

## A STUDY OF NATURE INSPIRED OPTIMIZATION ALGORITHMS

Amuthadevi<sup>1\*</sup>, Gayathri Monicka<sup>2</sup>, Madhusudhanan<sup>3</sup><sup>1</sup>Department of Computer Science & Engineering, Adhi College of Engineering & Technology, Chennai, INDIA<sup>2</sup>Department of EEE, Adhi College of Engineering & Technology, Chennai, INDIA<sup>3</sup>Department of Computer Science & Engineering, Perumal Manimekalai College of Engineering, Hosur, INDIA

## ABSTRACT

Nature has the abilities of balancing the 'eco-system', diversity maintenance and adaptation to changing environment which educated many strategies to the human beings and can be adaptable in the technologies. The generation of human beings and the behavior of many social agents and animals gave inspiration to design a set of meta-heuristic algorithms which are used to find optimal or best solutions for large number of complex problems. Most of these algorithms are independent of the nature of the problems to be solved. As many algorithms are being implemented for various applications, no one is proved as best among all the optimization problems. This paper surveys some of the nature inspired algorithms, their adaptability to real world problems and concludes with limitations and improvisation required in these algorithms.

Published on: 14<sup>th</sup>– August-2016

## KEY WORDS

Optimization algorithms, Bio-inspired algorithms, Natural Evolution.

\*Corresponding author: Email: [a.bcamuthadevi@gmail.com](mailto:a.bcamuthadevi@gmail.com), [drkssece@gmail.com](mailto:drkssece@gmail.com)

## INTRODUCTION

Charles Darwin analyzed the evolution of natural components and defined the "Theory of Natural Evolution". This theory describes about the "Survival of the Fittest" of the natural elements by taking on of the changing/dynamic environments. The natural elements have the ability of self-processing and self-learning. Best example for this is the 'generation of the human beings'. All the search/optimization problems go along with the "Survival of the Fittest". The optimization plays a major role in most of the engineering applications.

The problem solving methods are categorized into two types. They are Classical methods/traditional methods and Heuristic methods. Classical methods use either simple logical or mathematical steps and have clearly defined ways to get a solution. But heuristic methods are useful to solve NP-hard problems and need some optimization algorithms. These optimization algorithms mimic the behavior that inspired from the natural components [1].

The major reasons which makes trouble to solve a problem are [2,3]:

- Solution space has large number of possible solutions and this creates the need of exhaustive search to find the best answer.
- As the problem is more complicated, simple search is not useful.
- The evaluation strategy may vary with time or it may give a noisy solution.
- Constraint on the solution is so weighty and sometimes finding a single solution is so difficult.
- Wrong assumption about the problem/constraints may create barrier that prevents to find out a solution.

Some search algorithms like Gradient search are mostly problem dependent and convergence also depends on the selection of starting solutions. The algorithm cannot be parallelized. But nature inspired optimization algorithms can be used in wide class of applications and can be parallelized [2]. When adapting the nature inspired algorithms the following should be considered:

- The problem should be represented properly.

- The solution must be evaluated by a strategy. This will be useful to qualify the solution by using a fitness function.
- The operators should be designed to give the next set of solutions.
- This following contents of this paper is having 3 sections. First one explains about evolutionary algorithms and second one is swarm intelligence based optimization algorithms. Then the conclusion gives the merits and limitations of these algorithms.

## EVOLUTIONARY ALGORITHMS

### GENETIC ALGORITHM (GA)

GAs are powerful as they apply natural selection/natural evaluation concepts based stochastic search and optimization methods [4]. GAs work on individual populations, representing candidate solutions for optimization problems. Individuals comprise gene strings (chromosomes). GAs apply the survival of the fittest, selection, reproduction, crossover (recombining), and mutation principles on individuals to ensure better individuals (new solutions). GA's disadvantage is its inability to locate an exact global optimum, as there is no best solution guarantee.

