Role of Swarm Intelligence based Algorithms and their Applications for optimization in Software Reliability

Narender, Sona Malhotra

Abstract**:** ***The software has many features like functionality, maintainability, serviceability, usability, quality, performance. The reliability of the software is an imperative characteristic of software that leads to the eminence of the software. Software reliability is a great concern for software producers as well as users of the software. Keeping this concern in mind, there are already hundreds of software reliability models developed in the last four decades. This paper evaluates different algorithms based on Swarm intelligence in the way of optimization in software reliability. There are a number of swarm intelligence based algorithms that already have been used to improve the efficiency of the reliability of the software. Some of them are ant colony optimizer method (ACO), particle swarm optimizer method (PSO), artificial bee colony optimizer (ABC), bat algorithm, fish swarm algorithm, cuckoo search, bird flock algorithm. Still, there are so many algorithms based on Swarm intelligence that has not been used in this area. This paper investigates some known swarm intelligence based algorithms and their applications for optimizing software reliability.***

Index Terms**:** ***ABC, ACO, Bat, Fish, PSO, Software reliability optimization, Swarm intelligence.***

I. INTRODUCTION

A. Importance of Software Reliability

In today's world every minute a new technology is being introduced. Technologies are competing with each other. In fact, the developers of these technologies coming with new ideas for a similar kind of product. One can easily experience the advancement in the newer version of an application. The performance of any application or software is praised only when the performance is consistent and faultless. It means the application or software needs to be reliable to retain its name in the market.

Software reliability is measured by the nonstop function of a computer program/application, meant for a particular time, in a certain situation [1]. Software reliability is a significant feature of the software along with functionality, maintainability, serviceability, performance, usability that leads to the excellence of the software [2]. The reliability of the Software is an important factor need to be considered by the developer of the software because it decides the future of the product.

Revised Manuscript Received on 30 July 2019.
***** Correspondence Author

Narender, Ph. D Scholar*, Computer Engineering, Kurukshetra University/ UIET, Kurukshetra, Haryana, India,

Sona Malhotra, Assistant Professor, Computer Engineering, Kurukshetra University/ UIET, Kurukshetra, Haryana, India,

B. Swarm Intelligence

Swarm intelligence is the obedience that centers on the combined behavior of individuals, their local interaction with each other and with their environment [3]. There are a number of algorithms based on swarm intelligence that have been used in optimization problems from last two decades, still there are many algorithms that are to be used in future.

C. Various approaches in Swarm Intelligence

Ant Colony Optimizer (ACO): This algorithm is a technique to measure the likelihood of an event for solving computational problems [4]. Firstly, this was used to obtain the shortest distance through a graph. The algorithm uses artificial ant as a multi-agent. This algorithm has taken the inspiration from the actions of real ants that is how they maintain a path between the source of food and their colony. Each ant moves at random leaving pheromone behind it. Pheromone is accumulated on the trail. As new pheromone is deposited the frequency of the path being followed increases. In this way, the shortest path is found through pheromone trails.

Artificial Bee Colony (ABC): This algorithm uses the foraging activities of bees and searches for the best solution in various optimization problems [5]. Food source (flower) is treated as a candidate solution. A colony of artificial bees that are treated as an agent is used to search the solution space. When an artificial bee lands to a flower it checks its fitness.

Particle Swarm Optimizer (PSO): This is a mathematical calculation technique that improves efficiency of an issue by iteratively tries to find the best fit among a given set of solutions [6]. It looks at the problem as a population of a candidate solution, say particle. Each particle has some velocity at some position in search space. the movement of each particle depends upon the home best position of itself. This local best keep updated in the search space until some other best fit is found in search space. This leads to moving the swarm to the best fit.

Bat Algorithm: This technique also belongs to the metaheuristic algorithm that was designed for optimization problems. Each bat flies with some random speed at someplace with some loudness and frequency, the same concept is applied to build this optimization algorithm. When it discovers its target it modifies its loudness and frequency. Search is optimized by local arbitrary walk. The choice of best fit goes on until some stopping criterion is met [7]. The dynamic behavior can be controlled by frequency matching technique.

