ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol3, Issue 5, May 2016

# Test Case Selection Using Nature Inspired Algorithms

[1] Hemalatha T, [2] S Rohini, [3] D Vivekananda Reddy
[1] Department of CSE, SVUCESri Venkateswara UniversityTirupathi-517502
[2] Assistant Professor Department of ECE, AITSTirupati.
[3] Assistant ProfessorSri Venkateswara UniversityTirupathi-517502

***Abstract*— In software development life cycle (SDLC), testing phase is the most important phase. Without testing we can't give quality software or risk free software to the client. Software testing process typically consumes at least 50% of the total cost involved in software development. In regression testing there evolves the number of test cases. Due to some constraints, it is impractical to test all of them. Therefore to overcome this problem, testing is done using selected test cases to reduce the testing effort and get the desired result accurately. In this paper, the algorithms which are discussed imitate the processes running in nature. And due to this these process are named as "Nature Inspired Algorithms". The algorithms inspired from human body and its working and the algorithms inspired from the working of groups of social agents like ants, bees, and insects are the two classes of solving such Problems. This emerging new era is highly unexplored young for the research. The Algorithm proposed here uses swarm based intelligence technique on Maximum coverage to select the best of breed test case to evaluate any errors in the development cycle therefore the BCO produce optimal no. of test cases solving optimization problem.**

***Keywords* - Swarm Intelligence; Meta Heuristic; Ant Colony Optimization; Bee Colony Optimization; Regression Testing.**

## I. INTRODUCTION

Software testing is one of the important process in Software Development Life Cycle. As specified by Glen Myers," Testing is a process of executing a program with the intent of finding an error". Maintenance is required when some of the components of software need to be replaced, software is upgraded or enhanced to include additional features and to provide more services. ." Regression testing is defined as "the process of retesting the modified parts of the software and ensuring that no new errors have been introduced into previously tested code". Testers might re-run all test cases generated at earlier stages to ensure that the program behaves as expected. It is a cost effective and time consuming process, and Test suite is a collection of test cases that is used to test a software program to show that it behaves as expected. However, as the software evolves, test suite grows larger for which it is practically impossible to retest all the test cases. Hence Test suite optimization has to be done. Test suite optimization is a process of generating effective test cases in a test suite that can cover the given SUT (System Under Test) within less time [4]. Optimization can be defined as the process of finding the conditions that give the maximum or minimum value of a function. Goal is to either minimize the effort or maximize the desired benefit. Most of the classical optimization problems require different types of variables, objective and constraint functions in their formulation. Different types of classical optimization techniques are Linear Programming, Non linear Programming, Geometric Programming, Integer Programming and Stochastic Programming.

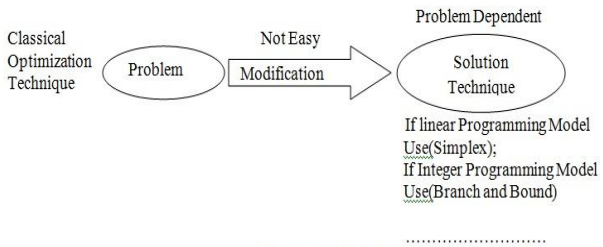


FIG 1: Classical Optimization Technique

Classical optimization algorithms are generally depended on the type of objective and constraint functions (linear, non-linear etc.) and the type of variables used in the problem modeling (integer, real etc.). Their efficiency is also very much dependent on the size of the solution space, number of variables and constraints used in the problem modeling, and the structure of the solution space (convex, non-convex, etc.). So they are inefficient in solving larger scale combinatorial and/or highly non-linear problems. One of the main characteristics of the classical optimization algorithms is their inflexibility to adapt the solution algorithm to a given problem. Generally a given problem is modeled in such a way that a classical algorithm like simplex algorithm can handle it. This generally requires making several assumptions which might not be easy to validate in many

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol4, Issue 4, May 2017

situations. Efficiency also dependent on size of solution space, num of variables and constrains used. In order to overcome these limitations more flexible and adaptable general purpose algorithms are needed.

