sci 2020;1:311. [CrossRef]
[22] EESA AS, Sadiq S, Hassan MM, Orman Z. Rule generation based on modified cuttlefish algorithm for intrusion detection system. Uludağ Univ J Fac Eng 2021;26:253−268. [CrossRef]
[23] Ma J, Liu Y, Zang S, Wang L. Robot path planning based on Genetic Algorithm fused with continuous Bezier Optimization. Comput Intell Neurosci 2020;2020:9813040. [CrossRef]
[24] Mashwani WK, Rehman ZU, Bakar MA, Kocak I. A customized differential evolutionary algorithm for bounded constrained optimization problems. Complexity 2021;2021;5515701. [CrossRef]
[25] Telleria MC, Zulueta E, Gamiz UF, Betoño DTF, Betoño ATF. Differential evolution optimal parameters tuning with artificial neural network. Mathematics 2021;9:427. [CrossRef]
[26] Yang C, Chen BQ, Jia L, Wen HY. Improved clonal selection algorithm based on biological forgetting mechanism. Complexity 2020;2020:2807056. [CrossRef]
[27] Zhang W, Gao K, Zhang W, Wang X, Zhang Q, Wang H. A hybrid clonal selection algorithm with modified combinatorial recombination and success history based adaptive mutation for numerical optimization. Artif Intell 2019;49:819−836. [CrossRef]
[28] Purbasari A, Hidayanto AN, Zulianto A. Parallelization clonal selection algorithm with MPI. NET for optimization problem. Int J Innov Technol Explor Eng 2019;8:184−191.
[29] Zhang W, Zhang W, Yen G, Jing H. A cluster-based clonal selection algorithm for Optimisation in dynamic environment. Swarm Evol Comput 2019;50:100454. [CrossRef]
[30] Barisik T, Guneri AF. BLEVE risk effect estimation using the Levenberg-Marquardt algorithm in an artificial neural network model. Sigma J Eng Nat Sci 2022;40:877−893. [CrossRef]
[31] Gutiérrez-Muñoz M, González-Salazar A, Coto-Jiménez M. Evaluation of mixed deep neural networks for reverberant speech enhancement. Biomimetics (Basel) 2019;5:1. [CrossRef]
[32] Kaya N, Erdem F. Modeling of copper removal from aqueous solutions by using carbon-based adsorbents derived from hazelnut and walnut shells by artificial neural network. Sigma J Eng Nat Sci 2022;40:695−704. [CrossRef]
[33] Fahad LG, Tahir SF, Shahzad W, Hassan M. Ant colony optimization-based streaming feature selection: An application to the medical image diagnosis. Sci Program 2020;9:1064934. [CrossRef]
[34] Li X, Wang L. Application of improved ant colony optimization in mobile robot trajectory planning. Math Biosci Eng 2020;17:6756−6774. [CrossRef]
[35] Wang H, Wang W, Zhou X, Zhao J, Wang Y, Xiao S, et al. Artificial bee colony algorithm based on knowledge fusion. Complex Intell Syst 2021;7:1139−1152. [CrossRef]
[36] Li X, Wu D, He J, Bashir M, Liping M. An improved method of particle swarm optimization for path planning of mobile robot. J Control Sci Eng 2020;2020:3857894. [CrossRef]
[37] Digehsara AP, Chegini NS, Bagheri A, Roknsaraei PM. An improved particle swarm optimization based on the reinforcement of the population initialization phase by scrambled halt on sequence. Cogent Eng 2020;7:1737283. [CrossRef]
[38] Yu H, Gao Y, Wang L, Meng J. A hybrid particle swarm optimization algorithm enhanced with nonlinear inertial weight and guassian mutation for job shop scheduling problems. Mathematics 2020;8:1355. [CrossRef]
[39] Guerrero-Luis M, Valdez F, Castillo O. A Review on the Cuckoo Search Algorithm. In: Castillo O, Melin P, editors. Fuzzy logic hybrid extensions of neural and optimization algorithms: Theory and applications, studies in computational intelligence. New York: Springer; 2021. p. 113−124. [CrossRef]
[40] Wu Y, Wang Z. Improved Cuckoo search algorithm for multiple odor sources localization. In the Proceedings of the 13th International Conference on Agents and Artificial Intelligence; 2021 Feb 4-6; Vienna, Austria: ICAART; 2021. p. 708−715. [CrossRef]
[41] Manar A, Al-Abaji MA. Cuckoo search algorithm: Review and its applications. Tikrit J Pure Sci 2021;26:137−144. [CrossRef]
[42] Pan JS, Song PC, Chu SC, Peng YJ. Improved compact Cuckoo search algorithm applied to location of drone Logistics hub. Mathematics 2020;8:333. [CrossRef]
[43] Huang J, Ma Y. Bat Algorithm based on an Integration Strategy and Gaussian distribution. Math Probl Eng 2020;2020:9495281. [CrossRef]
[44] Mashwani WK, Mehmood I, Bakar MA, Koçcak I. A modified Bat Algorithm for Solving large-scale bound constrained global optimization problems. Math Probl Eng 2021;2021:6636918. [CrossRef]

