International Journal of Advance Research in Science and Engineering
Vol. No.6, Issue No. 01 , January 2017
www.ijarse.com
IJARSE
ISSN (O) 2319 - 8354
ISSN (P) 2319 - 8346

A HYBRID MODEL OF PARTICLE SWARM AND ANT COLONY OPTIMIZATION ALGORITHM FOR TEST CASE OPTIMIZATION

Madan Singh[1], Shubhangi Sharma[2]

[1]*Computer Science and Engineering, SRM University (India)*

[2]*Computer Science and Engineering, SRM University (India)*

ABSTRACT

Regression testing is the process of validating modifications introduced in a system during software maintenance. It is done to check that a system update does not introduce errors that have been corrected or the change in one part of the program does not affect the other modules of that program. As the test suite is very large, system retesting consumes large amount of time and computing resources. In that scenario we use test case prioritization. In this approach the testers sort out the test cases such that those with higher priorities are run earlier than those with lower priorities. The purpose of this prioritization is to increase the likelihood that if the test cases are used for regression testing in the given order, they will more closely meet some objectives than they would if they were executed in some other order. In this paper a hybrid approach has been proposed that is divided into four phases. The first being clustering based on requirement prioritization, second being Intra-Cluster Prioritization, next being Test Case Selection and Inter-Cluster Prioritization. In the second phase i.e. Intra-Cluster Prioritization PSACO (PSO+ACO) algorithm has been used. The reason for combining PSO (Particle Swarm Optimization) with ACO (Ant Colony Optimization) is the guaranteed convergence of ACO. Also properties of low constraint on the continuity of objective function and ability of adapting to the dynamic environment make PSO one of the most important swarm intelligence algorithms. Due to these properties the proposed approach combines the strengths of PSO and ACO to give an optimized test suite.

Keywords: Regression Testing, Test suite, Particle Swarm Optimization, Ant Colony Optimization, Convergence

I. INTRODUCTION

Software testing is the design and implementation of a special kind of software system: one that exercises another software system with the intent of finding bugs.

Software Testing is a critical part of the development of any system. It can be carried out at a number of levels and is planned as an integral part of the development process.

IJARSE
ISSN (O) 2319 - 8354
ISSN (P) 2319 - 8346

Regression testing is a type of software testing that verifies that software previously developed and tested still performs correctly after it was changed or interfaced with other software. The changes may include software enhancements, patches, configuration changes, etc. It is the process of executing the set of test cases which have passed on previous builds or release of the application under test in order to validate that the original features and functions are still working as they were previously. Sometimes it is possible that due to budget and time constraints it is not possible to rerun all the test cases in the test suite. In that scenario we use Test Case Prioritization.

1.1 Test Case Prioritization

Test case prioritization schedule test cases in order to increase their ability to meet some performance goal:

- Rate of fault detection
- Rate of code coverage
- Rate of increase of confidence in reliability

Test Case Prioritization techniques, which are used to improve the cost effectiveness of regression testing, order test cases in such a way that that those cases that are expected to outperform others in detecting software faults are run earlier in the testing phase. The purpose of this prioritization is to increase the likelihood that if test cases are executed in the given order they will more closely meet some objectives than they would if they were executed in some other order.

1.2 Ant Colony Optimization

The Ant Colony Optimization is a probabilistic technique for solving computational problems which can be reduced to finding good paths through graphs. Initially proposed by Marco Dorigo in 1992 in his PhD thesis, the first algorithm was aiming to search for an optimal path between the ant colony and food source.

In the natural world, ants wander randomly and upon finding food return to their colony while laying down pheromone trials. If other ants find such a path, they are likely not to keep travelling at random, but instead to follow the trial, returning and reinforcing it if they eventually find food.

