A Survey: Evolutionary and Swarm Based Bio-Inspired Optimization Algorithms

Rashmi A. Mahale[*], Prof.S.D.Chavan[**]

[*] Dept. Electronics and Telecommunication, Dr. D Y Patil Institute of Engineering and Technology, Pune, India
[**] Dept. Electronics and Telecommunication, Dr. D Y Patil Institute of Engineering and Technology Pune, India

Abstract- Nature is the best tutor and its designs and strengths are extremely massive and strange that it gives inspiration to researches to imitate nature to solve hard and complex problems in computer sciences. Bio Inspired computing has come up as a new era in computation covering wide range of applications. This paper gives overview of most predominant and successful classes of bio inspired optimization methods involving evolutionary and swarm based algorithms inspired by natural evolution and collective behavior in animals respectively.

Index Terms- Bio Inspired Algorithms; Evolutionary Algorithms; Swarm based algorithms

I. Introduction

Bio-inspired algorithms are based on the structure and functioning of complex natural systems and tend to solve problems in an adaptable and distributed fashion. They are a problem solving methodology derived from the structure, behavior and operation of natural system and remarkably flexible and adaptable nature. Exploring the Bioinspired algorithms is the enormous computational efforts to solve optimization problems by the conventional algorithms which tend to increase the problem size. They have the ability to describe and resolve complex relationships from intrinsically very simple initial conditions and rule.

Bioinspired computing can solve the problems of almost all areas including wireless sensor networks, computer networks, security, robotics, biomedical engineering, control systems, parallel processing, data mining, power systems, production engineering, image processing and many more. Designing 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 new set of solutions.

The organization of paper as: Section II provides an overview of EAs. Algorithms of SI family are discussed in section III. Section IV explores merits and demerits of bio inspired algorithms. Section V compares BIAs with conventional algorithms. Conclusion is drawn in section VI.

II. Evolutionary Algorithms

The term Evolutionary algorithm [1] is used to designate a collection of optimization techniques whose functioning is loosely based on metaphors of biological processes. Evolutionary computation is a paradigm in the artificial intelligence that involves 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, based on the biological evolution in nature which is responsible for the purpose of all living beings on earth. They have different functional components as the fitness function, initialization, selection, recombination, mutation and replacement.

The algorithm maintains a collection of potential solutions of the problem, which are used to create new potential solutions through the use of operators. These operators act on and produce new collection of solutions, which are selected on the basis of their quality. The algorithm uses this process repeatedly to generate new collection of potential solutions until some stopping criterion is met. Each algorithm starts by generating an initial population of feasible solutions, and advances iteratively from generation to generation towards a best solution.

The EA family members are Genetic algorithm (GA), Genetic programming (GP), Differential Evolution (DE), Evolutionary strategy (ES), and Paddy Field Algorithms.

A. Geneic Algorithms

Genetic Algorithms proposed by Holland in 1975 [2][3], are numerical optimization algorithms inspired by Natural selection and Natural Genetics. They follow the principles of Charles Darwin theory of survival of the fittest. However its great performance in optimization, GA has been regarded as a function optimizer. GA techniques differ from more traditional search algorithms in that they work with a number of candidate solutions rather than one candidate solution.

Each candidate solution of a problem is represented by a data structure known as an individual. A group of individuals collectively comprise what is known as a population. GAs is initialized with a population of random guesses. GA includes operators such as Reproduction, Crossover, Mutation and Inversion. Reproduction is a process in which a new generation of population is formed by selecting the fittest individuals in the current population.

Crossover is responsible for producing new offspring by selecting two strings and exchanging portion of their structures. The new offspring may replace the weaker individuals in the population. Mutation is a local operator which is applied with a very low probability of occurrence. Its Function is to alter the value of a random position in a string. Finally, Inversion is a process which inverts the order of elements between two randomly chosen points on the string.

