

# A Survey of Bio inspired Optimization Algorithms

Binitha S, S Siva Sathya

**Abstract**—Nature is of course a great and immense source of inspiration for solving hard and complex problems in computer science since it exhibits extremely diverse, dynamic, robust, complex and fascinating phenomenon. It always finds the optimal solution to solve its problem maintaining perfect balance among its components. This is the thrust behind bio inspired computing. Nature inspired algorithms are meta heuristics that mimics the nature for solving optimization problems opening a new era in computation .For the past decades ,numerous research efforts has been concentrated in this particular area. Still being young and the results being very amazing, broadens the scope and viability of Bio Inspired Algorithms (BIAs) exploring new areas of application and more opportunities in computing. This paper presents a broad overview of biologically inspired optimization algorithms, grouped by the biological field that inspired each and the areas where these algorithms have been most successfully applied.

**Index Terms**— Bio Inspired Algorithm, Optimization algorithms

## I. INTRODUCTION

Optimization is a commonly encountered mathematical problem in all engineering disciplines. It literally means finding the best possible/desirable solution. Optimization problems are wide ranging and numerous, hence methods for solving these problems ought to be, an active research topic. Optimization algorithms can be either deterministic or stochastic in nature. Former methods to solve optimization problems require enormous computational efforts, which tend to fail as the problem size increases. This is the motivation for employing bio inspired stochastic optimization algorithms as computationally efficient alternatives to deterministic approach. Meta-heuristics are based on the iterative improvement of either a population of solutions (as in Evolutionary algorithms, Swarm based algorithms) or a single solution (eg. Tabu Search) and mostly employ randomization and local search to solve a given optimization problem.

### *Towards technology through Nature*

The real beauty of nature inspired algorithms lies in the fact that it receives its sole inspiration from nature. They have the ability to describe and resolve complex relationships from intrinsically very simple initial conditions and rules with little or no knowledge of the search space Nature is the perfect

example for optimization, because if we closely examine each and every features or phenomenon in nature it always find the optimal strategy, still addressing complex interaction among organisms ranging from microorganism to fully fledged human beings, balancing the ecosystem, maintaining diversity, adaptation, physical phenomenon like river formation, forest fire ,cloud, rain etc..Even though the strategy behind the solution is simple the results are amazing. Nature is the best teacher and its designs and capabilities are extremely enormous and mysterious that researchers are trying to mimic nature in technology. Also the two fields have a much stronger connection since, it seems entirely reasonable that new or persistent problems in computer science could have a lot in common with problems nature has encountered and resolved long ago. Thus an easy mapping is possible between nature and technology. Bio inspired computing has come up as a new era in computing encompassing a wide range of applications, covering all most all areas including computer networks, security, robotics, bio medical engineering, control systems ,parallel processing ,data mining, power systems, production engineering and many more.

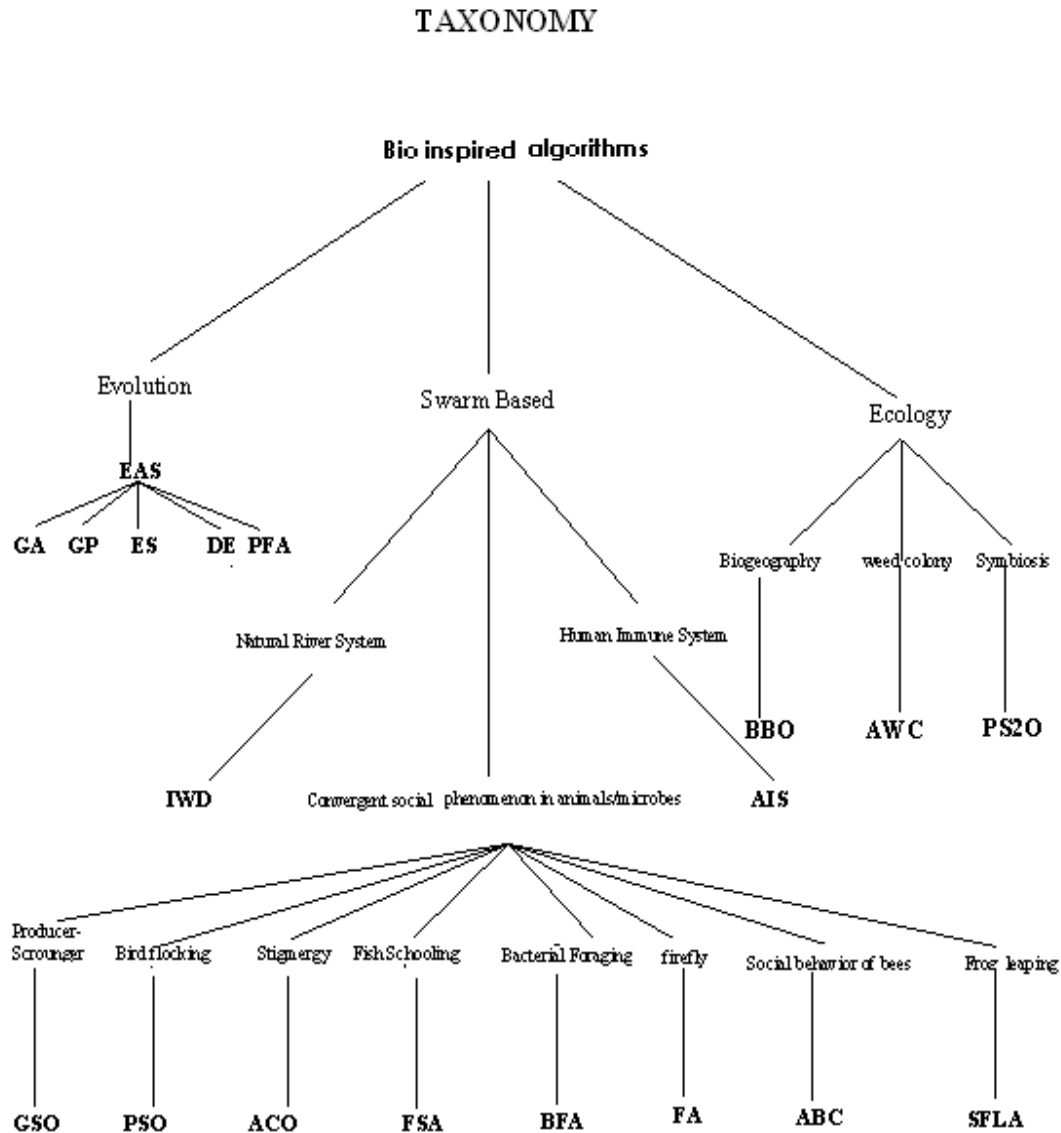
Classical problem solving methodologies involve two branches: Exact methods (logical, mathematical programming) and Heuristics. Heuristic approach seems to be superior in solving hard and complex optimization problems, particularly where the traditional methods fail. BIAs are such heuristics that mimics /imitate the strategy of nature since many biological processes can be thought of as processes of constrained optimization.They make use of many random decisions which classifies them as a special class of randomized algorithms. Formulating a design for bioinspired algorithms involves choosing a proper representation of problem, evaluating the quality of solution using a fitness function and defining operators so as to produce a new set of solutions.