So, a new set of problem and model independent nature inspired heuristic optimization algorithms are developed. Nature inspired computing is the computing which has its foundation in the biological components of the nature i.e., humans and animals. Nature has four powerful features which are basic building blocks are self optimization, self healing, self learning and self processing. Nature as Self optimizer is that it can automatically manage its resources in an efficient manner to meet enterprise need. Nature as self healer is as the components of nature on seeing any problem finds a solution and come out of it. Self learning and self processing are two related terms. They go hand in hand and moved together. Nature and its components self processes the changing conditions in the environment learn from the past and present conditions to evolve in the changed environment in natural evolution. As the individuals of nature have the capability to evolve according to the changing environment and solve highly complex problems as nature does. To fulfill this desire, we want our algorithms to adopt the techniques and features from nature and become more effective and efficient too. These algorithms model a given problem as close as to reality.

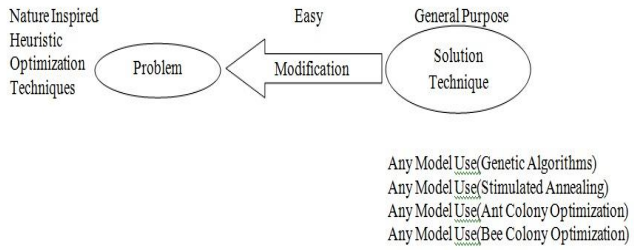


FIG 2: Modern Heuristic Optimization Technique

These techniques are efficient and flexible. To formulate a real life problem that suits a specific solution procedure, it is necessary to make modifications on the original problem parameters (rounding variables, softening constraints etc.,). This effects the solution quality. Various techniques have been proposed by researchers on regression test case selection using Meta heuristic techniques.

Meta heuristic is a higher-level procedure designed to generate a lower-level procedure or heuristic (partial search algorithm) [12]. These techniques provides us with the greater advantage of lesser execution time while making some negotiation on the solution obtained i.e. it provides the near optimal solution. They grant sufficiently good solution to an optimization problem especially with partial or inadequate information or limited computation capacity. Here, "heuristic" means search, that is, there is some learning component which is guiding the whole search process. Various Meta heuristic approaches have been studied to examine the potential benefits to regression testing.

Meta heuristic techniques can be classified into different groups depending on the criteria being considered, such as population based, iterative based, stochastic, deterministic, etc. Population based approach maintain and improve multiple candidate solution, often using population characteristics to guide the search. Important groups of population based algorithms are Evolutionary Algorithms (EA) and Swarm Intelligence (SI) based algorithms.

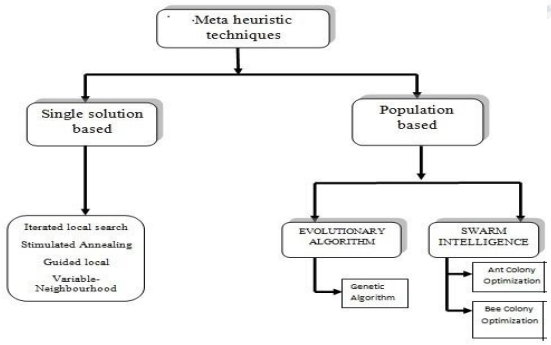


**FIG 3: Meta Heuristic Techniques**

## II. RELATED WORK

***Evolutionary Algorithms***

In the Origin of Species, Charles Darwin stated the theory of Natural Evolution. Over many stages of life, biological organisms develop according to the principles of natural selection like "Survival Of The Fittest" to attain some astounding form of accomplishment [3]. The best example of natural evolution can be seen in generation of human beings. So if it works so admirably in nature, it should be interesting to imitate natural evolution and try to procure a technique which may solve existing search and optimization problems.

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol4, Issue 4, May 2017

***Genetic Algorithm (GA):***

Genetic algorithm is directed search algorithm based on mechanics of biological evolution, introduced by John Holland broadly studied by Goldberg and De Jong. It is an optimization technique which provides near optimal solution to NP-hard problems.

Genetic Algorithm is based on the idea on the natural evolution. The foundation of GA lies on the concept of the survival of fittest into a solution space. GAs work with three operators:

a. Selection: Picking up of two chromosomes with higher fitness value from population for crossing.

b. Crossover: producing a child (a new better solution) by taking two parent solutions. Crossover is a recombination operator that proceeds in three steps:

- The reproduction operator chooses at random a pair of two individual strings for the mating.
- A cross site is then decided at random along the string length.
- Finally, the selected position values are swapped between the two strings following the cross site.