Over time, however, the pheromone trial starts to evaporate. The more time it takes for an ant to travel down the path and back again, the more time the pheromones have to evaporate. A short path by comparison, gets marched over more frequently, and thus the pheromone density becomes higher on shorter paths than longer ones. Pheromone evaporation has the advantage of avoiding the convergence to a locally optimal solution. If there were no evaporation at all, the paths chosen by the first ants would tend to be excessively attractive to the following ones. In that case, the exploration of the solution space would be constrained. The overall result is that when one ant finds a good path from the colony to a food source, other ants are more likely to follow that path, and positive feedback eventually leads to all the ants following a single path.

Edge Selection

An ant is a simple computational agent in the ACO algorithm. It iteratively constructs a solution for the problem at hand. The intermediate solutions are referred to as solution states. At each iteration of the algorithm, each ant moves from a state x to state y, corresponding to a more complete intermediate solution. Thus, each ant k computes a set $A_k(x)$ of feasible expansions to its current state in each iteration, and moves to one of these in probability. For ant k, the probability p_{xy}^{k} of moving from state x to state y depends on the combination of two values, viz., the attractiveness η of the move and the trial level T_{xy} of the move. Trials are updated usually when

IJARSE
ISSN (O) 2319 - 8354
ISSN (P) 2319 - 8346

all ants have completed their solution, increasing or decreasing the level of trials corresponding to moves that were part of "good" or "bad" solutions, respectively.

In general, the k^{th} ant moves from state x to state y with probability

$P_{ij} = T_{ij}^{\alpha} . \eta_{ij}^{\beta} / (\sum T_{ij}^{\alpha} . \eta_{ij}^{\beta})$

Where,

T_{ij} is the amount of pheromone deposited on edge ij.

η is the desirability of edge ij.

α is a parameter to control the influence of T_{ij}

β is a parameter to control the influence of η and,

$\eta_{ij} = 1/d_{ij}$

Pheromone Update

When all the ants have completed a solution, the trials are updated as

$$T_{ij} \leftarrow (1 - \rho) T_{ij} + \Delta T_{ij}$$

$$\Delta T_{ij} = 1 / L_k$$

1.3 Particle Swarm Optimization

PSO is a computational method that optimizes a problem by iteratively trying to improve a candidate solution with regard to a given measure of quality. It solves a problem by having a population of candidate solutions, and moving these particles around in the search space over the particle's position and velocity. Each particle's movement is influenced by its local best known position, but is also guided towards the best known positions in the search space, which are updated as better positions.

PSO is originally attributed to Kennedy, Eberhart and Shi and was first intended for simulating social behavior. PSO is a metaheuristic as it makes few or no assumptions about the problem being optimized and can search very large spaces of candidate solutions. Each particle modifies its position according to : its current position, its current velocity, distance between its current position and pBest (particle's best solution) and distance between its current position and gBest(best value obtained in the neighborhood of that particle). Inside the swarm (set of particles) a topology is defined : it is a set of links between particles, saying "who informs whom". When a particle is informed by another one, it means that the particle knows the previous best (position and fitness) of the "informing" particle. The set of particles that informs a particle is called its neighbor. Each particle in search space adjust its "flying" according to its own flying experience as well as the flying experience of other particles.

Position Update

Each particle updates its position as

$x_i^{k+1} = x_i^k + v_i^{k+1}$

Velocity Update

Each particle updates its velocity as follows

$v_i^{k+1} + v_i^k + c_1 r_1^k (pBest_i^k - x_i^k) + c_2 r_2^k (gBest_i^k - x_i^k)$

where,

$-v_{max} < v < v_{max}$

c_1 and c_2 controls the speed and c_1 and $c_2 = 2$

IJARSE
ISSN (O) 2319 - 8354
ISSN (P) 2319 - 8346

r_1 and r_2 are random fiction and are 0 or 1 (randomly)

II. RELATED WORK

Daniel Di Nardo et.al discussed an industrial case study on the Coverage-Based Test Case Prioritization with real regression faults. Because of the time or resource constraints Retest-All' strategy to perform the regression testing is time consuming and not efficient with the growing number of test. Therefore, strategies for regression testing such as test suite minimization, test case selection and test case prioritization will block and decision are better than the other mentioned coverage-based prioritization techniques.