A vast literature exists on bio inspired approaches for solving an impressive array of problems and, more recently, a number of studies have reported on the success of such techniques for solving difficult problems in all key areas of computer science.The two most predominant and successful classes or directions in BIAs involves Evolutionary Algorithms and Swarm based Algorithms which are inspired by the natural evolution and collective behavior in animals respectively. But still, this has been further refined so as to classify the algorithms based on the area of inspiration from nature so as to enhance a broader view over the domain. This paper presents a survey and review on the BIAs along with taxonomy and the relevant application areas.

**Manuscript received March 28, 2012**

**Binitha S**, Department of Computer Science , Pondicherry University, Pondicherry, India, 91 9483079670, (e-mail: binithacec@gmail.com).

**Dr. S Siva Sathya**, Department of Computer Science, Pondicherry University, Pondicherry, India, (e-mail: ssvivasathya@gmail.com).



*Fig: Taxonomy and nomenclature of various bio inspired optimization algorithms grouped by the area of inspiration*

The organization of the paper is as follows: Section 2 provides an overview of EAs. The new and emerging algorithms of SI family are discussed in Section 3. Section 4 introduces algorithms inspired from natural ecosystem. Section 5 sketches a tabular form summary of all the algorithms. The conclusion is drawn in Section 6.

## II. EVOLUTIONARY ALGORITHMS

Evolutionary computation (EC)[1] is a paradigm in the artificial intelligence realm that aims at benefiting from collective phenomena in adaptive populations of problem solvers utilizing the iterative progress comprising growth ,development, reproduction, selection, and survival as seen in a population . EAs are the most well known, classical and

established algorithms among nature inspired algorithms, which is based on the biological evolution in nature that is being responsible for the design of all living beings on earth, and for the strategies they use to interact with each other. EAs employ this powerful design philosophy to find solutions to hard problems. EAs are non-deterministic algorithms or cost based optimization algorithms

A family of successful EAs comprises genetic algorithm (GA), genetic programming (GP), Differential Evolution, evolutionary strategy (ES) and most recent Paddy Field Algorithm. The members of the EA family share a great number of features in common. They are all population-based stochastic search algorithms performing with best-to-survive criteria. Each algorithm commences by creating an initial

population of feasible solutions, and evolves iteratively from generation to generation towards a best solution. In successive iterations of the algorithm, fitness-based selection takes place within the population of solutions. Better solutions are preferentially selected for survival into the next generation of solutions.

#### A. Genetic Algorithm

GA is an evolutionary based stochastic optimization algorithm with a global search potential proposed by Holland in 1975[2]. GAs are among the most successful class of algorithms under EAs which are inspired by the evolutionary ideas of natural selection. They follow the principles of Charles Darwin Theory of survival of the fittest. However, because of its outstanding performance in optimization, GA has been regarded as a function optimizer. Algorithm begins by initializing a population of solution (chromosome). It comprises representation of the problem usually in the form of a bit vector. Then for each chromosome evaluate the fitness using an appropriate fitness function suitable for the problem. Based on this, the best chromosomes are selected into the mating pool, where they undergo cross over and mutation thus giving new set of solutions(offspring).

The three principal genetic operators in GA involve selection, crossover, and mutation.

GA is useful and efficient when:

- The search space is large complex or poorly known.
- No mathematical analysis is available.
- Domain knowledge is scarce to encode to narrow the search space
- For complex or loosely defined problems since it works by its own internal rules.
- Traditional search method fails.

Even though GAs can rapidly locate good solutions, for difficult search spaces, it has some disadvantages:

- 1) GA may have a tendency to converge towards local optima rather than the global optimum of the problem if the fitness function is not defined properly.
- 2) Operating on dynamic data sets is difficult.
- 3) For specific optimization problems, and given the same amount of computation time, simpler optimization algorithms may find better solutions than GAs.
- 4) GAs are not directly suitable for solving constraint optimization problems

#### B. Genetic Programming

Proposed by Koza in 1992[3], GP being an extension to Genetic algorithms differs from the latter in terms of representation of the solution. GP represent an indirect encoding of a potential solution (in the form of a tree), in which search is applied to the solution directly, and a solution could be a computer program. The second fundamental difference is in the variable-length representation adopted by GP in contrast with the fixed length encoding in GA. The population in GP generates diversity not only in the values of the genes but also in the structure of the individuals.

Hence GP resembles evolution of a population of computer programs. The four steps in Genetic programming involves:

- 1) Generate an initial population of computer programs comprising the functions and terminals.

- 2) Execute each program in the population and assign it a fitness value according to how well it solves the problem.
- 3) Create a new population of computer programs.
  - i) Copy the best existing programs
  - ii) Create new computer programs by mutation.
  - iii) Create new computer programs by crossover (sexual reproduction).

#### C. Evolution Strategies

Evolution Strategies was developed by three students (Bierert, Rechenberg, Schwefel) at the Technical University in Berlin in 1964[4] in an effort to robotically optimize an aerodynamic design problem. Evolution Strategies is a global optimization algorithm inspired by the theory of adaptation and evolution by means of natural selection. Specifically, the technique is inspired by macro-level or the species-level process of evolution (phenotype, hereditary, variation) unlike genetic algorithm which deals with micro or genomic level (genome, chromosomes, genes, alleles). A very important feature of ES is the utilization of self-adaptive mechanisms for controlling the application of mutation. These mechanisms are aimed at optimizing the progress of the search by evolving not only the solutions for the problem being considered, but also some parameters for mutating these solutions.

Some common Selection and Sampling schemes in ES are as follows:

**(1+1)-ES:** This is a simple selection mechanism in which works by creating one real-valued vector of object variables from its parent by applying mutation with an identical standard deviation to each object variable. Then, the resulting individual is evaluated and compared to its parent, and the better survives to become a parent of the next generation, while the other is discarded.

**( $\mu + \lambda$ )-ES:** Here  $\mu$  parents are selected from the current generation and generate  $\lambda$  offspring, through some recombination and/or mutation operators. Out of the union of parents and offspring ( $\mu + \lambda$ ), the best  $\mu$  kept for next generation. It inherently incorporates elitism.

**( $\mu, \lambda$ )-ES:** Currently used variant is ( $\mu, \lambda$ )-ES. Here  $\mu$  parents selected from the current generation and used to generate  $\lambda$  offspring (with  $\lambda \geq \mu$ ) and only the best  $\mu$  offspring individuals form the next generation discarding the parents completely. This does not incorporate elitism.

#### D. Differential Evolution

Another paradigm in EA family is differential evolution (DE) proposed by Storn and Price in 1995[5]. DE is similar to GAs since populations of individuals are used to search for an optimal solution. The main difference between GAs and DE is that, in GAs, mutation is the result of small perturbations to the genes of an individual while in DE mutation is the result of arithmetic combinations of individuals. At the beginning of the evolution process, the mutation operator of DE favors exploration. As evolution progresses, the mutation operator favors exploitation. Hence, DE automatically adapts the mutation increments to the best value based on the stage of the evolutionary process. Mutation in DE is therefore not based on a predefined probability density function.