***C.Mutation:*** For prevention of the algorithm to be trapped in a local minimum. Its use is to change the value of a random position in a string. Lastly the new generation which is evolved if contains a solution that produces an output to the problem that is very close to the desired result would be actual solution. If the problem does not reach to its solution i.e., after whole procedure the solution is not obtained, then the new generation goes through the same procedure as their parents did. This will continue until a solution is achieved that satisfies the minimum criteria or a fixed number of generations.

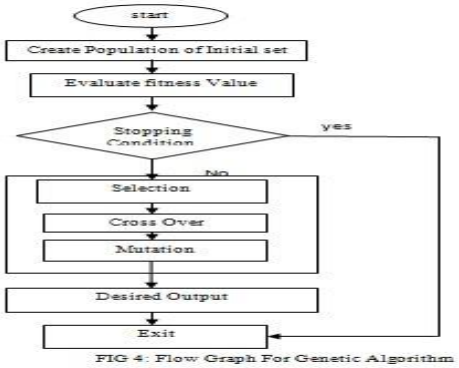

FIG 4: Flow Graph For Genetic Algorithm

***Pseudo code of Genetic Algorithm***

1 .Randomly initialize population(t)
2 .Determine fitness of population(t)
3. Repeat
    i.Select parents from population(t)
    ii.Perform crossover on parents creating population(t+1)
    iii.Perform mutation of population(t+1)
    iv.Determine fitness of population(t+1)
4. Until good solution is found

***Demerits of Genetic Algorithm:***

- Genetic Algorithms has difficulties in giving stable results (stuck up at local optima)
- Not so effective to solve problem involving single measure for evaluation.
- Convergence is slow and has non-explicit memorization of best individuals.

***Swarm intelligence:***

Gerardo Beni and Jing Wang introduced it. Swarm intelligence can be described as the collaborative conduct of a group of animals, especially insects such as ants, bees and termites, that are each following very basic rules but when seen in the field of computer science, swarm intelligence is a simulated way to problem solving using algorithms formed on the concept of self managed collective behavior of social insects. Bonabeau [3] gave a simple definition for better understanding of swarm intelligence i.e., "the popular way of simple and common intelligence of social agents". Swarm intelligence is an emerging new domain that visualizes intelligence as a method of communication between independent agents.

To mimic the performance evolved from a swarm several general postulates for swarm intelligence were defined:

1.***Proximity Principle***: The set of solutions should be able to execute space and time effective computations.
2.***Quality Principle:*** The set of solutions should be able to interact with the quality features of its environment.
3.***Diverse Response Principle***: The set of solution
Should not commit its activity along excessively
Narrow channels.
4.***Stability Principle:*** The set of solutions should not change their actions according to changing conditions.

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol4, Issue 4, May 2017

5.***Adaptability Principle:*** The set of solution should be able to change their actions when effective space and time computational price is needed.

***Ant Colony Optimization (ACO):***

For finding food, ants start out from their colony and move randomly in all directions. Once a ant find food, it returns to colony and leave a trail of chemical substances called pheromone along the path. Other ants of the swarm can sense pheromone trails and move on the same path. The interesting point is that how often the path visit by ants is determined by the concentration of pheromone along the path. Since pheromone will naturally evaporate over time, the length of the path is also a factor. Therefore under these conditions, a shorter and best path will be chosen because ants moving on that path keep adding pheromone to it which makes the concentration strong enough against evaporation. As a result, the shortest and best path from colony to food emerges. This type of interaction between social agents is known as stigmergy. Stigmsergy is a mechanism of indirect coordination between agents or actions. The law of stigmergy says that the indications left in the environment of current action simulate further the next subsequent actions [13]. This whole process of simulation in last creates a systematic way of performing some activity.

Ant colony optimization algorithm takes its inspiration from the real world of ant colonies to solve optimization problems. It was first introduced by Marco Dorigo. It is based on the behavior of ants in search of food sources. Artificial ants leave a virtual trail accumulated on the path segment they follow. The path for each ant is selected on the basis of the amount of "pheromone trail" present on the possible paths starting from the current node of the ant.

***ACO algorithm works in this way:***

1.***Initialization:*** Initialize ACO parameters (e.g., no. of artificial ants) and pheromone trails.

2.***Construct Ant Solutions***: A set of some artificial ants construct solutions from elements of a finite set of available solution components.