Chengying Mao have reformed the basic Ant Colony Optimization to generate better quality test data for the purpose of earlier fault- revealing .Four rules have been re-defined to achieve the objective namely: the local transfer rule, the global transfer rule and pheromone update rule. The criterion used is the branch coverage. The result of the reformed Ant Colony Optimization outperforms the algorithms namely: Simulated Annealing and genetic algorithm.

Rui Ding proposed an algorithm by applying particle swarm optimization into genetic algorithm, to overcome the limitation of genetic algorithm that is local convergence. They have used classical triangle problem and the result is compared with the original genetic algorithm and it resulted that the proposed algorithm yields better efficiency and better test cases.

Osman Gokalp reformed the original Ant Colony Algorithm by employing crossover mechanism in genetic algorithm. Four ant colonies have been employed rather than one colony and it also employs pheromone tables .Each colony will have their own pheromone tables and the best pheromone tables with best solution is chosen. The reformed algorithm is tested using Traveling Salesman Problem. The result of the reformed algorithm outperforms the original Ant Colony Algorithm.

Wang Jun proposed test case prioritization by using genetic algorithm, which is regression-based testing, such that faults can be revealed more quickly, increase the software reliability and also save computational time and resources.

K. Karnavel used the Bee Colony Optimization (BCO) algorithm and performed regression testing and prioritization. The fault coverage is incorporated with the BCO algorithm for fault revealing within less execution time. Using this approach to test the software application, it is found that the testing cost and the execution time have been reduced.

GenianaIoana Laiu used the evolutionary algorithms to generate path test data .The evolutionary algorithms are namely Genetic Algorithm (GA), Simulated Annealing (SA) and Particle Swarm Optimization (PSO).The test data generated from those algorithms are compared to find out which algorithms produces better quality of test data. The result shows that Simulated Annealing produces better quality of test data than Particle Swarm Optimization and Genetic Algorithm.

Nirmal Kumar Gupta, Mukesh Kumar, proposed a strategy that uses genetic algorithm to reduce the number of unfeasible test cases and establishing genetic algorithm to generate suitable test cases and hence improving the genetic algorithm. The genetic algorithm is used to generate test cases for Object Oriented Software. But practically there is a problem of local convergence in genetic algorithm.

IJARSE
ISSN (O) 2319 - 8354
ISSN (P) 2319 - 8346

Luciano S. de Souza proposed a multi-objective Particle Swarm Optimization (PSO) for selecting the test cases based on functional requirements coverage and execution effort. Binary Multi-Objective PSO (BMOPSO) and Binary Multi-Objective (PSO) algorithms are implemented. The result is found to be better than the random search approach for executing the test cases. They have also cited that future work can be performed based on the same experiment but withhigher number of test suites.

Bharti Suri implemented Ant Colony Optimization Algorithm for Test Case Selection and Prioritization. In their study, have used a tool ACO_TCSP for implementation. The best solution was not found by them for all cases but the results closely related to the optimal solution. There is a reduction in the test suite size which is achieved to be 62.5% in all the 4 test runs. Future work emphasis to apply the tool on more examples to prove the effectiveness of the proposed technique.

Minjie Yi proposed an algorithm for path oriented testing by combining two algorithms namely: Ant Colony System Algorithm and Genetic Algorithm (ACSGA) and the proposed algorithm is used to generate path oriented test data. The problem used is Triangle discrimination problem. The result of the proposed algorithm is then compared with the Genetic Algorithm and it is found that the proposed algorithm is more efficient than genetic algorithm.