Advantages:

- DE is easy to implement, requires little parameter tuning
- Exhibits fast convergence
- It is generally considered as a reliable, accurate, robust and fast optimization technique.

Limitations:

- According to Krink et al. (2004), noise may adversely affect the performance of DE due to its greedy nature.
- Also the user has to find the best values for the problem-dependent control parameters used in DE and this is a time consuming task.

A self-adaptive DE (SDE) algorithm can eliminate the need for manual tuning of control parameters

### E. Paddy Field Algorithm

Recent algorithm Proposed by **Premaratne et al** in 2009 [6], which operates on a reproductive principle dependant on proximity to the global solution and population density similar to plant populations. Unlike evolutionary algorithms, it does not involve combined behavior nor crossover between individuals instead it uses pollination and dispersal. PFA constitutes five basic steps.

1. Sowing: The algorithm operates by initially scattering seeds (initial population  $p_0$ ) at random in an uneven field.
2. Selection: Here the best plants are selected based on a threshold method so as to selectively weed out unfavorable solutions and also controls the population.
3. Seeding: In this stage each plant develops a number of seeds proportional to its health. The seeds that drop into the most favorable places (most fertile soil, best drainage, soil moisture etc.) tend to grow to be the best plants (taller) and produce more number of seeds. The highest plant of the population would correspond to the location of the optimum conditions and the plant's fitness is determined by a fitness function.
4. Pollination: For seed propagation pollination is a major factor either via animals or through wind. High population density would increase the chance of pollination for pollen carried by the wind
5. Dispersion: In order to prevent getting stuck in local minima, the seeds of each plant are dispersed. Depending on the status of the land it will grow into new plants and continue the cycle.

As per no free lunch rule, the PFA only has a lower computational cost. Since the PFA doesn't have crossover, the optimum solution can be migrated to reach the optimum solution.

## III. SWARM INTELLIGENCE

Swarm Intelligence (Kennedy and Eberhart, 2001[7]) is a recent and emerging paradigm in bio inspired computing for implementing adaptive systems. In this sense, it is an extension of EC. While EAs are based on **genetic adaptation** of organisms SI is based on collective **social behavior** of organisms. As per definitions in literature, Swarm Intelligent encompasses the implementation of collective intelligence of groups of simple agents that are based on the behavior of real world insect swarms, as a problem solving tool. The word

“swarm” comes from the irregular movements of the particles in the problem space. SI has been developed alongside with EAs. Some most well-known strategies in this area are discussed below. These trajectory tracking algorithms being inspired by the collective behavior of animals, exhibit decentralized, self-organized patterns in the foraging process.

**Swarm Intelligence Principles:** SI can be described by considering five fundamental principles.

- 1) *Proximity Principle:* the population should be able to carry out simple space and time computations.
- 2) *Quality Principle:* the population should be able to respond to quality factors in the environment.
- 3) *Diverse Response Principle:* the population should not commit its activity along excessively narrow channels.
- 4) *Stability Principle:* the population should not change its mode of behavior every time the environment changes.
- 5) *Adaptability Principle:* the population should be able to change its behavior mode when it is worth the computational price

### A. Particle Swarm Optimization

Particle swarm optimization (PSO) is a computational intelligence oriented, stochastic, population-based global optimization technique proposed by Kennedy and Eberhart in 1995[8]. It is inspired by the social behavior of bird flocking searching for food. PSO has been extensively applied to many engineering optimization areas due to its unique searching mechanism, simple concept, computational efficiency, and easy implementation. In PSO, the term “particles” refers to population members which are mass-less and volume-less (or with an arbitrarily small mass or volume) and are subject to velocities and accelerations towards a better mode of behavior. Each particle in the swarm represents a solution in a high-dimensional space with four vectors, its current position, best position found so far, the best position found by its neighborhood so far and its velocity and adjusts its position in the search space based on the best position reached by itself (pbest) and on the best position reached by its neighborhood (gbest) during the search process. In each iteration, each particle updates its position and velocity as follows:

$$x_{k+1}^i = x_k^i + v_{k+1}^i$$

$$v_{k+1}^i = v_k^i + c_1 r_1 (p_k^i - x_k^i) + c_2 r_2 (p_k^g - x_k^i)$$

where,  $x_k^i$  represents Particle position

$v_k^i$  represents Particle velocity

$p_k^i$  represents Best "remembered" position

$c_1, c_2$  represents cognitive and social parameters,

$r_1, r_2$  are random numbers between 0 and 1

**Steps in PSO algorithm can be briefed as below:**

- 1) Initialize the swarm by assigning a random position in the problem space to each particle.
- 2) Evaluate the fitness function for each particle.
- 3) For each individual particle, compare the particle's fitness value with its pbest. If the current value is better than



the pbest value, then set this value as the pbest and the current particle's position,  $x_i$ , as  $p_i$ .

4) Identify the particle that has the best fitness value. The value of its fitness function is identified as  $g_{best}$  and its position as  $p_{g_{best}}$ .

5) Update the velocities and positions of all the particles using (1) and (2).

6) Repeat steps 2–5 until a stopping criterion is met (e.g., maximum number of iterations or a sufficiently good fitness value).

#### Advantages over Genetic Algorithm:

(a) PSO is easier to implement and there are fewer parameters to adjust.

(b) PSO has a more effective memory capability than GA.

(c) PSO is more efficient in maintaining the diversity of the swarm, since all the particles use the information related to the most successful particle in order to improve themselves, whereas in Genetic algorithm, the worse solutions are discarded and only the new ones are saved; i.e. in GA the population evolve around a subset of the best individuals.

There are many similarities between the PSO and EAs. Both of them initialize solutions and update generations, while the PSO has no evolution operators as does the latter. In a PSO, particles try to reach the optimum by following the current global optimum instead of using evolutionary operators, such as mutation and crossover.

It is claimed that the PSO, in addition to continuous functions, has been showing stability and convergence in a multidimensional complex space also. (Clerc and Kennedy, 2002).

#### B. Ant Colony Optimization

ACO is among the most successful swarm based algorithms proposed by Dorigo & Di Caro in 1999 [9]. It is a meta heuristic inspired by the foraging behavior of ants in the wild, and moreover, the phenomena known as stigmergy, term introduced by Grasse in 1959. Stigmergy refers to the indirect communication amongst a self-organizing emergent system via individuals modifying their local environment. The most interesting aspect of the collaborative behavior of several ant species is their ability to find shortest paths between the ants' nest and the food sources by tracing pheromone trails. Then, ants choose the path to follow by a probabilistic decision biased by the amount of pheromone: the stronger the pheromone trail, the higher its desirability. Because ants in turn deposit pheromone on the path they are following, this behavior results in a *self-reinforcing process* leading to the formation of paths marked by high pheromone concentration. By modeling and simulating ant foraging behavior, brood sorting, nest building and self-assembling, etc. algorithms can be developed that could be used for complex, combinatorial optimization problems.

