Proceeding of IC-ITS 2015 e-ISBN:978-967-0850-07-8
International Conference on Information Technology & Society
8-9 June 2015, Kuala Lumpur, MALAYSIA

# Analysis on Generating Test Cases for Random Testing Using Optimization Techniques

K. Devika Rani Dhivya[a], V.S Meenakshi[b], B. Priyadarshini[c]

Sri Krishna Arts and Science College,
Coimbatore, Tamil Nadu. India.

devika58@gmail.com[a], meenasri70@yahoo.com[b], priyadarshinib@skasc.ac.in[c]

**Abstract.**An effective method for testing units of software is random testing. Its thoroughness is according to the settings of optimal parameters. For the purpose of checking the test input data the randomized testing uses randomization. This paper gives a comparison between optimization algorithms like Genetic Algorithm, Particle Swarm Optimization algorithm (PSO) and Artificial Bee Colony (ABC) which is used to produce an optimal test data for randomization within less time. It is used to evaluate the target method solutions for test coverage in test data. The main goal of these algorithms is to generate the optimal test parameter, reduce the size of test case and to achieve high coverage of the testing units with of optimal value.  One of thealgorithms achieves high coverage and produces the better optimal value within less time.

**Keywords:** Randomized unit testing, Particle Swarm Optimization algorithm (PSO) and Artificial Bee Colony(ABC).

## INTRODUCTION

Software testing is the process of examining a software product to find errors. The goal of testing is to evaluating the quality of an application and to improve it a primary purpose of testing is to detect software failures so that defects may be discovered and corrected. Unit testing is the process of checking the correctness for each unit.

Randomized unit testing is a process of testing the input data for some aspect. The thoroughness of randomized unit testing is dependent on when and how randomization is applied; e.g. the number of method calls to make, the relative frequency with which different methods are called, and the ranges from which numeric arguments are chosen. Randomization for some aspect of test input is used by randomized testing process. Randomized testing of a unit is very effective at forcing failures in even a well-tested unit. A test case is generated by the value which is given as an input for an application. Every input data of an application is a test data.The action and the accuracy performance of randomized unit testing are depended up on applying randomization, for e.g. the number of method calls are made, the relative frequency with which different methods are called and the ranges from which numeric arguments are chosen. Randomization for some aspect of test input is used by randomized testing process. Randomized testing of a unit is very effective at forcing failures in even a well-tested unit. Software testing is any activity aimed at evaluating an attribute or capability of a program or system and determining that it meets its required results.

### *A. Randomized Unit Testing*

Random testing can quickly generate many tests, is easy to implement, scales to large software applications, and reveals software errors. Randomized testing uses randomization for some aspects of test input data selection. The key benefit of randomized testing is the ability to generate many distinct test inputs in a short time, including test inputs that may not be selected by test engineers but which may nevertheless force failures.

Randomized unit testing depends on the generation of so many inputs that it is infeasible to get a human to check all test outputs. Randomized unit testing is a promising technology that has been shown to be effective, but whose thoroughness depends on the settings of test algorithm parameters. To solve the issues of the randomized unit testing thus generate the test data and test case generator by use the following algorithm to more coverage of the test cases. The algorithms are GA, PSO and ABC. These optimization methods are used to produce an optimization test data and coverage of the test case at highest level.

### *B. Test Case And Test Data Generation*

A test case is a set of condition or variables under which software tester will determine whether the application or a system is working correctly. The value which is given as an input for an application which is under test is said to be a test data [7].

Every test data is assumed under some conditions for the correctness of the application, thus the test case is generated. The algorithm takes two parameters to construct the test data and test case generation: a set M of java target methods and a genetic algorithm chromosome c appropriate to M. The method call is used to constructed and run test cases under randomized unit testing in genetic algorithm. A test case is defined as the sequence of call descriptions, the test case generation takes the possible data as a solution space to search, and apply to GA approach to find for a good solution. The test case generation collects the number of lines covered by the test case. Each gene is a pair consisting of a name and an integer, floating point, or BitSet value.