Rothermel et al. In 1999 proposed various code coverage based test case prioritization techniques. They proposed statement based, branch based, fault exposing potential (fep) based techniques. **Rothermel and Elbaum** suggested two main strategies for test case prioritization. Both strategies can be applied on any coverage criterion. Total strategy simply arranges the test cases in non-increasing order according to the number of statements covered by them, whereas, additional strategy sorts the test cases in decreasing order of covering the maximum statements not yet covered by previously executed test cases before any other previously unexecuted test cases. The techniques proposed by them include both total as well as additional strategies. They proposed total statement coverage, additional statement coverage, total branch coverage, additional branch coverage, total fep, additional fep based prioritization techniques.

Srivastava et al. proposed a combination of requirement based and risk based prioritization where the task on hand is performed on the basis of identified requirements and risk factors. Idea was to select the test cases that are associated with requirement of very high priority, priority being decided by priority as calculated by combining the preferences of stakeholders and combining the value with the risk computed.

The risk based prioritization technique proposed by **Kavitha et al.** attains the objective by ordering the test cases in a way to achieve maximum rate of fault identification with most severe defects identified at the earliest by test cases

A.Pravin et al. proposed that priority is given to the test cases based on the code coverage and improve the testing process by finding the faults earlier. For each test case a value is generated depending upon certain factors then comparison of test cases is done and priority is assigned to the test cases based on their values. Algorithm was compared with the random prioritization technique on two application projects and describes the effectiveness.

Gurinder Singh et al. have proposed test case selection approach from a large test suite using hybrid technique based on genetic algorithms and Ant colony optimizations. The technique developed using this approach

IJARSE
ISSN (O) 2319 - 8354
ISSN (P) 2319 - 8346

identifies and reduces the test data. The approach provides better results in the initial iteration of the whole process. It provides positive feedback and hence it can lead to better solutions in optimum time.

K.K Aggarwal, Y.Singh, A.Kaur proposed a regression testing that was basically a hybrid approach based on total statement coverage. The main idea behind this paper was to achieve total statement coverage at the earliest by prioritizing test cases in such a manner so as to achieve this aim.

Manika Tyagi and Sona Malhotra proposed an approach to prioritize regression test cases based on three factors which are rate of fault detection, percentage of fault detected and risk detection ability. The proposed approach is compared with different prioritization techniques such as no priority, reverse priority and random prioritization using APFD metric.

Wu et al. proposed improvement of Global Swarm Optimization (GSO) by hybridizing it with ABC and PSO. They used neighborhood solution generation scheme of ABC and accepted new solution only when it is better than previous one to improve GSO performance.

Deneubourg et al. comprehensively examined the pheromone laying and action manners of ants. In an observation known as the "double bridge experiment", the nest of a colony of ants was connected to a food source by two bridges of equal lengths. In such an arrangement, ants start to travel around the environs of the nest and finally reach the food source. Along their path between food source and nest ants deposit pheromone. Initially each ant incidentally chooses any one of the bridge among the two. Nevertheless, due to casual alternation, after some time one of the two bridges has higher focus of pheromone than the other one and therefore, attracts more ants. This brings a further amount of pheromone on that bridge making it more attractive with the result that after some time the whole colony traverse toward the use of the same bridge. This colony level behavior, based on autocatalysis, that is, on the exploitation of positive feedback, can be used by ants to find the shortest path between a food source and their nest.

Himanshi et al. proposed a technique to prioritize the path using ant colony optimization. The proposed approach allows tester to find out the probability for each and priority of the shortest path comes out to be maximum.

III. METHODOLOGY

The proposed approach is a hybrid approach that is divided into four phases. The first phase being clustering based on Statement Coverage. This technique groups the test cases into clusters. Once the test cases are housed in clusters, the next task is Intra-Cluster Prioritization. In this each cluster is selected and a hybrid PSACO (PSO+ACO) algorithm is implemented to find the optimal test cases. Once we have clusters and prioritized tests for each cluster, now the next task is to select and run the highest priority test case from each cluster and then the same process will continue in a round-robin manner for the remaining test cases. We can leave the lowest priority test cases from all the clusters. In this way we are left with prioritized and thus an optimized test suite.