3.***Daemon Actions:*** When solution set has been created, before increasing or decreasing pheromone values, some particular actions are to be taken according to the problem given. These actions are known as Daemon Actions. These actions vary problem to problem. The Daemon Actions are optional.

4.***Update Pheromone:*** Update pheromone is the mechanism of increasing or decreasing the pheromone values along the path. The pheromone values are increased for good solutions and decreased for bad solutions. There is another procedure which is executed inside update pheromone action which is evaporation. The pheromone values are evaporated so that bad solutions which were learned earlier can be forgotten.

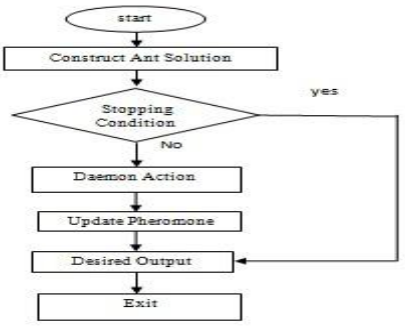


FIG 5 : Flow Graph for ACO

***Pseudo Code for ACO***

```
Procedure ACO
    While (termination condition is not satisfied)
            generateSolutions()
            daemonActions()
            pheromoneUpdate()
    end while
end Procedure
```

***Demerits of Ant Colony Optimization:***

- ACO wastes large amount of computing resource to generate test cases.
- Convergence is guaranteed but to convergence is uncertain.
  Has higher length of test sequences.
- Repetition of nodes within the same sequence.
- Two ants started at an initial node, and during random selection of next node, they will go to the same next node. Since the process is random.

***Bee colony optimization:***

Dervis Karaboga defined BCO, which is swarm based meta-heuristic algorithm for optimizing problems [2]. It is relatively a new member of swarm intelligence and a nature-inspired Meta heuristic, which imitates the foraging behavior of bees. The Bee Colony Optimization is capable to combinatorial problems, as well as combinatorial

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol4, Issue 4, May 2017

problems characterized by uncertainty. It is a stochastic technique which is easy to implement, has fewer control parameters, and could easily be modify and hybridized with other Meta heuristic algorithms.

***Behavior of bees in nature:***

*Bee system consists of two essential components:*

1.***Food Sources***

The value of food source depends on different parameters such as proximity (closest in time and space) to the nest, richness of energy and ease of extracting this energy.

2.***Foragers***

There are 3 different type of foragers:

***a.Unemployed Foragers:***

If it is assumed that a bee have no knowledge about the food sources in the search field, bee initializes search as an unemployed forager. There are two types of unemployed forager

***Scout Bee:***

If the bee starts searching spontaneously without any knowledge it will be scout bee.

***Recruit Bee:***

If the unemployed forager attends to a waggle dance done by some other bee, then that bee will start searching by using the knowledge of waggle dance.

***b. Employed Foragers:***

When the recruit bee finds and exploits the food source, it will raise to be an employed forager who memorizes the location of food source. After the employed foraging bee loads a portion of nectar from the food source, it returns to the hive and unloads the nectar to the food area in the hive.

- There are three possible options available related to residual amount of nectar for the foraging bee If the nectar amount decreased to low level, foraging bee abandons the food source and become an unemployed bee.
- If there are still sufficient amount of nectar in the food source, it can continue to forage without sharing the food source information with the nest mates.
- Or it can go to the dance area to perform waggle dance for informing the nest mates about the food source.

*C.**Experienced Foragers:***

These types of foragers use their historical memories for the location and quality food sources.

- It can be an **inspector** which controls the recent status of food source already discovered.
- It can be a **reactivated forager** by using the information from waggle dance, it tries to explore the same food source discovered by itself if there are some other bees confirm the quality of same food source.
- It can be **scout bee** to search new patches if the whole food source is exhausted.
- It can be **recruit bee** which is searching a new food source declared in dancing area by another employed bee.

In BCO, a colony of artificial forager bees (agents) search for rich artificial food sites (good solutions for a given problem). Then, the artificial bees randomly discover a population of initial solution vectors and then iteratively improve them by employing the strategies: moving towards better solutions by means of a neighbor search mechanism while abandoning poor solutions. Employed bees are associated with a particular food source which they are currently exploiting. Bees randomly search for food positions with the higher amount of nectar. Once the bees find such position, they go back to the hive and communicate about the food source position. The most important part of the hive is dancing area where the bees exchange information. The related dance is called waggle dance. If food source position is close to the hive, the bees perform waggle dance else bees perform the round dance. Waggle dance is eight like figure. The bees have capability of memorizing the location of food sources with the higher nectar amount. Once such food source is discovered, the bees start exploiting it. Hence, they become the employed forager.Hence the Meta heuristic techniques implemented by various researchers for regression testing have been compared and verified. When compared to Genetic Algorithm and Ant Colony Optimization Bee Colony has higher processing speed and so used for test case selection.