### *C. Genetic Algorithm*

Genetic algorithms (GAs) were first described by Holland. Candidate solutions are represented as “chromosomes”, with solution represented as “genes” in the chromosomes. The possible chromosomes form a search space and are associated with a fitness function representing the value of solutions encoded in the chromosome [1]. Search proceeds by evaluating the fitness of each of a population of chromosomes, and then performing point mutations and recombination on the successful chromosomes. The GA represents the design variables of each individual design with binary strings of 0’s and 1’s that are referred to as chromosomes. The GA begins its search from a randomly generated population of designs that evolve over successive generations (iterations). The first operator is the “Selection” operator that mimics the principal of “Survival of the Fittest”. The second operator is the “Crossover” operator propagates features of good surviving designs from the current population into the future population, which will have better fitness value on average. The last operator is “Mutation”, which promotes diversity in population characteristics. By using the above process GA retrieves a better optimal solution for a test case.

### *D. Particle Swarm Optimization (PSO)*

The concept of PSO was first introduced by Kennedy and Eberhart in 1995. Particle swarm optimization (PSO) is inspired by the social behavior observed in flocks of birds and schools of fish. In nature, there is a leader who leads the bird or fish group to move.In PSO, a potential solution to the considered problem is represented by a particle, similar to the individuals in the bird and fish group.PSO is a population-based stochastic optimization paradigm, in which each agent, named particle, of the population, named swarm, is thought of as a collision-proof bird and used to represent a potential solution. PSO is similar in some ways to other existing GAs, such as GA, but is defined in a social context as opposed to a biological context. Compared to GA PSO is generally characterized as simple in concept, easy to implement and computationally efficient for coverage of the tests cases and reduces the computational cost of the GA [4].

### *E. Artificial Bee Colony (ABC) algorithm*

In ABC algorithm, the colony of artificial bees contains three groups of bees: employed bees, onlookers and scouts.First half of the colony consists of the employed artificial bees and the second half includes the onlookers.

For every food source, there is only one employed bee. In other words, the number of employed bees is equal to the number of food sources. The employed bee of an abandoned food source becomes a scout. The search carried out by the artificial bees can be summarized as follows:

- Employed bees settle on a food source within the neighborhood of the food source in their memory. Employed bees share their information with onlookers within the hive and then the onlookers select one of the food sources. Onlookers select a food source within the neighborhood of the food sources chosen by them. An employed bee of which the source has been abandoned becomes a scout and starts to search a new food source randomly.

### **D.** *Objective and Scope of the Research*

- The objective is to minimize the time consumption.
- To improve an optimization process to produce optimal test data value.
- To reduce the size of test case generation and to achieve high test coverage.
- The scope of using the ABC algorithm, it improves the efficiency of the system and reduces the time to test coverage by initializing particles and also by parallel execution and Feature subset selection tool.

## GENETIC ALGORITHM FOR RANDOM TESTING

Randomized unit testing is unit testing where there is some randomization in the selection of the target method call on the successful chromosomes. A key concept in randomized unit testing is that of value reuse. Genetic algorithm approach is used to search and find for a good solution to search space. Genetic algorithm is used to making decisions about the features are worthy of modeling and mutating. It can defeat purely random search in finding solutions to complex problems. The Genetic algorithm performs the usual candidate solution evaluation steps (fitness selection, mutation, and recombination). Genetic algorithm uses Feature subset selection (FSS) tool to assess the size and content of the representations within the in GA.

The existing system of genetic algorithm is to find parameters for randomized unit testing. The systems optimize the test coverage under randomized unit testing. The tool can be used to reduce the size of the representation and achieving most of the coverage found using the full representation. Feature subset selection tool is used prune the needless information in the gene types and achieving high level of the nearly same coverage.

Improved GA with FSS for randomized unit testing are used but the optimization of testing process is not optimal and also less minimization of test case generation, system does not consider the fitness value for randomized unit testing it generates the random selection of software units.

Due to the iteration process comes too complex and time consuming. It does not cover of all test cases in the system and values of the parameters are not optimal. It does not minimize the test case generation and it will not quickly generate the test case.