B. *Genetic Programming*

GP proposed by John Koza in 1992[4], being an extension to Genetic algorithms it vary from the GA in terms of representation of the solution.GP represent an indirect encoding of a potential solution, in which search is applied to the solution directly, and solution can be a computer program. The basic difference in GP and GA is that GA involves fixed length encoding in contrast to GP which employ variable length encoding. In genetic programming, the individuals in the population are compositions of functions and terminals appropriate to the particular problem domain. The set of functions used typically includes arithmetic operations, mathematical functions, conditional logical operations, and domain-specific functions.

GP involves the following three steps:

Generate an initial population of random compositions of the functions and terminals of the computer programs. Execute each program in the population and assign it a fitness value according to how well it solves the problem. Create a new population of computer programs by applying the following two primary operations.

(I) Reproduction: Copy existing computer programs to the new population.

(II) Crossover: Create two new computer programs by genetically recombining parts of two existing programs.

C. *Evolution Strategies*

Evolution Strategies was developed by three students (Bienert, Rechenberg, Schwefel) at the Technical University in Berlin in 1964[5].ES is a global optimization algorithm inspired by the theory of adaptation and evolution by means of natural selection. ES involves macro-level or the species-level process of evolution (phenotype, hereditary, variation), which meant for optimizing the progress of the search, by evolving solutions for the problems being considered as well as some parameters for mutating these solutions.

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

(1+1)-ES: In this mechanism by mutation, resulting individual is evaluated and compared to its parent, and the better survives to become a parent of a next generation, while other is discarded.

(μ+λ)-ES: Here μ parents are selected from the current generation to generate λ offsprings, through some recombination and/ or mutation operators. Out of the union of parents and offsprings (μ+λ), the best μ kept for next generation.

(μ, λ)-ES: Here μ parents selected from the current generation to generate λ offsprings (With $\lambda \geq \mu$). Only the best μ offspring individuals form the next generation discarding the parents completely.

D. *Differential Evolution*

It was proposed by Storn and Price in 1995[6].The main difference between GA 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 starting of an evolutionary process, the mutation operator of DE prefers the 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.

Some advantages of DE are:

- 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.
- The user has to find the best values for the problem-dependent control parameters used in DE and this is a time consuming task.

E. *Paddy Field Algorithm*

It was proposed by Premaratne et al in 2009[7], which operate on a reproductive principle dependant on proximity to the global solution and population density similar to plant populations. In contrast to evolutionary algorithms, it does not involve combined behavior or crossover between individuals (optimum solution can migrate) instead it uses pollination and dispersal.

PFA consists of five basic steps:

1**. Sowing**: It begins with scattering seeds (initial population p0) at random in an irregular field.

2. Selection: Here the best plants are selected so as to selectively discard adverse 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 tend to grow to be the best plants 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.

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.

III. Swarm Intelligence

A swarm (irregular movements of the particles) has been defined as a set of (mobile) agents which are liable to communicate directly or indirectly with each other, and which collectively carry out a distributed problem solving. **Swarm intelligence [**Kennedy and Eberhart, 2001[8]**]** is the discipline that deals with natural and artificial systems composed of many individuals that coordinate using decentralization and self-organization.

The agents follow very simple rules, and although there is no centralized control structure dictating how individual agents should behave, local, and to a certain degree random, interactions between such agents lead to the emergence of "intelligent" global behavior, unknown to the individual agents. Natural examples of SI include ant colonies, bird flocking, bee colonies, bacterial growth, and fish schooling etc.

The typical swarm intelligence system has the following properties:

- It is composed of many individuals.
- The individuals are relatively homogeneous (i.e., they are either all identical or they belong to a few typologies).
- The interactions among the individuals are based on simple behavioral rules.
- The overall behavior of the system results from the interactions of individuals with each other and with their environment.

The simplest mathematical models of animal swarms generally represent individual animals as following three rules.

- Move in the same direction as your neighbors.
- Remain close to your neighbors
- Avoid collisions with your neighbors.

