

Hybrid swarm and GA based approach for software test case selection

Palak, Preeti Gulia

Department of Computer Science and Applications, Maharshi Dayanand University, India

Article Info

Article history:

Received Jan 22, 2019

Revised Apr 14, 2019

Accepted Jun 25, 2019

Keywords:

Ant colony optimization

Components

Genetic algorithm

Swarm intelligence

Test case selection

ABSTRACT

Being a crucial step and deciding factor for software reliability, software testing has evolved a long way and always attracted researchers due to various inherent challenges. The quality of a software application depends on the effectiveness of the testing carried out during development and maintenance phase. Testing is a crucial but time consuming activity that influences the overall cost of software development. Thus a minimal but efficient test suite selection is the need of the hour. This paper presents a hybrid technique based on swarm based search technique and GA (Genetic Algorithm) for selection of promising test cases to reduce the overall development cost and time of the application. We took component based software into consideration as they offer some inherent advantages over traditional software development paradigms.

Copyright © 2019 Institute of Advanced Engineering and Science.

All rights reserved.

Corresponding Author:

Palak,

Department of Computer Science and Applications,

Maharshi Dayanand University,

Delhi By Pass, Rohtak, India.

Email: palak.aug6@gmail.com

1. INTRODUCTION

In this era of technology, the hardware and software industries are growing together at very fast pace to meet the growing need of smart devices. The smart gadgets have invaded our lives so badly that we can't predict our future without them. The software embedded with these devices play a crucial role to provide best known user experiences to provide the intended functionality. This scenario raises many challenges in front of the software developers to fulfill the quality needs of the end user. Software testing is a crucial and unavoidable step to achieve the same. The role of test cases in the process of testing is very important to verify the functionality and detect faults. A software failure can claim many lives in case of critical systems. Moreover the development paradigms have evolved a long way from traditional procedural approach to a modular component based approach. Component based software engineering (CBSE) [1] evolved back in late 1980's and growing since then. It works on the principle of reusability and the software is developed in small chunks called components. Each component has some set of functionality and interacts with other components through interfaces. They provide a black box view of the functionality. Commercial off the shelf (COTS) is gaining popularity with time. Considering the impracticality of the exhaustive testing, it becomes the need of the hour to select a promising suite of test data that is capable of providing higher fault coverage.

Ant Colony Optimization [2] and Genetic Algorithm [3] are search based techniques that are inspired from nature and natural phenomenon. Swarm intelligence has provided us inspiration to solve many search based optimization problems. These are meta-heuristic techniques that are problem independent and can work with incomplete knowledge. In contrast to heuristics, meta-heuristics provide randomness during searching and prevent us to get stuck in local optima. ACO has been widely used in solving NP hard optimization problems in reasonable amount of time. It is inspired from the behavior of real ants in their natural habitat searching for food and traversing an intelligent path discovered through group behavior.

One ant follows other ant and communicates through a chemical substance called as pheromone. This substance has two operations: deposition and evaporation. The path with maximum deposition of pheromone is likely to be followed by more ants and finally chosen commute to and fro from nest to food source. This whole process inspired researchers to solve technical problems and to search a promising solution by simulating artificial ants and pheromone level. This is achieved by converting the problem into graphical form.

On the other hand, GA is an evolutionary approach to search for promising set of solutions from a pool of population. It is inspired from Darwin's theory of evolution and natural selection. The possible solutions of the problem are first encoded as chromosomes and initial population is created. The two operators: crossover and mutation are applied to produce new population. A fitness function is chosen to determine the effectiveness of new generation. Generation after generation an effective set of solution gradually evolve through this process satisfying the fitness function and convergence is achieved.

We exploited the advantages of above mentioned techniques to develop a hybrid approach i.e. HACGA (Hybrid Ant Colony – Genetic Algorithm) that is capable of selecting promising test cases to reduce the size of test suite without compromising with the efficiency and test coverage.