## PSO ALGORITHM FOR RANDOM TESTING

Particle swarm optimization (PSO) is like the behavior observed in flocks of birds and schools of fish. In nature, there is a leader who leads the bird or fish group to move, most members of the group follow the leader. In PSO, a potential solution to the considered problem is represented by a particle, similar to the individual in the bird and fish group. PSO algorithm is used to evaluate the target method solutions and operates the fitness value to the receiver candidate. It is a technique used to explore the search space between the set of target methods and to maximize a set of target method parameters. It uses a number of agents (particles) that constitute a swarm moving around in the search space to target method .

Each target method defines some function to the randomized unit testing. Each particle represents a possible solution to the optimization value. In every iteration, each particle calculates the fitness value. Every particle has a best value which is calculated by fitness function which is called $P_{best}$ value. According to the $P_{best}$ value the highest priority value is called as $G_{best}$ value which is an optimal solution [3].

One particle determined by its own best solution found and the global best position discovered by any of the particles in the swarm. This means that if a particle discovers a promising new solution, all the other particles will move closer to it, thus the region is nearly to the optimal value in the search area. The PSO algorithm is used to refer the calling method's object in target method objective function and arguments of the target method. The process of PSO is to produce an optimal parameter. The result of this approach is the test coverage time is less when compared with the GA with FSS.

## ARTIFICIAL BEE COLONY FOR RANDOM TESTING

Artificial bee colony (ABC) algorithm is an optimization algorithm based on a particular intelligent behavior of honeybee swarms. This model was based on inspecting the behaviors of real bees on finding nectar amounts and sharing the information of food sources to the other bees in the hive [5]. These specialized bees try to maximize the nectar amount stored in the hive by performing efficient division of labor and self-organization.

The three agents in Artificial Bee Colony are:

- **Employed Bee**

Employed bees visited the food basis and collect information about food source site and the quality. They have memory about the places they have visited before and the quality of food there. It performs the local search and tries to make use of the neighboring locations of the food basis and look for the best places of foods in the nearby areas of the present value.

- **Onlooker Bee**

Onlooker bees are bees that are waiting on the dance area to decide which food basis is better. This choice is made on the basis of information provided by employed bees. Onlooker bees carry out the global search for discover the global optimum.

- **Scout Bee**

Scout bees do a random search for the food. It discover the new region which are uncovered by the employed bees, these bees are totally random in nature and their action of search. It avoids the search procedure to get intent in local minima.

In ABC algorithm each food basis location represents a candidate solution of optimization problem. In optimization problem each solution is associated with the fitness value on the basis of fitness value it is decided that which solution is better. So the nectar amount of a food source corresponds to the fitness value of the associated solution in ABC algorithm. The number of employed bees or the onlooker bees is equal to the number of solutions in the population.

The ABC algorithm generates a random solution or initial population of size NF, where NF denotes the size of population or total number of food source.

Each solution is represents the position of food source and denoted as xij, where i represents a particular solution (i=1,2,…..,NF) and each solution is a D-dimensional vector so j represents a particular dimension of a particular solution (j=1,2,….,D). After initialization of random solution employed bees start their searching. Employed bees search the food source near the previous food basis, if the generated new solution is better than the previous solution than new solution replaces the old one.

The comparison of food source is done on the basis of fitness value or nectar amount of food source. After all employed bees complete the search processes, they share the nectar information of food sources (solutions) and their position information with onlooker bees. A food source depending on the probability value Pi associated with the food source is chooses by onlooker bee. Probability value for each food source is calculated by following equation (1):

$$P_i = \frac{f_i}{\sum_{n=1}^{NF} f_n} \tag{1}$$

Where $fi$ is the fitness value of the solution i or the nectar amount of food source evaluated by employed bee and NF is the number of food source. So after the evaluation of the food source by the employed bees the probability value for each food source is determined which is used by onlooker bees.