Fish Swarm Optimizer: This is an optimization algorithm that is based on population. The logic for the development of algorithm is taken from the combined behavior of the fishes, that is their social behavior, immigration, search for food, how to escape from danger and the coordination from the other fishes [8].

II. BACKGROUND

A. Ant Colony Optimizer Algorithm

ACO is a metaheuristic technique for problem-solving that comes into a conventional optimization problem-solving algorithm. It was noticed in 1997 by Dorigo and Garandella. As earlier said ants act as agents in this algorithm. They have the capability to search for their food by traveling minimum distance. The basic concept behind this is a pheromone trail released by ants while searching for food. Suppose ‘a’ is a nest and ‘b’ is a food source for an ant. Ant moves form ‘a’ to ‘b’ by releasing a pheromone trail, other ants sense that pheromone and follow the same path. The pheromone density gets thicker by the repetition of the path by a number of ants [9]. This path results in the direct path among shell and food resources. This is the first case and the other is when there is an obstacle between the shell and food resource.

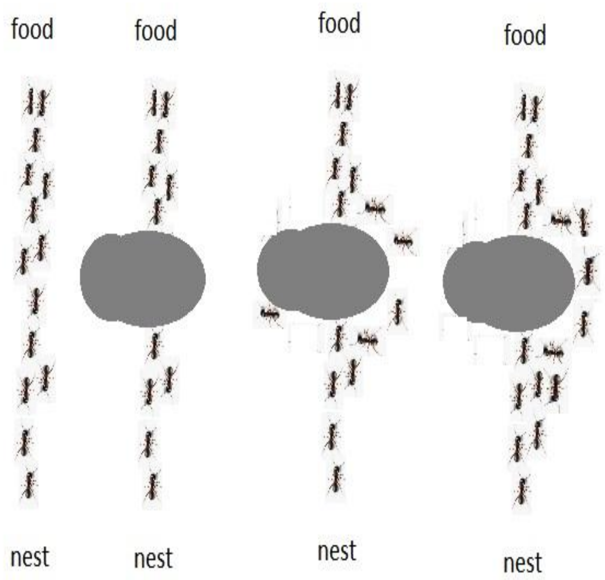

Fig 1. Natural behavior of Ants

Referring to the figure, some ants follow the one path with respect to obstacles and others follow the other path with respect to obstacles. An ant takes less time using the path with less distance. As earlier said the pheromone has the property of evaporation with time. The density of the pheromone on short paths gets thicker with time. The density of the pheromone keeps decreasing on the longer path [10]. The probability of k_{th} ant from ‘a’ to ‘b’ is given by

$$P_k^{ab} = ([I_{ab}]^{\alpha}[V_{ab}]^{\beta}) / \sum z \in allowed(I_{az})(V_{az}) \quad (1)$$

P_k^{ab} = The probability of K_{th} ant to move from ’a’ to ‘b’.

I_{ab} = Intensity of pheromone or amount of pheromone deposited.

V_{ab} = Visibility of ‘b’ form ‘a’.

$allowed$ = heuristic value that indicates the visibility from the current node.

I_{az} = attractiveness of node ‘z’ from ‘a’.

V_{az} = pheromone density for other state transition.

α, β = parameter to regulate the power of heuristic and pheromone.

Once the ants have visited their path, the trails update is given by

$$I_{ab} \leftarrow (1-\rho)I_{ab} + \Sigma \Delta I_k^{ab} \quad (2)$$

$$\Delta I_k^{ab} = \begin{cases} Q/L_k & \text{if } k_{th} \text{ ant uses path ab in its travel} \\ 0 & \text{rest of the case} \end{cases} \quad (3)$$

Here

I_{ab} = the amount of pheromone deposited for transition from a to b

$\rho =$ the pheromone evaporation coefficient and

$\Delta I_k^{ab} =$ the amount of pheromone deposited by k_{th} ant

L_k = is the cost of the k_{th} ant's tour and Q is a constant.