## III . PROPOSED STRATEGY:

The Proposed strategy is used to solve the problem of Test Case Optimization using BCO Algorithm [12]. The aim of proposed approach is to find the best food source (Test case with maximum coverage). The positions of food sources represent the solutions of the optimization problem

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol4, Issue 4, May 2017

and the amount of food which defines the fitness or quality of the solution.

***The Main steps of BCO Algorithms are:***

a. Place the Employed bees on food sources and measure their amount of food.
b. Calculate the probability value of food source with which they are selected by Selector bee.
c. Stop the exploitation process of food source unused by the bees.
d. Sent the scout bees for discovering new food source to search area randomly.
e. Remember the best food source found so far.

In this algorithm bees find food sites (test cases) then calculate their quality (fitness) used to identify the best test case with max coverage and less computing resources.
Model consists of three essential components

- Employed bees
- Unemployed bees
- Food sources

First two components employed foragers and unemployed foragers search for rich food sites. Third component food sources are close to their hive. Model defines two leading modes of behavior which are necessary for self organizing and collective intelligence

- Recruitment of foragers to rich food sources resulting in positive feedback.
- Abandonment of poor sources by foragers causing negative feedback.

In BCO, a colony of forager bees search for rich artificial food sites like wise here agent's searches for good solution for a given problem.

## VI. PRINCIPLE AND WORKING OF PROPOSED APPROACH

***The main Principle and working of proposed approach is:***
BCO approach is a Meta heuristic population based approach. Solution of optimization problem is represented by each test case. The quality of each test case is calculated by the fitness value of problem. The work proposes that by using the algorithm we generate the optimize test cases and will contain all possible independent paths along with its test data. The test data will be the required input to be given to the program.

- Initially we make the program corresponding Control Flow Graph.
- From control flow graph, the different independent paths are generated.
- Each independent path would comprise number of normal nodes and predicate nodes. Every independent path would represent a Test Case.

Now the BCO algorithm is applied to generate a test suite of selected test cases which would passes through the independent paths and hence into to the test cases.

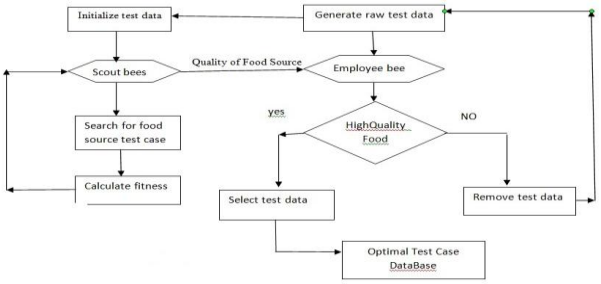


***Fig 6: Flow Diagram for Bee Colony Optimization***

In the figure, here the scouts' bee acts as search agent which searches for the execution state of the test cases also initializes the test cases with the initial test data with the help of white box testing techniques. Then the bee calculates the fitness value of each test case by computing the coverage of each node. This process is repeated till we found executable state. Then the employee's bee passes the fitness value of the processed nodes to the onlooker agent. The employee bee compares the fitness value of nodes with the fitness value of the neighboring nodes. If the fitness value of node is found greater than the neighboring fitness value, then the node's information is stored in the optimal test case database. The node whose fitness value is found less is stored in the abandoned repository. New Test data are again generated by scouts' bees from the abandoned repository and again the same process of employed bee is repeated. The algorithm for test case optimization using Bee Colony optimization approach is presented in the further section.