The first ant algorithm, named "Ant System" (AS), was developed in the nineties by Dorigo et al. (1996) and tested successfully on the well known benchmark Travelling Salesman Problem. The ACO meta heuristic was developed (Dorigo & Di Caro, 1999;) to generalize, the overall method of solving combinatorial problems by approximate solutions

based on the generic behavior of natural ants. ACO is structured into three main functions as follows:

**AntSolutionsConstruct:** This function performs the solution construction process where the artificial ants move through adjacent states of a problem according to a transition rule, iteratively building solutions.

**Pheromone Update:** performs pheromone trail updates. This may involve updating the pheromone trails once complete solutions have been built, or updating after each iteration.

In addition to pheromone trail reinforcement, ACO also includes pheromone trail evaporation. Evaporation of the pheromone trails helps ants to 'forget' bad solutions that were learned early in the algorithm run.

**DeamonActions:** is an optional step in the algorithm which involves applying additional updates from a global perspective (for this no natural counterpart exists). This may include applying additional pheromone reinforcement to the best solution generated (known as offline pheromone trail update).

An alternative approach, called the ant colony system (ACS) has been introduced by Dorigo and Gambardella (1997) to improve the performance of ant system. It is based on four modifications of ant system: a different transition rule, a different pheromone trail update rule, the use of local updates of pheromone trail to favor exploration, and the use of candidate list to restrict the choice.

#### C. Artificial Bee Colony Algorithm (ABC)

Based on the behavior of the bees in nature, various swarm intelligence algorithms are available. These algorithms are classified into two; foraging behavior and mating behavior. Examples of algorithms simulating the foraging behavior of the bees include the Artificial Bee Colony (ABC), the Virtual Bee algorithm proposed by Yang, the Bee Colony Optimization algorithm proposed by Teodorovic and Dell'Orco, the BeeHive algorithm proposed by Wedde et al., the Bee Swarm Optimization algorithm proposed by Drias et al. and the Bees algorithm proposed by Pham et al. An individual entity (e.g., a bee in a bee colony) exhibit a simple set of behavior policies (e.g., migration, replication, death), but a group of entities (e.g., a bee colony) shows complex emergent behavior with useful properties such as scalability and adaptability.

Artificial Bee Colony is a predominant algorithm simulating the intelligent foraging behavior of a honeybee swarm, proposed by Karaboga and Basturk [10]. In ABC algorithm, the colony of artificial bees contains three groups of bees: *employed bees, onlookers and scouts*.

A bee waiting on the dance area for making a decision to choose a food source is called onlooker and one going to the food source visited by it before is named employed bee. The other kind of bee is scout bee that carries out random search for discovering new sources. The position of a food source represents a possible solution to the optimization problem and the nectar amount of a food source corresponds to the quality (fitness) of the associated solution. A swarm of virtual bees is generated and started to move randomly in two-dimensional search space. Bees interact when they find some target nectar

and the solution of the problem is obtained from the intensity of these bee interactions.

A randomly distributed initial population solutions ( $x_i=1,2,...D$ ) is being spread over the D dimensional problem space. An employed bee produces a modification on the position (solution) in her memory depending on the local information (visual information) and tests the nectar amount (fitness value) of the new source (new solution). Provided that the nectar amount of the new one is higher than that of the previous one, the bee memorizes the new position and forgets the old one. After all employed bees complete the search process; they share the nectar information of the food sources and their position information with the onlooker bees on the dance area.

In the next phase **Reproduction**, based on the probability value associated with the food source,  $P_i$ , the artificial onlooker bee chooses a food source

$$P_i = \frac{fit_i}{\sum_{n=1}^N fit_n}$$

Where, N is the number of food sources (that is the number of employed bees),  $fit_i$  is the fitness value of the solution  $i$  which is proportional to the nectar amount of the food source in the position  $i$ .

In the last phase, **Replacement of bee and Selection**, if a position can not be improved further through a predetermined number of cycles, then that food source is assumed to be abandoned. The value of predetermined number of cycles is an important control parameter of the ABC algorithm, which is called “*limit*” for abandonment. After each candidate source position is produced and then evaluated by the artificial bee, its performance is compared with that of its old one. If the new food has an equal or better nectar than the old source, it is replaces the old one in the memory. Otherwise, the old one is retained in the memory.

The local search performance of ABC algorithm depends on neighborhood search and greedy selection mechanisms performed by employed and onlooker bees. The global search performance of the algorithm depends on random search process performed by scouts and neighbor solution production mechanism performed by employed and onlooker bees

#### D. Fish Swarm Algorithm

The fish swarm algorithm (FSA) is a new population-based/swarm intelligent evolutionary computation technique proposed by Li et al. [11] in 2002 which is inspired by the natural schooling behavior of fish. FSA presents a strong ability to avoid local minimums in order to achieve global optimization. A fish is represented by its D-dimensional position  $X_i = (x_1, x_2, \dots, x_k, \dots, x_D)$ , and food satisfaction for the fish is represented as  $FS_i$ . The relationship between two fish is denoted by their Euclidean distance  $d_{ij} = \|X_i - X_j\|$ .

FSA imitates three typical behaviors, defined as “searching for food”, “swarming in response to a threat”, and “following to increase the chance of achieving a successful result”.

**Searching** is a random search adopted by fish in search of food, with a tendency towards food concentration. The objective is to minimize FS (food satisfaction).

**Swarming:** aims in satisfying food intake needs, entertaining swarm members and attracting new swarm members. A fish located at  $X_i$  has neighbors within its visual.  $X_c$  identifies the center position of those neighbors and is used to describe the attributes of the entire neighboring swarm. If the swarm center has greater concentration of food than is available at the fish’s current position  $X_i$  (i.e.,  $FS_c < FS_i$ ), and if the swarm ( $X_c$ ) is not overly crowded ( $ns/n < \delta$ ), the fish will move from  $X_i$  to next  $X_{i+1}$ , toward  $X_c$ .

**Following** behavior implies when a fish locates food, neighboring individuals follow. Within a fish’s visual, certain fish will be perceived as finding a greater amount of food than others, and this fish will naturally try to follow the best one ( $X_{min}$ ) in order to increase satisfaction (i.e., gain relatively more food [ $FS_{min} < FS_i$ ] and less crowding [ $nf/n < \delta$ ]).  $nf$  represents number of fish within the visual of  $X_{min}$ .

Three major **parameters** involved in FSA include visual distance (visual), maximum step length (step), and a crowd factor. FSA effectiveness seems primarily influenced by the former two (visual and step).

#### E. Intelligent Water Drops Algorithm (IWD)

IWD is an innovative population based method proposed by Hamed Shah-hosseini in 2007[12]. It is inspired by the processes in natural river systems constituting the actions and reactions that take place between water drops in the river and the changes that happen in the environment that river is flowing. Based on the observation on the behavior of water drops, an artificial water drop is developed which possesses some of the remarkable properties of the natural water drop. This Intelligent Water Drop has two important properties:

1. The amount of the soil it carries now, Soil (IWD).
2. The velocity that it is moving now, Velocity (IWD).

The environment in which the water flows depend on the problem under consideration. An IWD moves in discrete finite-length steps. From its current location to its next location, the IWD velocity is increased by the amount nonlinearly proportional to the inverse of the soil between the two locations. Moreover, the IWD’s soil is increased by removing some soil of the path joining the two locations. The amount of soil added to the IWD is inversely (and nonlinearly) proportional to the time needed for the IWD to pass from its current location to the next location. This duration of time is calculated by the simple laws of physics for linear motion. Thus, the time taken is proportional to the velocity of the IWD and inversely proportional to the distance between the two locations.