B. Artificial Bee Colony Algorithm

It is a swarm-based meta-heuristic optimization method. Bees collect the food as nectar and accumulate for future use. A number of bees in group search for food. Once a food resource is found by a bee it returns to the hive and tells other bees for location, quantity, and quality of food source. It uses different movements to tell all these things. For example: dancing around the hive, number of rounds to the left or right side of the hive and with some ups and downs. Different movements have different meanings. In this way, they encourage the remaining bees to follow her. Some recruited numbers of bees follow the scout. This process continues until some better food source is found.

The algorithm treats the food sources as an individual solution in the search area. The food source with high quality of food is treated as the best fit solution [11], and can be given by the equations.

$$f(X_i) = \begin{cases} 1 & MSE_i = 0 \\ & \end{cases}$$

(4)

$1/[MSE_i+1] \quad MSE_i>0$

Where

$i=1$ to N_s;

MSE_i mean square error for the i_{th} solution.

Let us suppose, the count of foremost bees (N_l) equals to the count of recruited bees to follow (N_r), and equals to the count of food source (N_s) in a particular search space. Follower choose the food resource based on the probability P_i, that can be calculated as

$$P_i = f(X_i) / \sum_{n=1}^{N_s} f(X_n) \quad (5)$$

Leaders and followers carry out local search by,

$$V_{in} = X_{in} + rand(-1,1)(X_{in} - X_{mn}) \quad (6)$$

Here,

i = count of the solution, $m \in \{1,2,\ldots,Ns\}$, is arbitrarily produced, and $i \neq m$. $n \in \{1,2,\ldots,D\}$, where $D<N_s$.

The goodness of the leader ant is calculated, and if it not the present best fir for Ni, the same is dumped and substituted with the food resource arbitrarily hunt by the scouters. Assume that the discarded solution is X_i, the new solution can be generated as,

$$X_i = X_{minimum} + rand(0,1)(X_{maximum} - X_{minimum}) \quad (7)$$

C. Particle Swarm Optimizer Algorithm

This is a assumptive optimization method supported by the activities and intelligent actions of the swarms. The concept of PSO is totally dependent on the working mechanism of natural phenomena. In PSO swarm of n particles communicate directly or indirectly using gradients. They move into a hunting space to find the global best. They update their position with their previous best and neighbor's current best [12].

The X contains the present location of the particle.

$$X_i(t+1) = X_i(t) + V_i(t+1) \quad (8)$$

$$X_{ij}(t+1) = X_{ij}(t) + V_{ij}(t+1) \quad (9)$$

Where, $X_i(t)$ is the current position of i_{th} swarm at time t.

$X_i(t+1)$ is the position of the i_{th} swarm at time $t+1$

$V_i(t+1)$ is the velocity of the swarm at time $t+1$

The V contains the direction for the particle to travel in.

$$V_i(t+1) = W * V_i(t) + A_1(P_i(t) - X_i(t)) + A_2(g(t) - X_i(t)) \quad (10)$$

$$V_{ji}(t+1) = W * V_{ji}(t) + r_1 A_1(P_{ij}(t) - X_{ij}(t)) + A_2(g_j(t) - X_{ij}(t)) \quad (11)$$

Where $W*V_{ij}(t)$ is inertia term

$P_i(t)$= i_{th} particle's individual best

$g(t)$= overall best for the entire particles in the swarm

A_1, A_2 = coefficients of acceleration.

r_1 and r_2 are the arbitrary constants.

D. Bat Algorithm

This was a meta-heuristic technique, enthused by the echo actions of bats, by changing loudness and pulse rate of emissions. Bats generate the signal to locate the potential prey. The signals spring back if it strikes an entity. Bats can identify the movement and size of the object by interpreting the signal [13].