2. RELATED WORK

Soft computing based techniques have attracted the researchers over many years due to their potential to deal with uncertainty and incomplete knowledge. The field of software testing over component based system is also been influenced with these search based techniques and has resulted into a vast literature and research work done over years. A few important recent researches over last five years in this field are summarized here. Abhishek Singh *et al.* in [4] presented a modified genetic algorithm based technique for test case generation. They used particle swarm optimization (PSO) for fitness enhancement. Neha *et al.* in [5] applied ACO for reducing cost of regression testing and implemented it in C++. Traditional ACO has scarce initial pheromone, keeping that point in mind Shunkun Yang *et al.* in [6] proposed improved pheromone deposition and updation coefficients and compared the results with random testing and GA based testing. Various soft computing based techniques like neural network, ant system etc. are compared in [7] for software fault prediction. Authors in [8] utilized potential of ACO for reducing test cases for object oriented systems and implemented their proposed approach using MATLAB. Maunika *et al.* in [9] exploited Bee colony optimization for test case selection and to improve path coverage. Authors in [10] used genetic algorithm for regression test suite prioritization and produced mutants for object oriented codes. Wasir Rhmann *et al.* in [11] presented their research in which they applied GA for improving test efficiency in early stages of software development. They tried to improve test coverage of activity diagram created from design specification. Researchers are also attracted towards the adaptive behavior of ACO in which they tried to modify the algorithm based on some parameters to get better results in case of test case selection as done by [12-15]. Similarly many researchers and practitioners are more attracted towards genetic algorithm for software testing and applied the same at various phases of testing as in [16, 17].

A variant of GA is presented in [18] as bacteriologic algorithm (BA) and introduced new memorization operator. To consider the fact that there is always a scope of improvement, researchers went one more step ahead and developed hybrid techniques by combining two or more soft computing based techniques to further enhance the potential to optimize problems. One such research is presented in [19] which applies crossover between ants to reduce the regression testing cost. P. Gulia *et al.* in [20] presented a review of all the soft computing based techniques for testing reusable components and concluded that GA and ACO are the prominent nature inspired techniques that attracted researchers in recent years. Authors in [21] proposed a hybrid approach for test case selection using fuzzy inference system and ACO. Further Bee colony optimization (BCO) has also attracted researchers as in [22] where authors implemented GA based BCO for automation of various testing phases. Palak *et al.* in [23] proposed an ACO based model for testing component based software and their interaction failure. To summarize, a vast literature is available in this field which shows its industrial importance and coverage.

3. PROPOSED MODEL

In this section, a hybrid approach is proposed that combines the benefits of ACO and GA. First of all the system under test (SUT) is converted into its respective component diagram. The main idea is to populate the system with some random ants as done in traditional ACO. Each ant while moving to the neighboring components in the search of food deposits some amount of pheromone on its path. Remaining ants follow the pheromone and choose next hop on probability basis depositing more pheromone

on it. A set of test cases is selected by this process and GA is applied to further refine the test case suite. The proposed approach HACGA that is given in Figure 1 is summarized here:

Input: Fault Matrix, Component Diagram

Step 1: Convert the system into component diagram.

Step 2: Apply ACO over component diagram.

Step 3: Over the result of ACO, further apply GA crossover and mutation operation.

Step 4: Repeat steps 2 and 3 until stopping criteria is met.

Output: Reduced set of test cases.

The stopping criteria in this research is based on two parameters: one by limiting the total execution time of the test cases; other by checking whether all the faults has been covered.