To produce the candidate solution from the previous solution artificial bee uses the following equation (2):

$$\mathbf{Vij=Xij+Øij(Xij-Xkj)} \tag{2}$$

Where j is a index for dimension (j=1,2,….,D), k is a index which represents particular individual or solution from the population (k=1,2,3…..,NF), and i is also a index represents a particular solution (i=1,2,…..,NF). The difference between i and k is that k is determined randomly and value of k has to be different from i. Øij is a random number between [-1,1]. It controls the production of the neighbor food positions around Xij. The difference between the parameters of the Xij and Xkj decreases, the perturbation on the position Xij decreases, too. Thus, as the search approaches to the optimum solution in the search space, the step length is reduced. After the production of candidate solution Vij , its fitness value is calculated and then it is compared with the fitness of xij. If the new candidate solution has equal or better nectar or fitness than the old, it is replaced with the old one in the memory. Otherwise, the old is retained. If a solution is not improved further through a predetermined number of cycles then that food source is assumed to be exhausted. Exhausted food source is replaced by new food source generated by scout bees.

## RESULT AND DISCUSSION

This paper gives a comparison of optimization algorithms like GA, PSO and ABC for Randomized Testing. The representativeness of the units under test is the major threat to external validity. The process of assigning optimization search algorithm for a particular test case of an application shows the effective test coverage. While comparing with the GA and PSO, ABC algorithm gives a best result. The ABC reduces the time and yields a better accuracy of the unit under test. The lines of code covered by the 9 test cases run, as measured by the number in code covered is the ratio of lines covered.

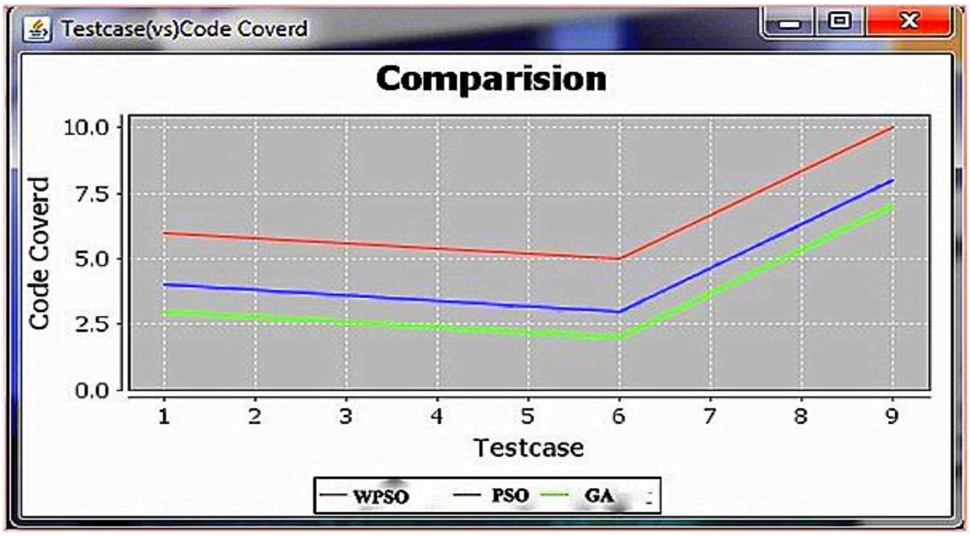


**Figure 1.** Comparison of the Test case Vs Code Coverage

The above figure shows the coverage of the test case for all the methods. The performance of test case coverage with the number of code coverage is compared between methods GA, PSO &ABC. The number of the test case is high, when the code coverage or the result of software is high. Thus, the proposed ABC performs high test case coverage.