A control parameters (optimal or near-optimal) set for a GA or GA application does not generalize all cases. The GA is defined by control parameter set  $II = \{P, C, U, M\}$ , where:

- P is population size.
- C is crossover rate. It decides convergence to pull a population to local maximum or minimum. Values range from 0 to 1. Taking higher value ensures faster convergence.
- U is the probability of an allele involved in a crossover. The probability specifies how often a crossover is allowed. 100% probability makes all offsprings and 0% makes new generation an exact copy of the earlier generation.
- M is mutation rate. It is a divergence operator to break one/more population members out of local maximum/minimum to get better maximum/minimum. Mutation rate ranges from 0 to 1. It is less frequent than cross over, so small values are taken for it.

GA operations are shown in the [Figure -1].

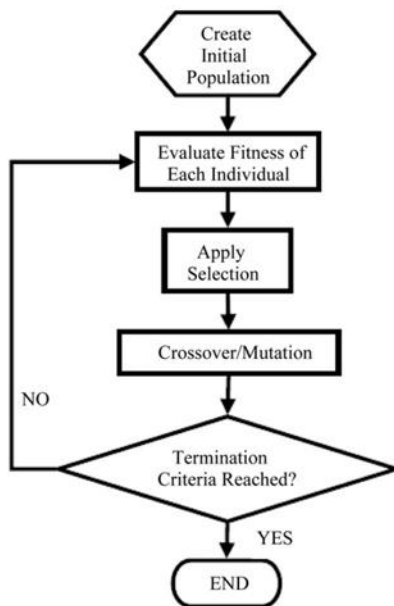


Fig: 1: Flow chart of GA

GA operations are:

- Initial population generation: This is generated randomly in a range of each parameter.
- Evaluation of fitness: Once initial population generation is over, each individual's fitness is determined. Fitness is a numeric index, measuring each individual's effectiveness as a solution. This is utilized from the population to choose members for reproduction.
- Selection Operation: An individual pair selected from current population using selection method.
- Crossover Operation: One/multipoint crossover is applied to newly selected (parents) individuals to generate two offspring. In detail, numbers of crossover points/parameters to be optimized are equal.
- Mutation Operation: Mutation operator applied randomly to newly generated offspring to prevent premature convergence to local minima [5].

- The population count is usually set as 50 for small dataset and 100 to 200 is the standard value for large data set. For mutation rate  $1/L$  is set, where  $L$  is the length of the encoding. Number of iterations is related to fitness function. Fitness function shows major improvements in earlier generations and then asymptotically reaches optimum.

## EVALUATION STRATEGIES

These use a simple variation in the process of GA [6,7]. Some of the ES schemes are:

- **(1+1)ES**: Single solution is selected and mutation is performed. The new one is compared with the solution that was before mutation. The best will be used as parent in further iterations.
- **( $\mu+\lambda$ )-ES**:  $\mu$  number of parent are selected from current iteration and  $\lambda$  off-strings are generated. From  $\mu$  and  $\lambda$  offstrings, best  $\mu$  number of offstrings will survive for next iteration.
- **( $\mu,\lambda$ )-ES**:  $\mu$  number of parent are selected from current iteration and  $\lambda$  off-strings are generated (with the condition that  $\lambda > \mu$ ). From the offstrings, best  $\mu$  number of offstrings will survive for next iteration and previously participated parents are discarded completely.

## SWARM INTELLIGENCE BASED ALGORITHMS

These algorithms are inspired based on the food searching behavior of social agents like insects and fishes. Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO) and Artificial Bee Colony Optimization (ABC), Cuckoo Search, Bat algorithm, Firefly algorithm come under this category.