Algorithm for the proposed methodology

```
TCP()
{
   For each test case t ϵ T do
```

IJARSE
ISSN (O) 2319 - 8354
ISSN (P) 2319 - 8346

Clustering();
For each cluster C_i do
Intra_Cluster_Prioritization();
Return the optimal test cases of each cluster;
For each cluster C_i do
Test_Case_Selection();
End For
End For
End For
Return the Optimized Test Suite
}

Clustering()

1. Initialize k ← Number of clusters
2. For each test case t ϵ T do
3. Calculate Statement Coverage as
   Statement Coverage = (Statements Covered by a Test Case / Total number of Statements) * 100
   End For
4. Calculate the mean value of the Statement Coverage
   Mean Value = 1/k $\sum_{i=1}^{k} Statement\ Coverage$
5. Group all the test cases in a single cluster that have statement coverage in proximity to the mean value.
6. Move the test cases that are in proximity with the mean value in set S.
7. Return S.
8. T← T-S
9. For each t ϵ T repeat steps 5 to 8 until all the test cases are clustered or a termination condition is reached.
10. Return Cluster set.

Intra_Cluster_prioritization()

1. Initially select test case randomly by an ant and initialize each ant with a random velocity.
2. Find priority of each next test case chosen by an ant by the following formula
   Path Probability = $T_{ij}^{\alpha} .\eta_{ij}^{\beta} / (\sum T_{ij}^{\alpha} . \eta_{ij}^{\beta})$
3. Calculate the cost for each ant. If the current cost is lower than the best value so far, remember this position (pBest).
4. Choose the ant with the lowest cost of all ants. This is gBest.
5. Use this value of gBest to update the pheromone
   $T_{ij} \leftarrow (1 - \rho)\ T_{ij} + \Delta T_{ij}$
   Where, $\Delta T_{ij} = 1 / L_k$
6. Else choose other test cases randomly and find their priorities.

IJARSE
ISSN (O) 2319 - 8354
ISSN (P) 2319 - 8346

7. Repeat the above steps for all the ants to get the optimal path to their destination.

```
Test_Case_Selection()
    {
     For each Cluster do
            Select the optimal test suite.
     end
}

Inter_Cluster_Prioritization()
 {
  Run the test cases selected by the Test_Case_Selection() module
  }
```

IV. CONCLUSION AND FUTURE WORK

In this paper we have presented PAC based Test Case Prioritization which leads to optimal solution for selecting the test cases. The use of ACO that is a promising technique for solving test case selection and prioritization in the proposed work guarantees convergence and efficient code coverage and fault coverage. In future we wish to apply other swarm intelligence algorithms on the same approach to check the efficiency.

REFERENCES

[1.] A.Pravin, S.Srinivasan (2013): *"An Efficient Algorithm for reducing the test cases which is used for performing regression testing",* 2nd International Conference on Computational Techniques and Artificial Intelligence, Dubai, 194-197.

[2.] Bharti Suri, Shweta Singhal (2012) "*Literature Survey of Ant Colony Optimization in Software Testing"* CSI Sixth International Conference on Software Engineering (CONSEG), Indore, pp.1-7.

[3.] Chengying Mao, YuXinxin, Chen Jifu, Chen Jinfu (2012) *"Generating Test Data for Structural Testing Based on Ant Colony Optimization "*12th International Conference on Quality Software, Xi'an, Shaanxi, pp. 98 – 101.

[4.] Colorni, A. Dorigo, M. Maniezzo, V. Trubian 1994. Ant System for Job-Shop Scheduling. Belgian Journal of Operations Research, Statistics and Computer Science, 34, 1 (1999), 39-53.