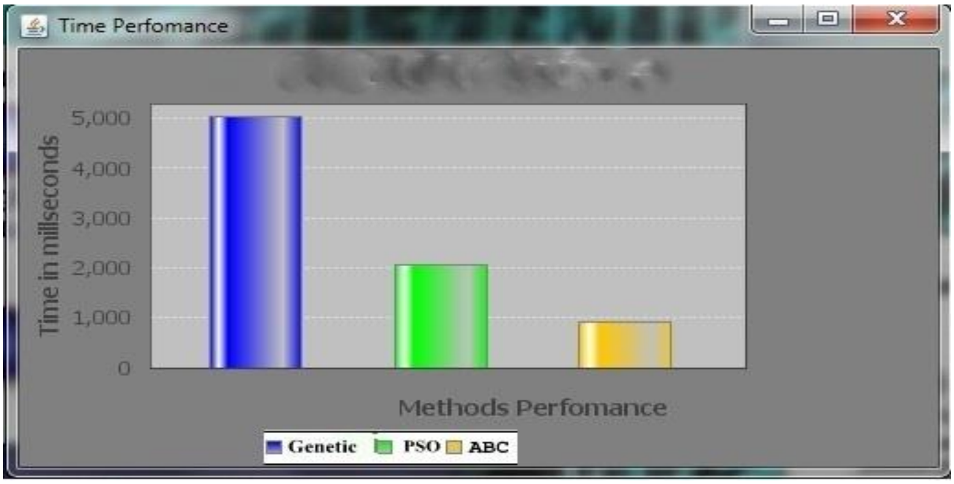


**Figure 2.** Comparison of the Time Vs Methods

The above shows the time coverage for each method. Time is calculated by the time taken for each method to execute the test case of an application. The comparisons of time for every method are given to analysis the time taken to execute the system. Measuring the time performance of the proposed algorithm (PSO) by using the chart, the performance is compared between Genetic Algorithms, PSO & ABC. Here, the performance time of all the methods calculated by milliseconds. Time taken to execute test case of an application by:

**Table 1.** Time Comparison with Methods

| Methodology | Time |
|---|---|
| GA | 5048 milliseconds |
| PSO | 2070 milliseconds |
| ABC | 920 milliseconds |

Thus the executing time taken by ABC is 20% which is very less while comparing with GA and PSO. Fig. shows the performance of the ABC algorithm by using the chart, the performance is compared between GA, PSO & ABC.

## CONCLUSION

ABC algorithm automatically derives good parameter values for randomized unit test algorithm, and it can able to achieve high achieve high coverage and also the ability to generate many new high-coverage test cases quickly and easily. Thus the proposed algorithm, ABC is used to evaluate the target method solutions and operates the fitness value to the receiver candidate. It solves the parameter space of problem between the target method and objective function of the test data. It produces an optimal test value for RT. It achieves high test case coverage within less time and. In future work, Bat-inspired, Ant-colony algorithms and Branch and bound algorithm and other advanced Meta heuristic optimization algorithm also can be used for Randomized Testing to derive an optimal solution. They may reach much better result and high test case cover for an application**.**

## REFERENCES

1. James H. Andrews, Tim Menzies and Felix C. H. Li ,” Genetic Algorithms for Randomized Unit Testing”, Ieee transactions on software engineering, vol. 1, no. 1, January 2011.
2. Jahnavi K and BashaP,”Dynamic Optimization of Genetic Algorithms for Randomized Unit testing by using FSS “, International Conference on Computing and Control Engineering (ICCCE 2012), 12 & 13 April, 2012.
3. SerkanKiranyaz, TurkerInce, AlperYildirim, and MoncefGabbouj , “ Fractional Particle Swarm Optimization in Multidimensional Search Space”, IEEE Trans Syst Man Cybern B Cybern. 2010 Apr;40(2):298-319. Epub Aug 4 2009.
4. Shenga Z, Ying Zhang, H Z and Qingguan H, “Automatic path test data generation based on GA-PSO”,Intelligent Computing and Intelligent Systems (ICIS), 2010 IEEE International Conference on 29-31 Oct. 2010Volume: **1** Page(s): 142 – 146.
5. Karaboga&BahriyeAkay, “A comparative study of Artificial Bee Colony algorithm”, Erciyes University, The Department of Computer Engineering, Melikgazi, 38039 Kayseri, Turkey, Elsevier Available online 11 April 2009.
6. Zhi-huixing, Hai-bin duan, AND chun-fang xu, “A Hybrid Artificial Bee Colony Optimization And Quantum Evolutionary Algorithm For Continuous Optimization Problems”, International Journal of Neural Systems, Vol. 20, No. 1 (2010) 39–50 c World Scientific Publishing Company, February 8, 2010 18:9 S012906571000222X.
7. K. Devika Rani Dhivya, A Review on Optimization Methodologies Used for Randomized Unit Testing, International Journal of Advanced Research in Computer Science and Software Engineering , Volume 4, Issue 6, June 2014 ISSN: 2277 128X.