### PARTICLE SWARM OPTIMIZATION (PSO)

PSO is based on rapid change in the movements and communications among the birds/fishes. Each particle finds its best solution from current position called as 'pbest' and all the particles have best among its group in nearby distances called as 'gbest' [9,12]. The purpose of PSO is to speed up the particles towards the gbest locations with a biased speeding factor in all the iterations. The algorithm of PSO is,

- Initialize a set of particles.
- Fitness value is calculated for each particle called as pbest.
- Compare pbest with previous iterations pbest and assign the best value for particle best.
- Compare all the nearby particles pbest and assign the best one as gbest for the group.
- For each particle, calculate particle velocity and the position according to equation based on current position, velocity and direction of particle that has gbest.
- Repeat the steps 2 to 5 until stopping criteria is reached.

The particle velocity is updated by the following equations:

- $v = v + c1 * \text{rand}() * (\text{pbest} - \text{present}) + c2 * \text{rand}() * (\text{gbest} - \text{present})$  (1)
- $\text{present} = \text{percent} + v$  (2)

where  $v$  is the particle velocity,  $\text{percent}$  is the current particle (position or solution).  $\text{pbest}$  and  $\text{gbest}$  are defined as stated before.  $\text{rand}()$  is a random number between (0,1).

PSO shares some of the properties like initial population generation and fitness function with GA. As PSO has no evolution operation, particles update the solution by their velocities towards the solution and have some memory utilization. PSO has only few parameters to set [12].

Even 10 particles can give good solutions. Standard number of particles ranges from 10 to 20. For some specific problems and wider solution space, particle range can be set as 100 to 200. Dimension of the particles depends on the problem to be solved.  $V_{\max}$  is the parameter used to define the maximum change of the particle in each iteration. In a particle  $x_1$ , if  $V_{\max}=20$ , then the changes in the solution can be  $[-10, 10]$ . There are 2 learning factors  $c_1$  and  $c_2$ , and it depends on the problem. Stop condition may be based on either the fitness value or the maximum number of iterations (Standard number of iterations is set as 1000 to 2000 for extremely complex problems)

### ANT COLONY OPTIMIZATION (ACO)

Real ants deposit a pheromone trail in the path of forward and return journey while food searching and nest building etc., This idea was used in ACO and originally implemented for Travelling salesperson problem (TSP) to find an optimal path in the weighted graphs. In ACO, artificial ants are used to find an optimal solution and then only best solutions are updated by increasing pheromone trail values and bad solutions are discarded by decreasing pheromone values [9]. ACO algorithm has the following steps.

- Parameter Setting: Number of artificial Ants and Pheromone trail, Pheromone evaporation rate and amount of reinforcement
- For each ant construct set of possible solutions.
- Daemon action is optional and depends on the problem before updating pheromone trails.
- For good solutions increase the pheromone value and decrease for others.