## V. ALGORITHM STRUCTURE OF BCO

The general algorithmic structure of the BCO optimization approach and pseudo code is given as follows:
Initialization Phase
REPEAT
Employed Bees Phase
Onlooker Bees Phase
Scout Bees Phase

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol4, Issue 4, May 2017

Memorize the best solution achieved so far
UNTIL (Cycle = Maximum Cycle Number or a Maximum CPU time)

1. In the initialization phase, the population of food sites (solutions) is initialized by artificial scout bees and control parameters are set.
2. Search for an executable state and evaluate the test node.
3. Initialize the current path as cycle=1
4. Repeat
5. Produce initial food sites randomly that correspond to solution using:
   Xij= Xjmin + rand (Xjmax-Xjmin)
   //For finding the initial test case
   Xij is the initial test case
   Xjmin is the minimum no. of test cases
   Xjmax is the maximum no. of test cases
   rand () is a random number generation function which selects either 1 or 0 randomly.
6. Test data are generated with the help of white box testing techniques like equivalence partitioning and boundary value analysis.
7. Greedy selection process is applied on generated test data.
8. Pass the generated test data to SUT and calculate the fitness value.
9. Test cases with highest fitness value are selected by onlookers' bee and leave the rest.
10. Same process is repeated till a particular test data with 100% fitness value and 0% fitness value is produced.
11. Pm= fitm (Xm)/Σ fitm (Xm)
12. Where Pm is the probability function which signifies the probability with which the ith test data traverses an independent test path successfully.
13. 11. Store the test case to the optimal database.
14. 12. New Test data generated by scouts bee in next iteration and go to step 5

For implementation of the above algorithm, our approach uses the BCO Test Case Optimization tool to select the best Test Cases with maximum coverage by applying the BCO algorithm. The tool considers a program as an input to generate independent paths. Using the generated independent paths Test Cases are traversed along the paths with the help of ABC algorithm. On executing these test cases with maximum coverage (High fitness Value) are selected. Finally the optimal Test cases are generated.

## VI. PROBLEM SOLVED USING BCO BASED ON MAXIMUM APPROACH

Test Case Selection based on white box testing helps to prioritize the test cases based on the internal structure of the program. Therefore knowledge of internal structure of code used to find the test cases required to provide maximum test coverage. [4] Path testing is a type of white box testing that selects the set of test paths exist in the program. In path testing it picks number of paths to assure that every program statement is executed at least once.

## VII. CASE STUDY

For illustration purpose we have taken the triangle problem to classify a triangle. Inputs are given to three sides of triangle and out will be one of these: scalene, isosceles, equilateral and not a triangle. On executing the program code provided as input to the tool. The forager bees i.e. onlooker bees will generate different numbers of paths. In the program there are 18 nodes and number of independent path generated are:

1 ABFGNPQR
2. ABFGNOQR
3. ABCEGNPQR
4. ABCEQNOQR
5. ABFGHJIMQR
6. ABFGHJKMQR
7. ABFGHIMQR

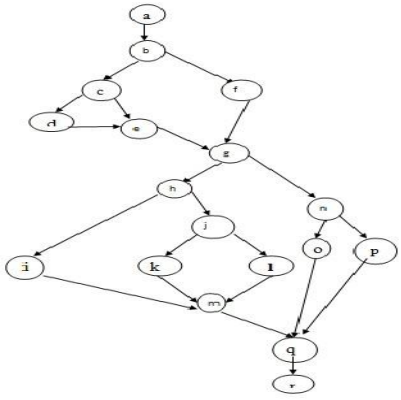


***Fig 7: Flow Graph of the triangle problem code***

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol4, Issue 4, May 2017

***According to proposed BCO algorithm, the steps are:***

1. All the paths are initializes as test cases.
2. Scout bees Initialize the current traversal path as Cycle=1
3. Repeat
4. Generate new test cases Xij using formula for movement of scout bees:
   Xij= Xjmin + rand (Xjmax-Xjmin)
   Xij= initial test case
   Xjmin (minimum numbers of test cases) =1
   Xjmax (maximum numbers of test cases) =7
   Rand=A random Number (0, 1)
   So, Xij= 1 + 0 * 7-1 = 1 (Path number)
5. ABFGNPQR is selected by employee bee as it traverses.
6. Generate the test data using functional testing techniques.
7. Calculate the fitness value.
8. Onlooker bees selects the test cases which having highest fitness value and leave the rest.
9. In the next iteration cycle scouts bee generate the new test data and repeat step5.
10. Selected test cases or data are stored in database.
11. Go to Step 1 and Repeat the same the process for other test paths.