Swarm intelligence can be described by considering following 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.

Swarm intelligence algorithms can be divided on the basis of social behavior of animals/microbes, natural river systems (water drop algorithm) and human immune system (artificial immune system), out of which algorithms based on collective behavior of animals can be explored as:

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[9]. It is inspired by the social behavior of bird flocking and fish schooling.

PSO has been applied to many engineering problems due to its unique searching Mechanism, simple concept, computational efficiency and easy implementation. It utilizes a “population” of particles that fly through the problem hyperspace with given velocities.

At each iteration, the velocities of the individual particles are stochastically adjusted according to the historical best position for the particle itself and the neighborhood best position. Both the particle best and the neighborhood best are derived according to a user defined fitness function.

The movement of each particle naturally evolves to an optimal or near-optimal solution. PSO is not largely affected by the size and nonlinearity of the problem, and can converge to the optimal solution in many problems where most analytical methods fail to converge.

Each particle (population member) in the swarm correspond to a solution in a high-dimensional space with four vectors, its current position, best position found so far, the best position found so far by its neighborhood and its velocity and adjusts its position in the search space based on the best position reached by itself (pbest) and its neighbor (gbest) during the search process.

Steps in PSO algorithm can be briefed as below:

1. Initialize the swarm by assigning a random position.
2. Estimate 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 as pbest and the current particle‘s position, xi, as pi.
4. Identify the particle that has the best fitness value. This fitness function identified as gbest, pg
5. Revise the velocities and positions of all the particles using (1) and (2).
6. Repeat steps 2–5 until a sufficiently good fitness value is achieved.

Advantages of PSO over GA

- PSO is easier to implement and there are fewer parameters to adjust.
- PSO has a more effective memory capability than the GA.
- PSO maintains diversity as all the particles use the information related to the most successful particle in order to improve themselves, whereas in GA the worse solutions are removed and only the good ones are saved

B. Ant Colony Optimization

ACO is among the most successful swarm based algorithms proposed by Dorigo & Di Caro in 1992[10],

It is a probabilistic method for solving computational problems, which can be reduced to find good paths through graphs It is inspired by the behavior of ants in finding paths from the colony to the food. In the real World, ants initially wander randomly, after finding food, they return to their colony while laying down pheromone trails.

If other ants find such a path, they do not keep traveling at random, but rather follow the trail, returning and reinforcing it if they eventually find food. However, the pheromone trail starts to evaporate over time, therefore reducing its attractive strength.

More the time to travel down and back again for ant, more will be evaporation of pheromones. A short path, by comparison, gets marched over faster, and thus the pheromone density remains high as it is laid on the path as fast as it can evaporate. Pheromone evaporation also has the advantage of avoiding the convergence to a locally optimal solution. If there were no evaporation at all, the paths chosen by the first ants would tend to be excessively attractive to the following ones. In that case, the exploration of the solution space would be constrained. Thus, when one ant finds a short path from the colony to a food source (i.e., a good solution), other ants are more likely to follow that path, and positive feedback eventually leaves all the ants following a single path.

C. Artificial Bee Colony Algorithm

These algorithms are inspired by the behavior of bees in nature which are classified into two; foraging behavior & mating behavior. Algorithm simulating foraging behavior of the bees, proposed by Karaboga and Basturk[11] includes artificial bee colony (ABC), the virtual Bee algorithm, the bee colony optimization algorithm, Bee hive algorithm. In the Bee swarm optimization algorithm an individual entity exhibit a simple set of behavior policies, but a group entity shows complex evolving behavior with useful properties such as scalability and adaptability.

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 on looker and one going to the food source visited by it before is named employed bee. The scout bee is the kind of bee that carries out random search for new sources
The location of food source corresponds to a possible solution to the optimization problem and the nectar amount of a food source corresponds to the quality of the solution.