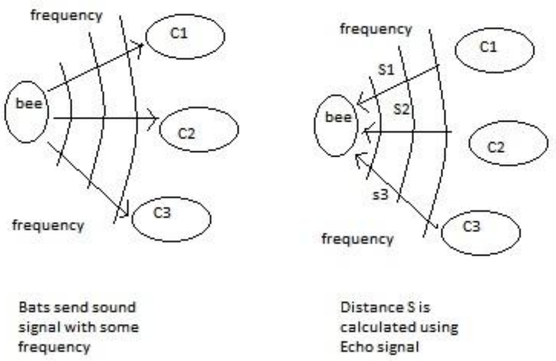

Fig 2. Behavior of bats

Bats move arbitrarily in the search of food with some velocity and frequency at a particular position. Suppose the velocity is vi, frequency is f, the wavelength is l and loudness is L while searching for a food resource. Bat has the ability to automatically alter the frequency/wavelength of its generated pulse and regulate the speed r∈ [0,1], with respect to the position of the prey. Frequency velocity and position can be found by the following equations respectively:

$$f_i = f_{minimum} + (f_{maximum} - f_{minimum})\alpha \quad (12)$$

where $\alpha \in [0,1]$

$$V_i^{t+1} = V_i^t + (X_i^{t-1} - X_{best})F_i \quad (13)$$

X_{best}=current global best location

$$X_i^{t+1} = X_i^t + V_i^t \quad (14)$$

where, t=number of iteration

E. Fish Swarm Optimizer Algorithm

Fishes are the candidate solution to the problem in the optimization process. Fishes move in the hunt of food. Food density can be treated as the goal function that is to be improve. The weight of the fishes can be considered as the accumulation of food and it can be treated as the fitness function. The direction of the fishes is influenced by the individual execution of the movement resulted in the goodness of fit [14]. Fishes update their position by

$$X_i^{t+1} = X_i^t + r * W \quad (15)$$

Where X_i= final position of the fish $_i$.

r= random generator

W= weighted parameter

The Weight of the fishes can be updated by

$$W_i^{t+1} = W_i^t + dF_i / \max(dF_i) \quad (16)$$

Where W_i = the weight of the fish i at t.

dF_i is the difference between current and new goal function.

The resulting direction can be

$$V^t = \sum_{i=1}^{n} (dX_i * dF_i) / \sum_{i=1}^{n} dF_i \tag{17}$$

III. OBSERVATIONS OF RESEARCH PROBLEM

Table I: Analysis of different Swarm intelligent based algorithms

Algorithms	Pros	Cons	Observations
ACO	-Efficient in solving TSP problems and other discrete problems -Fast	-Premature convergence -Not effective for continuous problems	-Accuracy of reliability significant -Not suitable for all kind of problems
ABC	-Flexible, Easy implementation -Ability to explore local solutions -Broad applicability, complex functions	-New tests might be calculated for new parameters of the algorithm. -Objective function evaluation frequency is high	-Accuracy of reliability is not that much -Suitable for most of the problems
PSO	-Easy implementation -Fast -Higher probability and efficiency in finding the global optima	-Can be trapped into a local minima specially in complex problems -Difficult to design initial design parameters	-Accuracy of reliability is significant -Not suitable for all kind of problems
Bat	-Flexible, Easy implementation -Can solve non-linear problems efficiently -Works well with complicated problems -Gives optimal solution	-Allowance of switching from exploration to exploitation by changing pulse and loudness may result to stagnation after some initial stage.	-Accuracy of reliability is high, but sometimes becomes inactive -Suitable for most of the problems
Fish Swarm	-Gives optimal solution -One of the most appropriate method	-Falling in local optimum -Time consuming -Advanced convergence	-Accuracy of reliability is high -Not suitable for all kind of problems