| S.No. | Test Case | Output | Path Covered | Fitness value % |
|---|---|---|---|---|
| 1 | 50-50-1 | Isosceles | ABCDEGHJKMQR | 100 |
| 1 | 50-50-2 | Isosceles | ABCDEGHJKMQR | 100 |
| 1 | 50-50-99 | Isosceles | ABCDEGHJKMQR | 0 |
| 2 | 50-50-50 | Equilateral | ABCDEGHIMQR | 100 |
| 2 | 80-50-80 | Isosceles | ABCDEGHIMQR | 0 |
| 3 | 50-1-50 | Isosceles | ABCDEGHJKMQR | 100 |
| 3 | 50-2-50 | Isosceles | ABCDEGHJKMQR | 0 |
| 3 | 1-50-50 | Isosceles | ABCDEGHJKMQR | 0 |
| 4 | 50-50-100 | Not a triangle | ABCDEGNOQR | 0 |
| 4 | 100-50-50 | Not a trianlge | ABCDEGNOQR | 100 |

***Table: Test cases with input, output path covered and fitness value***

If the test case data covered all the nodes of path so the fitness value with respect to path is 100% where if they don't covered test node completely then for the path ABFGNPQR, fitness value are 0%.

## VII. APPLICATION OF THE PROPOSED APPROACH

Software practitioners may use the algorithm developed to reduce the time & effort required for selection of test cases. This algorithm may lead to greater time & cost savings when applied to larger & complex test suites, as compared to the smaller ones. Using BCO approach, software practitioners can effectively select & prioritize test cases from a test suite, with minimum execution time. Hence, the proposed algorithm may prove to be useful in real-life situations.

## VIII. CONCLUSION

Using the swarm based intelligence algorithm there exist a probability of falling into a Local optimum but using Bee colony optimization algorithm probability is low, As BCO work as combination of both local and global search ability .Bee inspired algorithms have a very promising potential for modeling and solving complex optimization problems. The proposed bee algorithm has been explained using example on maximum coverage or maximum path coverage and is found very effective in solving small to medium sized generalized assignment problems. We investigated the performance of standard of the Bee Colony algorithm and compared their performances against other swarm based intelligence algorithm. Also the results' shows that BCO approach is much better than the other optimizing algorithm and is more efficient and work faster for optimizing test cases.

## IX. SCOPE FOR FUTURE RESEARCH

The scope in the field of nature inspired algorithm is very vast. This field is largely unexplored and therefore no limit on development. As a future work, different versions of BCO have to be applied for minimizing the cost of regression testing and analytical study can be conducted in finding the best BCO version to achieve near global optimal solution, their different other factors can be added to increase the efficiency of the BCO approach and give more prioritized and optimized test case results. This research work may be extended for further research in the following dimensions/environments: which concentrates on optimization based on test adequacy criteria such as coverage. In the future research work, this can be done based on other test adequacy criteria such as data based and flow based measures.