[5.] Daniel Di Nardo, N.A. (2013), *"Coverage - Based Test Case Prioritization: An Industrial Case Study",* IEEE Sixth International Conference on Software Testing, Verification and Validation, Leumbourg, 302-311.

[6.] DingRui, Feng Xian bin, Li Shuping, Dong Hongbin (2012*) "Automatic Generation of Software Test Data Based on Hybrid Particle Swarm Genetic Algorithm"*, IEEE Symposium on Electrical & Electronics Engineering (EEESYM), Mudanjiang, Kuala Lumpur, pp. 670 – 673.

IJARSE
ISSN (O) 2319 - 8354
ISSN (P) 2319 - 8346

[7.] Elbaum, S. Malishevisky, Rothermel, 2002. "*Test Case Prioritization: a family of empirical studies",* IEEE Transactions on Software Engineering. 28, 2 (Feb. 2002).

[8.] Gurinder Singh and Dinesh Gupta (2013). : *"An Integrated approach to Test Suite Selection using ACO and Genetic algorithm",* International Journal of Advanced Research in Computer Science & Software Engineering, vol.3, Issue 6, pp. 1770-1778.

[9.] H Kan Stephen (2002).Metrics and Models in Software Quality Engineering, Second Edition, Addison Wesley.

[10.] K.Karnavel (2013), *"Automated Software Testing for Application Maintenance by using Bee Colony Optimization"* Information Communication and Embedded Systems, Chennai, 327-330.

[11.] Kothari, C.R (2004).Research Methodology, Methods and Techniques, 2nd Revised edition, New Age International Publisher, Jaipur, pp.348-350.

[12.] Luciano S. de Souza, Miranda Pericles, B.C. Prudencio Ricardo , A. Barros Flavia de (2011*) "A Multi-Objective Particle Swarm Optimization for Test Case Selection Based on Functional Requirements Coverage and Execution Effort"*, 23rd IEEE International Conference on Tools with Artificial Intelligence, Boca Raton, FL, pp. 245 – 252.

[13.] Minjie Yi (2012*),"The Research of path-oriented test data generation based on a mixed ant colony system algorithm and genetic algorithm"*, International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM), Shangha, pp.1-4.

[14.] Musa John D (2004).Software Reliability Engineering: More Reliable Software Faster and Cheaper,2nd. Edition, JWOMlNGTON, lND1ANA47403.

[15.] Nirmal Kumar Gupta and Mukesh Kumar Rohil (2013) *"Improving GA based Automated Test Data Generation Technique for Object Oriented Software"*, IEEE International Advance Computing Conference (IACC),Ghaziabad, pp.249 - 253.

[16.] Osman Gokalp and Aybars Ugur (2012) *"Improving Performance of ACO Algorithms Using Crossover Mechanism Based on Mean of Pheromone Tables"*. International Symposium on Innovations in Intelligent Systems and Applications (INISTA), Trabzon, pp. 1 – 4.

[17.] P.R. Srivastav 2005. Test Case Prioritization. Journal of Theoretical and Applied Information Technology, 178-181.

[18.] Rothermel, R.H. Untch and M.J. Harrold 1999. Test Case Prioritization: An Empirical Study. Proceeding of the International Conference on Software Maintenance, Oxford, U.K, September (1999).

[19.] Rui Ding, X.F. (2012*), "Automatic Generation of Software Test Data Based on Hybrid Particle Swarm Genetic Algorithm",* International Conference on Internet Computing and Information Services, Hong Kong, 173-175.

[20.] Y.Singh, A.Kaur, Bharti Suri 2010. Test case Prioritization using Ant Colony Optimization. ACM SIGSOFT Software Engineering Notes, 35, 4 (July 2010).

[21.] Y.X. Wang and Y. Wang, *"Cognitive informatics models of the brain",* IEEE Transactions Syst., Man, Cybern C, Appl. Rev., *vol. 36*, no. 2, pp. 203-207, March 2006.