Another property of an IWD is that it prefers the paths with low soils on its beds to the paths with higher soils on its beds. To implement this behavior of path choosing, a uniform random distribution is used among the soils of the available paths such that the probability of the next path to choose is inversely proportional to the soils of the available paths. The lower the soil of the path, the more chance it has for being selected by the IWD.

#### F. Bacterial Foraging Optimization Algorithm

This has been evolving as a new and promising branch in Bio inspired Algorithms that can bridge the gap between microbiology and engineering. These classes of algorithms inherit the characteristics of bacterial foraging patterns such as chemo taxis, metabolism, reproduction and quorum sensing. The complex and organized activities exhibited in bacterial foraging patterns inspire a new approach to solve complex optimization problems.

The Bacterial Foraging Optimization Algorithm (BFO) was introduced by Passino in 2002[13]. Foraging is a phenomenon of a bacterial colony rather than an individual behavior. BFOA consists of three principal mechanisms namely, chemo taxis, reproduction, and elimination-dispersal.

**Chemotaxis** (cell movement) is the activity of bacteria gathering to nutrient-rich areas in a spontaneous fashion; in this context, a cell-to-cell communication mechanism is established to simulate the biological behavior of bacterial movement (swim/tumble).

**Reproduction** comes from the concept of natural selection; under this procedure, only the best adapted bacteria tend to survive and transmit their genetic characters to succeeding generations, while the less adapted ones tend to perish.

**Elimination-dispersal** events randomly select parts of the bacteria population to diminish and disperse into random positions in the environment; this way the algorithm ensures the diversity of the species, and prevents getting trapped to local optima, improving global search ability.

#### G. Artificial Immune System Algorithm

Proposed by Dasgupta, in 1999 [14]. Artificial Immune algorithm is based on clonal selection principle and is a population based algorithm. AIS is inspired by the human immune system which is a highly evolved, parallel and distributed adaptive system that exhibits the following strengths: immune recognition, reinforcement learning, feature extraction, immune memory, diversity and robustness. The artificial immune system (AIS) combines these strengths and has been gaining significant attention due to its powerful adaptive learning and memory capabilities.

The main search power in AIS relies on the mutation operator and hence, the efficiency deciding factor of this technique. The steps in AIS are as follows:

**1. Initialization** of antibodies (potential solutions to the problem). Antigens represent the value of the objective function  $f(x)$  to be optimized.

**2. Cloning**, where the affinity or fitness of each antibody is determined. Based on this fitness the antibodies are cloned; that is the best will be cloned the most. The number of clones generated from the  $n$  selected antibodies is given by:

$$N_c = \sum \text{round}(\beta * j / i) \quad i = 1, 2, \dots, n,$$

Where  $N_c$  is the total number of clones,  $\beta$  is a multiplier factor and  $j$  is the population size of the antibodies.

**3. Hypermutation:** The clones are then subjected to a hyper mutation process in which the clones are mutated in inverse proportion to their affinity; the best antibody's clones are mutated lesser and worst antibody's clones are mutated most. The clones are then evaluated along with their original antibodies out of which the best  $N$  antibodies are selected for

the next iteration. The mutation can be uniform, Gaussian or exponential.

#### H. Firefly algorithm

Firefly algorithm proposed by Yang [15] can be considered as an unconventional swarm-based heuristic algorithm for constrained optimization tasks inspired by the flashing behavior of fireflies. The algorithm constitutes a population-based iterative procedure with numerous agents (perceived as fire flies) concurrently solving a considered optimization problem. Agents communicate with each other via bioluminescent glowing which enables them to explore cost function space more effectively than in standard distributed random search. Intelligence optimization technique is based on the assumption that solution of an optimization problem can be perceived as agent (fire fly) which glows proportionally to its quality in a considered problem setting. Consequently each brighter fire fly attracts its partners (regardless of their sex), which makes the search space being explored more efficiently.

The firefly algorithm has three particular idealized rules which are based on some of the basic flashing characteristics of real fireflies. They are the following:

1) All fireflies are unisex and they will move towards more attractive and brighter ones regardless of their sex.

2) The degree of attractiveness of a firefly is proportional to its brightness. Also the brightness may decrease as the distance from the other fire flies increases due to the fact that the air absorbs light. If there is not a brighter or more attractive fire fly than a particular one it will then move randomly.

3) The brightness or light intensity of a fire fly is determined by the value of the objective function of a given problem.

Advantage:

Mainly uses real random numbers and is based on the global communication among the swarm particles (ie the firefly), hence more effective in multi objective optimization.

#### I. Group search optimizer

The group search optimizer (GSO), was investigated at the University of Liverpool (He et al., 2006)[16]. It is a population based optimization algorithm, which adopts the producer-scrounger (PS) model metaphorically for designing optimum searching strategies, inspired by animal foraging behavior. Similar to the PSO, the population of the GSO is called a **group** and each individual in the population is called a **member**. In the search space, each member knows its own position, its head angle and a head direction, which can be calculated from the head angle via a polar to Cartesian co-ordinate transformation. A group constitutes three types of members: producers, scroungers and rangers.

**Producers:** perform producing strategies, searching for food.

**Scroungers:** perform scrounging strategies, joining resources uncovered by others.



*Rangers*: perform random walk motions and will be dispersed from their current positions.

The producer will find the best point with the best resource (fitness value). If the best point has a better fitness than its current position (in minimization problem as an example), then it will fly to this point. Or it will stay in its current position and turn its head to a new randomly generated angle. If the producer cannot find a better area after a number of iterations, it will turn its head back to zero degree. For scroungers, area copying is adopted, which is the commonest scrounging behavior in sparrows. Random walks are employed by the rangers. If a scrounger (or ranger) finds a better location than the current producer and other scroungers, in the next searching iteration it will switch to be a producer, and all the other members, including the producer in the previous searching iteration, will perform scrounging strategies. It is also assumed that the producer and the scroungers do not differ in their relevant phenotypic characteristics. Therefore, they can switch between the two roles.

### J. Shuffled frog Leaping Algorithm

Proposed by Muzaffar Eusuff and Kevin Lansey in 2003[17]. Shuffled frog-leaping algorithm (SFLA) is a population-based cooperative meta-heuristic algorithm with efficient mathematical function and global search capability. The SFLA is a search metaphor inspired by natural memetics and evolution. It is inspired by the interactive behavior and global exchange of information of frogs searching for food laid on discrete stones randomly located in a pond. It combines the advantages of the genetic-based memetic algorithm (MA) and the social behavior-based PSO algorithm with such characteristics as simple concept, fewer parameters adjustment, prompt formation, great capability in global search and easy implementation.