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol4, Issue 4, May 2017

## REFERENCES


1. Back, T. 1996: Evolutionary algorithms in theory and practice, Oxford University Press.
2. Automated Generation of Independent Paths and Test Suite optimization Using Artificial Bee Colony (IJSRD/Vol. 3/Issue 03/2015/452)
3. An Exhaustive Survey on Nature Inspired Optimization Algorithms. International journal of software engineering and its application.
4. Swarm Intelligence: focus on ant Book edited by flex T.S.Chan and Manoj kumar Tiwari. ISBN978-3-902613-09-7, pp. 532, December 2007, Itech Education and Publishing, Vienna, Austria .
5. J.H. Holland, Genetic algorithms and the optimal allocation of trials, SIAM J. Comput. 2 (2) (1973) 88–105.
6. Koza, John R. 1992. Genetic Programming: On the Programming of Computers by Means of Natural Selection. Cambridge, MA: The MIT Press.
7. http://en.wikibooks.org/wiki/Introduction_to_Software_Engineering/Process/Life_Cycle#cite_note-1.
8. http://en.wikipedia.org/wiki/Software_testing
9. Srivastava P.R., Baby K., Raghurama G., "An Approach of Optimal Path Generation using Ant Colony Optimization",TENCON 2009 - 2009 *IEEE Region 10 Conference*, Jan, 2009, pp. 1-6, 23-26. Karaboga D., "An Idea Based on Honey Bee Swarm for Numerical Optimization", Technical Report-TR06, Computer Engineering Department, Erciyes University, October, 2005.
10. Adil B., Lale O., Tapkan P. "Artificial Bee Colony Algorithm and Its Application to Generalized Assignment Problem",Swarm Intelligence: Focus on *Ant and Particle Swarm Optimization*, December, 2007, pp. 532-564.
11. http://www.jofcis.com/publishedpapers/2011_7_9_3309_3316.pdf
12. Chong, C. S. et. al. (2006): A Bee Colony Optimization Algorithm to Job Shop Scheduling, Journal Winter Simulation Conference, Monterey, CA, pp. 1954-1961.
13. McCaffrey, J. D. (2009): Generation of pair wise test sets using a simulated Bee Colony Algorithm, 10th IEEE International Conference, IEEE Press Piscataway, NJ, USA, pp. 115-119.
14. Navrat, P.; Jelinek, T.; Jastrzembska, L. (2009): Bee hive at work: A problem solving, optimizing mechanism, In Proceedings of Nature & Biologically Inspired Computing, IEEE Conferences, Coimbatore, pp. 122-127.
15. Rahmatizadeh, Sh.; Shah-Hosseini, H.; Torkaman, H.(2009): The ant-bee routing algorithm: A new Agent based Nature-Inspired Routing Algorithm, Journal of Applied Sciences, Addison Wesley Publishers, Harlow. Esses. UK, pp. 983-987.
16. Rothermel, G. Untch, R.H.; Chu, C.; Harrold, M.J. (1999): Test Case Prioritization: An Empirical Study, In Proceedings of the International Conference on Software Maintenance, Oxford, UK, pp. 179-188.
17. Yang, X.S. (2005): Engineering Optimizations via Nature-Inspired Virtual Bee Algorithms, Artificial Intelligence and Knowledge Engineering Applications: A Bio Inspired Approach, Lecture notes in computer science, Springer Berlin / Heidelberg, pp. 317-323.
18. Srinivasan Desikan, Gopalaswamy Ramesh (2006): A book on Software Testing principles and practices published by pear son education 2006.
19. Aditya P.Mathur, A book on foundations of software testing published by pear son education ISBN 978-81- 317-9476-0.
20. Sonam Kamboj1, Mohinder Singh "Research Paper on Improving Quality Using Testing Strategies" Journal of Global Research in Computer Science Vol. 2, No. 6, June 2011.
21. Ruchika Malhotra, Arvinder Kaur and Yogesh Singh "A Regression Test Selection and Prioritization Technique" Journal of Information Processing Systems, Vol.6, No.2, June 2010.
22. T. Prem and T. Ravi "Optimization of Test Cases by Prioritization" IAES International Journal of Artificial Intelligence (IJ-AI), Vol. 1, No. 3, pp. 112~120, 2012.
23. Kevilienuo Kire, Neha Malhotra "Software Testing using Intelligent Technique"International Journal of Computer Applications (0975 – 8887) Vol. 90, No.19, March 2014.
24. Vimal Nayak, Haresh Suthar, Jagrut Gadit "Implementation of Artificial Bee Colony Algorithm"Journal homepage.
25. D. J. Mala, V. Mohan, ABC Tester – Artificial Bee Colony Based Software Test Suite Optimization Approach, International Journal of Software Engineering, Sprinter Global Publication, pp. 15-43, 2009.
26. AdiSrikanth, Nandakishore J Test Case Optimization Using Artificial Bee Colony

Algorithm, A.Abraham et al. (Eds.): ACC 2011, Part III, CCIS 192,pp. 570–579, 2011.
27. International Conference on Communication Technology and System Design, 2011.
28. "Automated Generation of Independent Paths and Test Suite Optimization Using ArtificialBee
29. Colony" Soma Sekhara Babu Lama, M L Hari Prasad Rajub, Uday Kiran Mb, Swaraj Chb, 2011.
30. Mustafa Servet Kiran, Ahmet Babalik "Improved Artificial Bee Colony Algorithm for Continuous Optimization Problems" Journal of Computer and Communications, 2, 108-116, 2014.
31. Arvinder Kaur, Shivangi Goyal "Implementation and Analysis of the Bee Colony Optimization algorithm for Fault based Regression Test Suite Prioritization" International Journal of Computer Applications (0975 – 8887) Volume 41, No.14,March 2012.
32. Dr. Arvinder Kaur "A Bee Colony Optimization Algorithm for Code Coverage Test Suite.