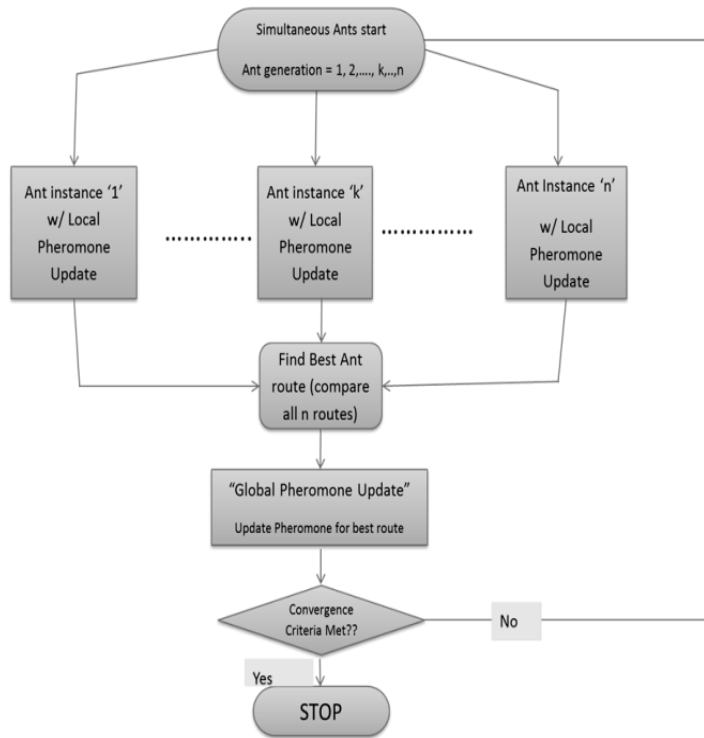


Fig. 2. Flow chart of ACO algorithm

When parameter setting, the number of ants depends on the optimization algorithm. M number of ants move from one solution x to y (in graphs one node x to another node y). In that tour, Pheromone relative importance  $\alpha$  and heuristic importance  $\beta$  (some prior information about movement from x to y). This denotes the strength of movement of solution x to y (in graphs selection of path from node x to node y).

An ant k moves from x to y with the probability,

$$p_{xy}^k = \frac{(\tau_{xy}^\alpha)(\eta_{xy}^\beta)}{\sum_{z \in \text{allowed}_x} (\tau_{xz}^\alpha)(\eta_{xz}^\beta)} \quad (3)$$

Where  $\tau_{xy}$  is the amount of pheromone deposited in the movement from x to y, and  $\eta_{xy}$  is the prior information. (In graphs this is based on the distance from node x to node y).

All the m ants complete their search then the trails are updated by,

$$\tau_{xy} \leftarrow (1 - \rho)\tau_{xy} + \sum_k \Delta\tau_{xy}^k \quad (4)$$

$$\Delta\tau_{xy}^k = \begin{cases} Q/L_k & \text{if ant } k \text{ uses curve } xy \text{ in its tour} \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

Where  $\rho$  is the pheromone evaporation coefficient ((1- $\rho$ ) indicates the pheromone persistence factor) and Q is the amount of pheromone ants release [14].

### ARTIFICIAL BEE COLONY OPTIMIZATION(ABC)

ABC algorithm is based on the food searching behavior of honey bees. Bees colony has three types of bees called as scout bees, onlooker bee and employee bees.

The initial step in this algorithm is to send the bees in different directions to search for 'best quality food'. When the bees found the location and the quality of the desired food, they come back to the colony and inform to the remaining bees using a communication mechanism called as 'waggle dance'.

This informs about the distance of food resource from colony, the way to reach the food and the quality of source of food. Onlooker bee is responsible for selection of best source. Then all the bees are attracted with the bee that brought information about the best quality food source [10, 11].

Actual Bee colonies behavior is given in the following pseudo code.

- Send the scouts onto the initial food sources  
REPEAT
- Send employee bees to find food sources & nectar Amounts
- Calculate probability of food sources to prefer by onlooker bees
- Send the onlooker bees onto food sources and calculate the nectar amounts.
- Stop the Exploitation process of the sources exhausted by the bees
- Send the scout bees in the search area for discovering new food sources
- Memorize the food source found so far  
UNTIL (termination conditions met)

In ABC optimization algorithm, the original concept is simply modified. Each food source is considered as a solution and the nectar amount qualifies the solution.

- Initialization: Assign an initial set of solutions in which 'employee bees' can search to find a set of possible solutions.
- Fitness Evaluation: Bees which searched the solutions are evaluated based on the fitness function using identified solutions and their visited locations.
- Evaluating Best Value: The bees having higher fitness value will be selected and visited locations are used for 'neighborhood search'.
- Iteration: If the solution is not optimal solution, then other employee bees are sent for new search. If a solution representing a food source is not improved by a predetermined number of trials, then that solution (food source) is abandoned.

In every search, quality of solution is better than previous, artificial bees forgot the previous solution and location and saves the better as new one. For better parameter setting, the number of employee and onlooker bees must be equal to the number of solutions to be searched. Maximum iteration count depends on the problem.