The steps in SFLA include the following:

- **Initial population**: Individual frogs are equivalent to the GA chromosomes, and represent a set of solutions.
- **Sorting and distribution** : Frogs are sorted in descending order based on their fitness values, then each frog is distributed to a different subset of the whole population called a memplex, the entire population is divided into  $m$  memplexes, each containing  $n$  frogs
- **Memplex evolution**: An independent local search is conducted for each frog memplex, in what is called memplex evolution.
- **Shuffling**: After a defined number of memetic evolutionary steps, frogs are shuffled among memplexes, enabling frogs to interchange messages among different memplexes and ensure that they move to an optimal position, similar to particles in PSO.
- **Terminal condition** : If a global solution or a fixed iteration number is reached, the algorithm stops

## IV. ECOLOGY

Natural ecosystems provides rich source of mechanisms for designing and solving difficult engineering and computer science problems. It comprises the living organisms along with the abiotic environment with which organisms interact

such as air, soil, water etc. There can be numerous and complex types of interactions among the species of ecosystem. Also this can occur as interspecies interaction (between species) or intra species interaction (within species). The nature of these interactions can be cooperative/competitive. Cooperation includes division of labor and represents the core of sociality. Inter species interaction can be of mainly 3 types based on the outcome of interaction (positive, negative, neutral) termed as mutualism, parasitism, commensalism respectively. Examples of cooperation in nature: within species (i.e., homogeneous cooperation, also called social evolution), as in the social foraging behaviors of animal herds, bird flocks, insect groups and bacterial colonies; between species (i.e., heterogeneous cooperation, also called symbiosis), as in the mutualism between human and honey guide. Moreover Biogeography includes the study of distribution of species over specified time and space.

### A. PS20

Proposed by Hanning Chen and Yunlong Zhu in 2008 [18], inspired by the ideas from the co evolution of symbiotic species in natural ecosystems and heterogeneous interaction between species. , PS20 is a multi-species optimizer which extends the dynamics of the canonical PSO algorithm by adding a significant ingredient that takes into account the symbiotic co evolution between species.

The algorithm initially create an ecosystem containing a species set  $X = \{S1, S2, \dots, Sn\}$ , and each species possesses a members set  $Sn = \{xi, x2, \dots, xm\}$  i.e., totally  $n * m$  ( $n$  species and  $m$  members within species) individuals co evolve in the ecosystem. The  $i$ th member of the  $k$ th species is characterized by the vector  $xik = \{xi1k, xi2k, \dots, xidk\}$  and the fitness being  $f(xki)$  and lower value of the fitness represents the higher ability of survival. Under this presumed external environmental stress, all individuals in this model co evolve to the states of lower and lower fitness by cooperating each other both within species and between species.

In each generation  $t$ , each individual  $xki$  will have social evolution as well as symbiotic evolution. Social evolution resembles the cooperation between individuals of the same species. Due to the socio biological background of the canonical PSO model,  $xki$  evolve according to the rules of the canonical PSO algorithm in this process thus accelerating towards the personal best position and the best position found by its neighbors where as symbiotic evolution addresses the cooperation between individuals of distinct species.  $Xki$  beneficially interacts with and rewards all its symbiotic partners (individuals of dissimilar species), i.e., each symbiotic partner donates its knowledge to aid other partners. Then  $xki$  accelerate towards its symbiotic partner of the best fitness. Finally if all individuals in the ecosystem cannot find a better position after a (here  $a$  is a constant) generations, it means that all species suffer a severe external environmental stress. Then randomly choose half species of the ecosystem to go extinct to release this stress for other species to survive. At the same time, randomly initiate equal number of species in the ecosystem for new experimentations and adaptations. In PS20 cooperation occurred in two levels, i.e., species level (interaction between species) and individual level (interaction within species).



### B. Invasive Weed Colony Optimization

Invasive Weed Optimization (IWO) is a numerical stochastic search algorithm proposed by Mehrabian and Lucas in 2006 [19], inspired by the ecological process of weed colonization and distribution. It is capable of solving general multi-dimensional, linear and nonlinear optimization problems with appreciable efficiency. Adapting with their environments, invasive weeds cover spaces of opportunity left behind by improper tillage; followed by enduring occupation of the field. Their behavior changes with time since as the colony become dense there is lesser opportunity of life for the ones with lesser fitness. The steps of the algorithm are described as below:

**Initialization:** includes a population of initial solutions being dispread over the D dimensional problem space with random positions.

**Fitness Evaluation:** Evaluate the individual fitness and rank the population according to their fitness.

**Reproduction:** Allowed to produce seeds depending on its own and the colony's lowest and highest fitness. This helps to concentrate on the highest fitness values in the search domain and hence increases convergence towards the group best value.

**Spatial Dispersal:** The generated seeds are being randomly dispersed over the D dimensional search space by normally distributed random numbers with mean

#### Table: Brief summary of the Bio Inspired optimization algorithms

equal to zero; but varying variance. The standard deviation (SD),  $\sigma$ , of the random function will be reduced from a previously defined initial value  $\sigma_{initial}$ , to a final value,  $\sigma_{final}$ , in every generation, which is given as follows:

$$\sigma_{iter} = \frac{(iter\ max - iter)^n}{(iter\ max)^n} (\sigma_{initial} - \sigma_{final}) + (\sigma_{final})$$

**Selection:** Select the Pmax or maximum number of plants from the best plants reproduced.

### C. Biogeography- Based Optimization

Biogeography-Based Optimization (BBO) is a global optimization algorithm developed by **Dan Simon** in 2008[20] and is inspired by mathematical models of biogeography by Robert MacArthur and Edward Wilson. Biogeography is the study of distribution of species in nature over time and space; that is the immigration and emigration of species between habitats. The application of this idea to allow information sharing between candidate solutions.

Each possible solution is an island and their features that characterize habitability are called suitability index variables (SIV). The fitness of each solution is called its habitat suitability index (HSI) and depends on many features of the habitat. High-HSI solutions tend to share their features with low-HSI solutions by emigrating solution features to other habitats. Low- HSI solutions accept a lot of new features from high-HSI solutions by immigration from other habitats. Immigration and emigration tend to improve the solutions and thus evolve a solution to the optimization problem. The value of HSI is considered as the objective function, and the

algorithm is intended to determine the solutions which maximize the HSI by immigrating and emigrating features of the habitats.

In BBO, there are two main operators: migration (which includes both emigration and immigration) and mutation. A habitat H is a vector of N (SIVs) integers initialized randomly. Before optimizing, each individual of population is evaluated and then follows migration and mutation step to reach global minima. In migration the information is shared between habitats that depend on emigration rates  $\mu$  and immigration rates  $\lambda$  of each solution. Each solution is modified depending on probability  $P_{mod}$  that is a user defined parameter. Each individual has its own  $\lambda$  and  $\mu$  and are functions of the number of species K in the habitat. Poor solutions accept more useful information from good solution, which improve the exploitation ability of algorithm. In BBO, the mutation is used to increase the diversity of the population to get the good solutions.

#### Features:

- In BBO the original population is not discarded after each generation. It is rather modified by migration.
- Another distinctive feature is that, for each generation, BBO uses the fitness of each solution to determine its immigration and emigration rate.