An employed bee produces a modification on the position in her memory depending on the local 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.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

Fish Swarm Algorithm (FSA) is a new intelligent swarm modeling approach that consists primarily of searching, swarming, and following behaviors of fish. FSA technique proposed by Li et al in 2002[12], 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 Xi and food satisfaction for the fish is represented as FSi. The relationship between two fish is denoted by their Euclidean distance

FSA responds to three typical behaviors of a fish, which are:

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 Xi with its neighbor at center position Xc, if center swarm has greater food concentration than fish's Xi position, it will move to Xi+1 towards Xc.

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 (Xmin) in order to increase satisfaction(i.e. gain relatively more food and less crowding. 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 visual and step parameters.

E. Intelligent Water Drop Algorithm

IWD is an innovative population based method proposed by Hamed Shah-hosseini in 2007[13]. The intelligent water drops (IWD) algorithm is a new swarm-based optimization algorithm inspired from observing natural water drops that flow in rivers. 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
2. The velocity that it is moving now, Velocity

During the journey of IWD from source to destination, it travels through the environment from which it removes some soil, with gain in speed. During journey from its current location to its next location, the IWD velocity is increased by the amount non-linearly proportional to the inverse of the soil between the two locations. Therefore, a path with less soil lets the IWD become faster than a path with more soil. An IWD collects soil during its trip in the environment which is collected from the path going towards the destination. The amount of soil added to the IWD is non-linearly proportional to the inverse of the time needed for the IWD to pass from its current location to the next location. This time interval 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.IWD prefers the paths with low soils than paths with higher soils. 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. Artificial Immune System

Artificial Immune algorithm[14] is based on clonal selection principle and is a population based algorithm. The AIS is inspired by the human immune system which is a highly evolved, parallel and distributed adaptive system that exhibits the strengths like: immune recognition, reinforcement learning, feature extraction, immune memory, diversity and robustness. The mutation operator is the efficiency deciding factor of this technique. The steps in AIS are as follows;

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

Cloning, determines the resemblance or fitness of each antibody. Based on this fitness the antibodies are cloned which means the best antibodies are cloned mostly.

Hyper mutation: The clones are then undergoes to a hyper mutation process in which the clones are mutated in inverse proportion to their affinity. The clones are then evaluated along with their original antibodies out of which the best N antibodies are selected for the next iteration.

IV. Merits and Demerits

Bio-inspired algorithm are designed to be flexible, completely distributed and efficient. Bioinspired systems can grow, organize, and improve themselves with little direction from humans.

A. *Merits*

Merits of Bioinspired Algorithm can be discussed on the basis following criteria:

a) Flexibility – Strength though flexibility or strength in numbers, starts with flexible size

b) Performance - Work well even when the task is poorly defined, No saturation limit in performance

c) Scalability - Scalability is not really a challenge

d) Flexibility in decision making -Tend to find the alternate best available solution, not depends on programmer's understanding of the program

e) Improvement Scope and innovation - Largely unexplored field, no limit for development

B. *Demerits*

Bioinspired Algorithms has a few demerits.

1. Component Design: A major drawback in case of Bio-inspired algorithms is whether to compromise on competitive interactions or cooperative interactions.
2. Lack of data: Biological systems are extremely hard to study, and the lack of data on a system may affect the design of the algorithm derived from the analogous biological system..
3. Lack of complete adaptability: Bio-inspired algorithms cannot be completely adapted to real world systems because of conflicts in scalability or performance issues. For example, in the Bird-flocking algorithm, achieving individual safety from being singled out will require that we work out the path of each individual in an explicit manner.
4. Low performance: Bio-inspired algorithms typically have low performance. This is because biological methods aim to behave well in a wide variety of situations as against aiming to reach the goal quickly. However, we improve performance by compromising on the adaptability or flexibility of the algorithm if we know parameters about the environment that the algorithm will be working in.