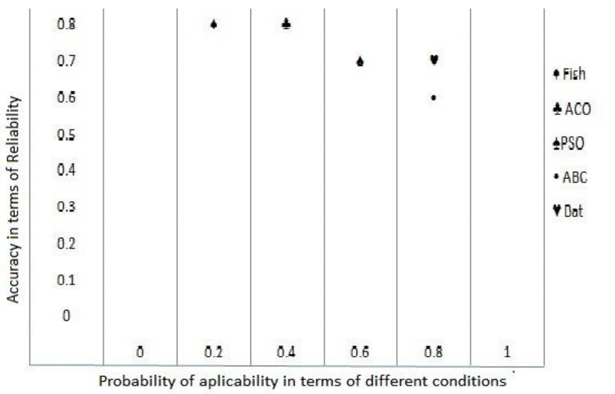

Fig 3: Analysis of different Swarm intelligent based algorithms

IV. CONCLUSION

This paper searches scopes for finding the accuracy in reliability and applicability to different kinds of problems by the different swarm intelligent algorithms. The paper include the observations and evaluation of various swarm intelligence algorithms that tells how a fitness function can be obtained for an optimization problem and can be used in calculating the reliability of the software. Observation also tells whether an algorithm is applicable to all kinds of problems or not. It also creates a relation between accuracy of the reliability and applicability of the model to different kinds of problems in different environments. This study can help the researchers in identifying the algorithm that can best suit a particular kind of problem.

REFERENCES:

1. Zygmunt Jelinski, P. Moranda, "Software reliability research", Statistical computer performance evaluation (Elsevier), pp 465-484, 1972.
2. Deepshikha Jamwal, "Analysis of software quality models for organizations", International Journal of Latest Trends in Computing, Volume 1, Issue 2, pp 19-23, 2010.
3. J. Derrac, D. Molina, F. Herrera, "A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms", Swarm and Evolutionary Computation, Elsevier, Volume 1, Issue 1, pp 3-18, 2011.
4. Marco Dorigo, Christian Blum, "Ant colony optimization theory: A survey", Theoretical computer science, Elsevier, Volume 344, Issue 2-3, pp 243-278, 2005.
5. Dervis Karaboga, Bahriye Akay, "A comparative study of artificial bee colony algorithm", Applied mathematics and computation, Elsevier, Volume 214, Issue 1, pp 108-132, 2009.
6. Yuhui Shi, "Particle swarm optimization: developments, applications and resources", Proceedings of the 2001 congress o evolutionary computation, IEEE, Volume 1, pp 81-86, 2001.
7. Xin-She Yang , "A new metaheuristic bat-inspired algorithm", Nature inspired cooperative strategies for optimization (NICSO 2010), Springer, pp 65-74, 2010.
8. X. L. Li, JX Qian, "Studies on artificial fish swarm optimization algorithm based on decomposition and coordination techniques", Journal of circuits and systems, Volume 1, pp 1-6, 2003.
9. Marco Dorigo, Thomas Stuzle, "The ant colony optimization metaheuristic: Algorithms, applications, and advances", Handbook of metaheuristics, Springer, pp 250-285, 2003.
10. Chen Zhiming, S. Zhou, J. Luo, "A robust ant colony optimization for continuous functions", Expert system with applications, Elsevier, Volume 81, pp 309-320, 2017.
11. Dervis Karabogas, Bahrive Basturk,"On the performance of artificial bee colony (ABC) algorithm", Journal of Applied soft computing, Elsevier, Volume 8, Issue 1, pp 687-697, 2008.
12. Bai, Qinghai, "Analysis of particle swarm optimization algorithm", Journal of Computer and information science, Canadian Center of Science and Education, Volume 3, Issue 1, pp 180, 2010.
13. Xin-She Yang, "Bat algorithm: literature review and applications", International Journal of Bio-Inspired Computation, Vol. 5, No.3, pp. 141-149, 2013.
14. Bastos Filho, F. B. De Lima Neto,"A novel search algorithm based on fish school behavior", International Conference on Systems, IEEE, Man and Cybernetics, pp 2646-2651

AUTHORS PROFILE

Narender He is an assistant Professor in Manav Rachna University, Faridabad, Haryana, India. He is persuing his Ph. D. in Computer Engineering form UIET, Kurukshetra University, Kurukshetra, Haryana, India. He has done his masters in Computer Engineering form YMCA Institute of Engineering, Faridabad, Haryana, India (now known as J C Bose university of Science & Technology (YMCA). He has done his bachelor's in Computer Engineering from Ch. Devi Lal memorial Engineering College, Panniwala Mota, Sirsa, Haryana, India (now known as Ch. Devi Lal State Institute of Engineering Technology). His area of research is Software Reliability.