### V. COMPARATIVE ANALYSIS:

This section presents a comparative analysis of the algorithms seen so far in terms of the representation, operators, areas of application and control parameters

## A Survey of Bio inspired Optimization Algorithms

NAME OF ALGORITHM	REPRESENTATION	OPERATORS	AREAS OF APPLICATION	CONTROL PARAMETERS
GA	Binary, real no:s, permutation of elements, list of rules, program elements, data structure, tree, Matrix	Crossover, mutation, selection, Inversion , Gene Silencing	optimization problems in data mining and rule extraction, dynamic and multiple criteria web-site optimizations , decision thresholds for distributed detection in wireless sensor networks , Computer aided design path planning of mobile robots, fixed charge transportation problem, various scheduling problems ,assignment problems, flight control system design, pattern recognition , reactive power dispatch , sensor-based robot path planning, training of radial basis function, multi-objective vehicle routing problem, minimum energy broadcast problem in wireless ad hoc network, software engineering problems, pollutant emission reduction problem in the manufacturing industry, Power System Optimization problems, port folio Optimization ,optimal learning path in e learning, Web page classification system ,closest sting problem in bioinformatics ,structural optimization, defect identification system, molecular modeling, web service selection, cutting stock problem,drug design, personalized e-learning system, SAT Solvers	Population size, max.generation number, cross over probability, mutation probability, length of chromosome, chromosome encoding
GP	Tree structure (terminals & function set)	Crossover, Reproduction ,mutation, permutation,E diting, Encapsulation ,,Decimation	portfolio optimization, Design of image exploring agent, epileptic pattern recognition, automated synthesis of analogue electrical circuits symbolic regression, robotics, data mining(Automatic feature extraction, classification etc.),cancer diagnosis, power transformer fault classification automatic synthesis of analog electrical circuits	Population size, Maximum number of generations, Probability of crossover, Probability of mutation
ES	Real-valued vectors	Mutation, Selection, discrete Recombination	parameter estimation (Hatanaka et al., 1996), image processing (Gonzalez et al., 2001) ,computer vision system (Bergener et al., 2001),Task scheduling and car automation ,structural optimization, Evolution strategy for gas-turbine fault-diagnoses, A multi-parametric evolution strategies algorithm for vehicle routing problems, clustering	Population size, Maximum number of generations, Probability of crossover, Probability of mutation
DE	Real-valued vectors	Crossover, mutation ,selection	unsupervised image classification ,clustering , digital filter design , optimization of non-linear functions , global optimization of non-linear chemical engineering processes and multi-objective optimization .	S Poulation size ,Nd diamension of problem,F scale factor,Pr probability of crossover

PSO	D dimensional vector for position ,speed,best state	initializer, updater and evaluator.	multimodal biomedical image registration (Wachowiak et al., 2004) and the Iterated Prisoner's Dilemma (Franken and Engelbrechet, 2005), classification of instances in multiclass databases, feature selection, web service composition course composition, Power System Optimization problems (economic dispatching), Edge detection in noisy images, finding optimal machining parameter assembly line balancing problem in production and operations management, various scheduling problems ,vehicle routing problems, prediction of tool life in ANN, multi-objective, dynamic, constrained and combinatorial optimization problems ,QoS in adhoc multicast, Anomaly detection, color image segmentation , sequential ordering problem, constrained portfolio optimization problem, selective particle regeneration for data clustering , Extracting rules from fuzzy neural network, machinery fault detection, Unit commitment computation, Signature verification	number of particles, Dimension of particles, Range of particles, Vmax, Learning factors: c1c2, inertia weight , maximum number of iterations
ACO	Undirected graph	Pheromone Update and Measure, trail evaporation	TSP Problem, Quadratic Assignment problem (QAP) Job-Shop Scheduling problem. dynamic problem of data network routing, a shortest path problem where properties of the system such as node availability vary over time. continuous optimization and parallel processing implementations . vehicle routing problem ,graph colouring and set covering, agent-based dynamic scheduling, digital image processing, classification problem in data mining, protein folding problem	number of ants ,iterations , pheromone evaporation rate, amount of reinforcement
PFA	Linear=[x1,x2..x]	Dispersal, pollination	Continuous function optimization. tuning parameters in PID Controllers in higher order systems and RBF Neural Network Parameters Optimization	size of population, the boundary of parameter space, initial value of the maximum number of seeds
AIS	attribute string( a real-valued vector), integer string , binary string, symbolic string	immune operators( cloning, hyper mutation and selection based on elitism)	computer security ,anomaly detection, clustering /classification, numeric function optimization, ,learning ,IIR filter design, control, robotics, data mining[149],virus detection, pattern recognition , tuning of controllers, multi- modal optimization, job shop scheduling	Antibody population size ,, Number of antibodies to be selected for hyper-mutation ,number of antibodies to be replaced, multiplier factor $\beta$
ABC	D-dimensional vector (xi=1,2...D)	Reproduction, replacement of bee, selection	Scheduling problems, image segmentation, capacitated vehicle routing problem , (WSNs),assembly line balancing problem, Solving reliability redundancy allocation problem, training neural networks , XOR, Decoder–Encoder and 3-Bit Parity benchmark problems , pattern classification , reliability redundancy allocation problems, clustering, resource-constrained project scheduling problem, p-center problem	number of food sources which is equal to the number of employed or onlooker bees (SN), the value of limit, the maximum cycle number (MCN)

## A Survey of Bio inspired Optimization Algorithms

FSA	$X_i = (x_1, x_2, \dots, x_k, \dots, x_D)$ ,	Swarming ,following, searching	function optimization , Parameter estimation , combinatorial optimization], least squares support vector machine and geo technical engineering problems	Visual distance,max step length ,crowd factor
GSO	Unit vector	Scrounging, ranging, producing	Truss structure design, benchmark functions (He et al., 2006) and applied for optimal power flow problems (Fei et al.,2007), mechanical design optimization problems, multi objective optimization, Optimal placement of FACTS devices, machine condition monitoring, optimal location and capacity of distributed generations.	Population size, percentage of rangers, no: of rangers, Head angle, position, maximum pursuit angle, maximum turning angle, maximum pursuit distance
SFLA	$X_i=(x_{i1}, x_{i2}, \dots, x_{iS})$	Replacement, shuffling	Color Image Segmentation , Solving TSP , Automatic recognition of speech emotion water , Unit Commitment Problem , Grid Task scheduling, Optimal viewpoint selection for volume rendering , multi-user detection in DS-CDMA distribution , Fuzzy controller design , Optimal Reactive Power Flow , A Web Document Classification , Mobile robot path planning , classification rule mining , Combined Economic Emission Dispatch , Job-shop scheduling, ground water model calibration problems, Multicast Routing Optimization	number of frogs P, number of memeplexes, and number of evolutionary iterations for each memeplex before shuffling.
BFA	$\theta_i(j, k, l)$ represents $i$ -th bacterium at $j$ -th chemotactic, $k$ -th reproductive and $l$ -th elimination dispersal step.	Reproduction, chemotaxis, Dispersion , elimination	inverse airfoil design ,application for harmonic estimation problem in power systems, optimal power system stabilizers design , tuning the PID controller of an AVR, an optimal power flow solution, machine learning, an application of job shop scheduling benchmark problems; the parameters of membership functions and the weights of rules of a fuzzy rule set are estimated, transmission loss reduction ,implemented as the parameter estimation of nonlinear system model (NSM) for heavy oil thermal cracking, evaluation of independent components to work with mixed signals, solve constrained economic load dispatch problems ,application in the null steering of linear antenna arrays by controlling the element amplitudes, applications in multi objective optimization.	Dimension of the search space. ,number of bacteria , number of chemotactic steps , number of elimination and dispersal events , number of reproduction steps , probability of elimination and dispersal, location of each bacterium ,no: of iterations, step size $c(i)$
IWCO	Vector in D dimensional space	Reproduction, dispersal, selection	time-modulated linear antenna array synthesis, cooperative multiple task assignment of UAV, fractional order PID Controller, Training of Feed-Forward Neural Networks, Nash equilibrium search in electricity markets, blind multi-user detection for MC-CDMA interference suppression over multipath fading channel, recommender system	weed population size, modulation index ,standard deviations