V. Comparison with conventional

Comparison of bioinspired algorithms with conventional algorithms can be discussed on the basis of following criteria:

Intelligence: Bioinspired Algorithms are based on simple rules which take bottom-up approach. While conventional algorithms takes top-down approach.

Testing: In Bioinspired methods, improvements have to be verified on successive generations taking more time while in conventional, testing results can be obtained immediately.

Improvement: Improving of the Bio-inspired algorithms is not easy because verifiability compared to conventional algorithms.

Flexibility to practical situation: Bioinspired algorithms have to be modified when applied to practical problems, while conventional algorithms are built keeping the practical situations and the end result in mind.

VI. Conclusion

Bioinspired Algorithms have the ability to describe and resolve complex relationships from essentially very simple conditions and rules. This paper provides overview of a range of Bioinspired Algorithms drawn from evolutionary phenomenon including EAs and some SI algorithms. These algorithms perform with metaheuristic population based search procedures. The merits and demerits of bioinspired algorithms in practical problems are also shown with comparison to conventional algorithm.

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. Compute. 2 (2) (1973) 88–105

[3] R.Shivakumar and Dr R.Lakshmipathi, "Implementation Of an innovative Bio Inspired GA and PS Algorithm for Controller design considering steam GT Dynamics" , IJCSI International Journal of Computer Science Issues, , Vol. 7, Issue 1, No. 3, January 2010

[4] John R. Koza, "Evolution of Subsumption Using Genetic Programming" Computer Science Department, Stanford University, Stanford, CA 94305 USA

[5] Beyer, H.G. and Schwefel, H.P. 2002: Evolution Strategies. Natural Computing 1,3–52.

[6] R. Storn, K. Price, Differential evolution – a simple and Efficient heuristic for global optimization over spaces, Continuous Journal of Global Optimization 11 (1997) 341–359.

[7] Upeka Premaratne, Jagath Samarabandu, and Tarlochan Sidhu, ―A New Biologically Inspired Optimization Algorithm, Fourth International Conference on Industrial and Information Systems, ICIIS 2009, 28-31December 2009, Sri Lanka.

[8] Mark Fleischer," Foundations of Swarm Intelligence: From Principles to Practice", SWARMING: NETWORK ENABLED C4ISR c at 2003

[9] Kennedy, J.; Eberhart, R. (1995). "Particle Swarm Optimization". Proceedings of IEEE International Conference on Neural Networks. **IV**. pp. 1942–1948.

[10] Dorigo, M., Maniezzo, V., & Colorni, A. (1996). Ant System: Optimization by a colony of cooperating Agents.IEEE Transactions on Systems, Man, and Cybernetics – Part B, 26, 29–41.

[11] Dervis Karaboga, Bahriye Akay. "A comparative study Of Artificial Bee Colony algorithm", Applied Mathematics and Computation 214 (2009) 108–132

[12] X. Li, Z. Shao, J. Qian,Anoptimizing method base on Autonomous animates: fish- swarm algorithm, Systems Engineering Theory and Practice 22 (2002) 32–38.

[13] Shah-Hosseini, H. (2009) 'The intelligent water drops Algorithm: a nature-inspired swarm-based optimization Algorithm', *Int. J. Bio-Inspired Computation*, Vol. 1, Nos. 1/2, pp.71–79.

[14] D. Dasgupta, Artificial Immune Systems and Their Applications, Springer, Berlin, 1999

[15] Abbott, R. (2005), "Challenges for Bio-inspired, Computing", The Proceedings of The BioGEC *Workshop, GECCO*, New York: ACM, pp. 12-22

AUTHORS

First Author – Rashmi A. Mahale, Dept. Electronics and Telecommunication, Dr. D Y Patil Institute of Engineering and Technology, Pune, India, e-mail: rashmi.180405@gmail.com
Second Author – Prof.S.D.Chavan, Dept. Electronics and Telecommunication, Dr. D Y Patil Institute of Engineering and Technology Pune, India, e-mail: sdchavan27@rediffmail.com