[45] Karakonstantis I, Vlachos A. Bat algorithm applied to continuous constrained Optimization problems. J Inf Optim Sci 2021;42:57−75. [CrossRef]
[46] Khursheed M, Nadeem MF, Khalil A, Sajjad IA, Raza A, Iqbal MQ, et al. Review of flower pollination algorithm: Applications and Variants. In Proceedings of the 2020 International Conference on Engineering and Emerging Technologies; 2020 Feb 22-23; Lahore, Pakistan: ICEET; 2020. [CrossRef]
[47] Suwannarongsri S. Application of modified flower pollination algorithm to multiple vehicle routing problems with time constraints. WSEAS Trans Syst Control 2020;15:459−467. [CrossRef]
[48] Wang K, Li X, Gao L. A multi-objective discrete flower pollination algorithm for stochastic two-sided partial disassembly line balancing problem. Comput Ind Eng 2019;130:634−649. [CrossRef]
[49] Wu J, Wang Y, Burrage K, Tian Y, Lawson BA, Ding Z. An improved firefly algorithm for global continuous optimization problems. Expert Syst Appl 2020;149:113340. [CrossRef]
[50] Dai W, Liang L, Zhang B. Firefly optimization algorithm for the prediction of uplift due to high-pressure jet grouting. Adv Civ Eng 2020;2020:8833784. [CrossRef]
[51] Marie-Sainte SL, Alalyani N. Firefly algorithm based feature selection for Arabic text classification. J King Saud Univ Comput Inf Sci 2020;32:320−328. [CrossRef]
[52] Daweri MSA, Abdullah S, Ariffin KAZ. A migration based cuttlefish algorithm with short term memory for optimization problems. IEEE Access 2020;8:70270−70292. [CrossRef]
[53] Hussien AM, Mekhamer SF, Hasanien HM. Cuttlefish Optimisation algorithm based optimal PI controller for performance enhancement of an autonomous operation of a DG system. In Proceedings of the 2nd International Conference on Smart Power and Internet Energy Systems; 2020 Sept 15-18; Bangkok, Thailand; 2020. p. 293−298. [CrossRef]
[54] Eesa AS, Orman Z. A new clustering method based on the bio-inspired cuttlefish optimization algorithm. Expert Syst 2020;37:e12478. [CrossRef]
[55] Liang S, Song B, Xue D. Landing route planning method for micro drones based on hybrid optimization algorithm. Biomimetic Intell Robot 2021;1:100003. [CrossRef]
[56] Aracri S, Giorgio-Serchi F, Suaria G, Sayed ME, Nemitz MP, Mahon S, et al. Soft robots for ocean exploration and offshore operations: A perspective. Soft Robot 2021;8:625−639. [CrossRef]
[57] Bhowmik P, Pantho MJH, Bobda C. Bio-inspired smart vision sensor: Toward a reconfigurable hardware modeling of the hierarchical processing in the brain. J Real Time Image Proc 2021;18:157−174. [CrossRef]
[58] Drotman D, Jadhav S, Sharp D, Chan C, Tolley MT. Electronics-free pneumatic circuits for controlling soft-legged robots. Sci Robot 2021;6:eaay2627. [CrossRef]
[59] Ivanova N. Biomimetic optics: Liquid-based optical elements imitating the eye functionality. Philos Trans A Math Phys Eng Sci 2020;378:20190442. [CrossRef]
[60] Ren Z, Hu W, Dong X, Sitti M. Multi-functional soft-bodied jellyfish-like swimming. Nat Commun 2019;10:2703. [CrossRef]
[61] Le LM, Ly HB, Pham BT, Le VM, Pham TA, Nguyen DH, et al. Hybrid artificial ıntelligence approaches for predicting buckling damage of steel columns under axial compression. Materials (Basel) 2019;12:1670. [CrossRef]
[62] Orozco-Rosas U, Picos K, Montiel O. Hybrid path planning algorithm based on membrane pseudo-bacterial potential field for autonomous mobile robots. IEEE Access 2019;7:156787−156803. [CrossRef]
[63] Lin T, Lin H, Lin C, Chen C. Mobile robot wall-following control using a behavior-based fuzzy controller in unknown environments. Iran J Fuzzy Syst 2019;16:113−124.
[64] Martin-Palma JR, Kolle M. Biomimetic photonic structures for optical sensing. Opt Laser Technol 2019;109:270−277. [CrossRef]
[65] Hancock M, Sood S, Higgins T, Sproul K. Biomimicry and Machine Learning in the Context of Healthcare Digitization. In: Schmorrow DD, Fidopiastis CM, editors. Augment cognition. New York: Springer; 2019. p. 273−283. [CrossRef]
[66] Lamata L. Quantum machine learning and quantum biomimetics: A perspective. Mach Learn Sci Technol 2020;1:033002. [CrossRef]
[67] Wanieck K, Ritzinger D, Zollfrank C, Jacobs S. Biomimetics: Teaching the tools of the trade. FEBS Open Bio 2020;10:2250−2267. [CrossRef]
[68] Di Salvo S. Advances in research for biomimetic materials. Adv Mater Res 2018;1149:28−40. [CrossRef]
[69] Ababsa T, Djedl N, Duthen Y. Genetic programming-based self-reconfiguration planning for metamorphic robot. Int J Autom Comput 2018;15:431−442. [CrossRef]
[70] Blanco E, Uchiyama Y, Kohsaka R. Application of biomimetics to architectural and urban design: A review across scales. Sustainability 2020;12:9813. [CrossRef]
[71] Ha N, Lu G. A review of recent research on bio-inspired structured and materials for energy absorption applications. Compos Part B Eng 2020;181:107496. [CrossRef]
[72] Sreenivasulu R, Chaitanya G. Self adaptive penalty method coupled with metaheuristic algorithms to optimization of varying geometrical parameters in drilling for multi hole parts. Sigma J Eng Nat Sci 2022;40:855−867. [CrossRef]

[73] Izci D. A novel modified arithmetic optimization algorithm for power system stabilizer design. Sigma J Eng Nat Sci 2022;40:529–541. [CrossRef]

[74] Ahmad AA, Sari M. Parameter estimation to an anemia model using the Particle Swarm Optimization. Sigma J Eng Nat Sci 2019;37:1335–1347.