PS2O	D dimensional vector for position , speed, best state	initializer, updater ,extinction, evaluator	Cooperative Cognitive Wireless Communication, constructing collaborative service systems (CSSs)	number of particles, Dimension of particles, Range of particles, Vmax, Learning factors: inertia weight , maximum number of iterations
BBO	H=h1,h2..hn as individuals of habitat.	migration (emigration and immigration) , mutation	general benchmark functions, constrained optimization ,the sensor selection problem for aircraft engine health estimation , power system optimization, groundwater detection and satellite image classification , web-based BBO graphical user interface , global numerical optimization, optimal meter placement for security constrained state estimation	number of habitats (population size), maximum migration rates, mutation rate,

VI. CONCLUSION

Bio inspired algorithms are going to be a new revolution in computer science. The scope of this area is really vast since as compared to nature, computer science problems are only a subset, opening a new era in next generation computing, modeling and algorithm engineering. This paper provides an overview of a range of BIAs drawn from an evolutionary metaphor or natural phenomena including the EAs (GA, GP, ES, DE and PFA), SI algorithms (PSO, ACO, ABC, BFA, FFA, AIS, FSA, IWD, SFLA, GSO) and ecology inspired algorithms (IWC, PS2O, BBO). Generally speaking, almost all of the EA and SI algorithms perform with heuristic population-based search procedures that incorporate random variation and selection. It has been witnessed that the applications and growth of natural computing in the last years is very drastic and has been applied to numerous optimization problems in computer networks, control systems, bioinformatics, data mining, game theory, music, biometrics, power systems, image processing, industry and engineering, parallel and distributed computing, robotics, economics and finance, forecasting problems, applications involving the security of information systems etc. Biologically inspired computing still has much room to grow since this research community is quite young. There still remain significantly challenging tasks for the research community to address for the realization of many existing and most of the emerging areas in technology. In particular, there are great opportunities in exploring a new approach/algorithm. For this it requires collaboration of researchers from different communities like computer science, artificial intelligence, biology, ecology, social science etc. in order to have a broader and deeper view and analysis of each micro level steps/interactions there by having much more significant and outstanding results. Nevertheless, nature-inspired algorithms are among the most powerful algorithms for optimization which is going to have a wide impact on future generation computing.

ACKNOWLEDGEMENT

This work is a part of the AICTE funded project titled ‘Bio inspired intrusion response system through feature relevance analysis on attack classification’, Under the Research Promotion Scheme (RPS), Ref.No: 8023/BOR/RID/RPS-59/2009-10.

REFERENCES

- [1] Back, T. 1996: Evolutionary algorithms in theory and practice. Oxford University Press.
- [2] J.H. Holland, Genetic algorithms and the optimal allocation of trials, SIAM J. Comput. 2 (2) (1973) 88–105
- [3] Koza, John R. 1992. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. Cambridge, MA: The MIT Press.
- [4] Beyer, H.G. and Schwefel, H.P. 2002: Evolution strategies. *Natural Computing* 1,3–52.
- [5] R. Storn, K. Price, Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces, *Journal of Global Optimization* 11 (1997) 341–359.
- [6] Upeka Premaratne , Jagath Samarabandu, and Tarlochan Sidhu, “A New Biologically Inspired Optimization Algorithm”, Fourth International Conference on Industrial and Information Systems, ICIIS 2009, 28-31 December 2009, Sri Lanka.
- [7] Bonabeau, E., Dorigo, M. and Theraulaz, G. 1999: *Swarm intelligence*. Oxford University Press
- [8] Kennedy, J.; Eberhart, R. (1995). "Particle Swarm Optimization". *Proceedings of IEEE International Conference on Neural Networks*. **IV**. pp. 1942–1948.
- [9] Dorigo, M., Maniezzo, V., & Coloni, A. (1996). *Ant System: Optimization by a colony of cooperating agents*. *IEEE Transactions on Systems, Man, and Cybernetics – Part B*, 26, 29–41.
- [10] D. Karaboga, B. Basturk, A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm, *Journal of Global Optimization* 39 (2007) 459–471
- [11] X. Li, Z. Shao, J. Qian, An optimizing method base on autonomous animates: fish- swarm algorithm, *Systems Engineering Theory and Practice* 22 (2002) 32–38.
- [12] Shah\_Hosseini, H. Shahid Beheshti Univ., Tehran Problem solving by intelligent water drops IEEE Congress on Evolutionary Computation, 2007. CEC 2007.
- [13] K. M. Passino, “Biomimicry of bacterial foraging for distributed optimization and control,” *IEEE Control Syst. Mag.*, vol. 22, no. 3, pp.52–67, Jun. 2002
- [14] D. Dasgupta, *Artificial Immune Systems and Their Applications*, Springer, Berlin, 1999
- [15] X.S Yang, “Fire fly algorithm for multimodal optimization”, in proceedings of the stochastic Algorithms. Foundations and Applications (SAGA 109) vol.5792 of Lecture notes in Computer Sciences Springer, Oct. 2009
- [16] S.He, Q.H.Wu, Senior Member IEEE and J.R Saunders. A novel group search optimizer inspired by Animal Behavioral Ecology; 2006, IEEE Congress on Evolutionary Computation, 1272-1278
- [17] Eusuff MM and K.E Lansey; Optimization of water distribution network design using SFLA (2003)
- [18] Hanning Chen , Yunlong Zhu, Optimization based on symbiotic multi-species coevolution; journal on Applied Mathematics and Computation 205 (2008)
- [19] A.R. Mehrabian, C. Lucas, A novel numerical optimization algorithm inspired from weed colonization, *Ecological Informatics* 1 (2006) 355–366.

[20] Simon, D., 2008. Biogeography-based optimization. IEEE Transactions on Evolutionary Computation. 12 (6), 702–713.

**Binitha S**, Received the BTech degree in Computer Science and Engineering from College of Engineering Cherthala under CUSAT, Kerala. Currently pursuing M-Tech. in Computer Science from Pondicherry University, Pondicherry, India.

**Dr. S.Siva Sathya** is an Associate professor in the Dept. of Computer Science, Pondicherry University. Her areas of interests include Evolutionary Algorithms, Bioinformatics, Intrusion detection, etc. She has to her credit a number of research papers in International journals and conferences.