## CONCLUSION

This paper gives a review of few popular optimization algorithms. In real applications, many modified versions of these are used based on the nature of the problem and the size of the solution space.

GA uses many parameters to control the evolutionary search for a problem's solution. These include rates of crossover and mutation, maximum generations and number of individuals in a population. There are no hard and fast rules to choose appropriate values for parameters. PSO, ACO and ABC have few parameters to adjust when comparing to GA. As swarm intelligence algorithms does not have evolutionary process the stability and convergence is high. These also use some memory to remember good solutions.

GA can be used in wide range of applications including classification, data mining, bio-informatics and defect identification systems. ACO algorithms are suitable for scheduling, Routing and Graph designed problems. ABC is widely used in scheduling, classification and clustering algorithms.

In optimization problems the scope of the field is very vast. If the problem is not well-formulated for optimization, poor performance will be the result. There is no guarantee to get an optimal solution in finite amount of time. Expensive computation is also needed. Result mainly depends on the parameter setting. Scalability and performance evaluation are the major difficulties. Even though, some algorithms are stated as suitable for particular class of problems, no one optimization algorithm was proved as best one for particular problem.

Proper design of problems, self-adaption of parameters and hybrid optimization will improve the results with fast convergence that can reduce the computational needs for hard optimization problems.

## CONFLICT OF INTEREST

The authors declare no conflict of interests.

## ACKNOWLEDGEMENT

None

## FINANCIAL DISCLOSURE

The authors report no financial interests or potential conflicts of interest.

## REFERENCES

- [1] Binitha S, S Siva Sathya.[2012] A Survey of Bio inspired Optimization Algorithms, *International Journal of Soft Computing and Engineering (IJSCE)*, 2(2): 2231-2307.
- [2] KarthikSindhya.[ 2014]An Introduction to Nature Inspired Algorithms,.
- [3] Michalewicz Z, Fogel DB.[ 2004] How to Solve It: Modern Heuristics, Springer.
- [4] Goldberg DE.[1989] Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley.
- [5] Acharya DP, Panda G, Mishra S, Lakshmi YVS.[2007] ‘Bacteria foraging based independent component analysis’, In *IEEE International Conference on Computational Intelligence and Multimedia Applications*, 2: 527-531.
- [6] Miettinen K,Neittaanmaki P.[1999] Evolutionary Algorithms in Engineering and Computer Science: Recent Advances in Genetic Algorithms”, Evolution Strategies, Evolutionary Programming, GE ,John Wiley & Sons, Inc. New York, NY, USA,1999, ISBN:0471999024
- [7] Beyer HG, Schwefel HP. [2002]Evolution Strategies A Comprehensive Introduction, *Natural Computing*,1:3-52, Kluwevr Academic Publishers.
- [8] Schaffer JD, Caruana RA, Eshelman, LJ, Rajarshi Das. [1989]A study of control parameters affecting online performance of genetic algorithms for function optimization”, In Proceedings of the third international conference on Genetic algorithms, , pp 51–60. Morgan Kaufmann Publishers Inc., San Francisco.
- [9] Ant Colony optimization. <[http://en.wikipedia.org/wiki/Ant\\_colony\\_optimization\\_algorithms](http://en.wikipedia.org/wiki/Ant_colony_optimization_algorithms)>[Online][2015].
- [10] Karaboga D, Basturk B.[ 2007] A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm, *Journal of Global Optimization*, 39: 459–471.
- [11] Karaboga D, Akay B. [2009] A comparative study Of Artificial Bee Colony algorithm, *Applied Mathematics and Computation*, 214: 108–132.
- [12] C. Rajan, N. Shanthi, Genetic based Optimization for multicast Routing algorithm for Manet’ Sadhana - Academy Proceedings in Engineering Science, 40 (7): 2341-2352, 2015.
- [13] PSO Tutorial’, Available in [www.swarmintelligence.org/tutorials.php](http://www.swarmintelligence.org/tutorials.php)