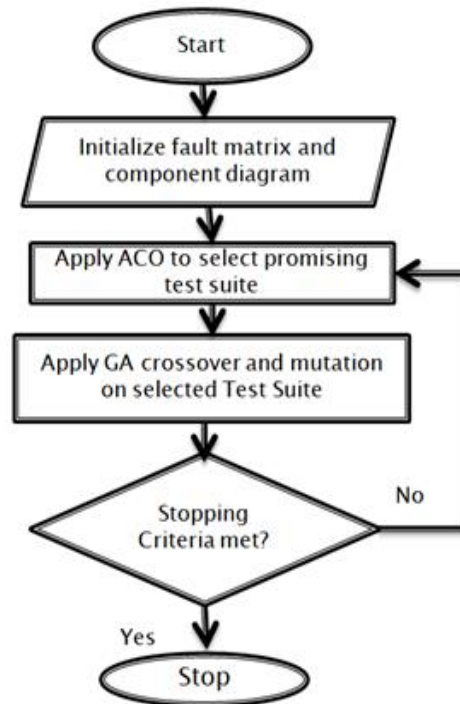


Figure 1. Proposed model

4. METHODOLOGY

Before applying the proposed technique there is a need of a data structure to hold test data showing relationship with the possible faults covered under each test case. A fault matrix $F(m,n)$ is m by n matrix, where m is the number of test cases into consideration and n is the total number of faults. Each row i in this matrix represent a sequence of 0 and 1 indicating the capability of test case T_i to cover a subset of total faults. Various coding schemes can be used for populating fault matrix but we have used 0-1 encoding just for simplicity. The proposed technique is applied on the fault matrix given in Table 1. The entry $F_{i,j}$ in fault matrix at the intersection of row i and column j is determined using the following notation:

$F_{i,j}=1$, if Test case T_i uncovers Fault F_j

$F_{i,j}=0$, Otherwise

We took fifteen different faults and fifteen test cases into consideration with assumption of at least one fault detected per test case. The problem of test case selection can be viewed as selection of subset of rows from fault matrix that covers each column at least once with minimum amount of execution time. First of all traditional ACO and GA is applied on the given fault matrix and results are plotted as shown in Figure 2. Then the proposed HACGA (Hybrid Ant Colony – Genetic Algorithm) approach is applied and it was found that this technique performs better than the traditional techniques and results are shown in next section.

Table 1. Fault matrix

Test Case	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13	F14	F15	Execution Time
T1	1	0	1	0	0	0	0	0	1	0	0	0	0	0	1	6
T2	0	1	0	0	0	0	1	0	1	0	0	1	0	0	0	5
T3	1	0	0	0	0	0	0	1	0	0	0	0	1	0	0	4
T4	0	0	0	0	0	1	0	1	0	0	1	0	0	0	0	5
T5	1	0	0	0	0	1	0	0	0	0	1	0	0	1	0	5
T6	1	0	1	0	0	0	0	0	0	0	0	0	1	0	0	6
T7	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	6
T8	0	0	1	1	0	1	0	0	0	0	1	0	0	0	0	6
T9	1	0	1	0	0	0	0	0	1	0	0	0	0	1	0	5
T10	0	1	0	1	0	1	0	1	0	0	1	0	0	0	0	5
T11	0	0	0	1	1	0	0	0	0	1	0	1	0	0	0	4
T12	0	0	1	0	0	1	0	1	0	0	0	0	1	0	0	5
T13	0	1	0	1	0	1	0	0	0	0	1	0	0	0	0	7
T14	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	6
T15	0	1	0	1	0	0	0	0	0	0	0	0	1	0	0	5

5. RESULTS AND ANALYSIS

Table 2 shows reduced fault matrix $F'_{i,j}$ that contains subset of rows with selected test cases after applying HACGA. The graph shown in Figure 2 compares the proposed technique with traditional techniques. In Figure 2, traditional ACO and traditional GA are compared with proposed approach HACGA in terms of percentage of test cases that need to be executed to achieve higher percentage of faults detection. It was resulted that HACGA based technique is capable of achieving 100% fault coverage in 33% of test cases. While traditional techniques underperform in this scenario. Figure 3 shows comparison of HACGA with the two traditional techniques on the basis on total execution time to detect all the faults and percentage saving in execution time. Graph clearly depicts that the proposed hybrid technique performs better than the traditional technique and resulted into better time saving.

Table 2. Reduced fault matrix with selected test cases

Test Case	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13	F14	F15	Execution Time
T1	1	0	1	0	0	0	0	0	1	0	0	0	0	0	1	6
T2	0	1	0	0	0	0	1	0	1	0	0	1	0	0	0	5
T5	1	0	0	0	0	1	0	0	0	0	1	0	0	1	0	5
T11	0	0	0	1	1	0	0	0	0	1	0	1	0	0	0	4
T12	0	0	1	0	0	1	0	1	0	0	0	0	1	0	0	5

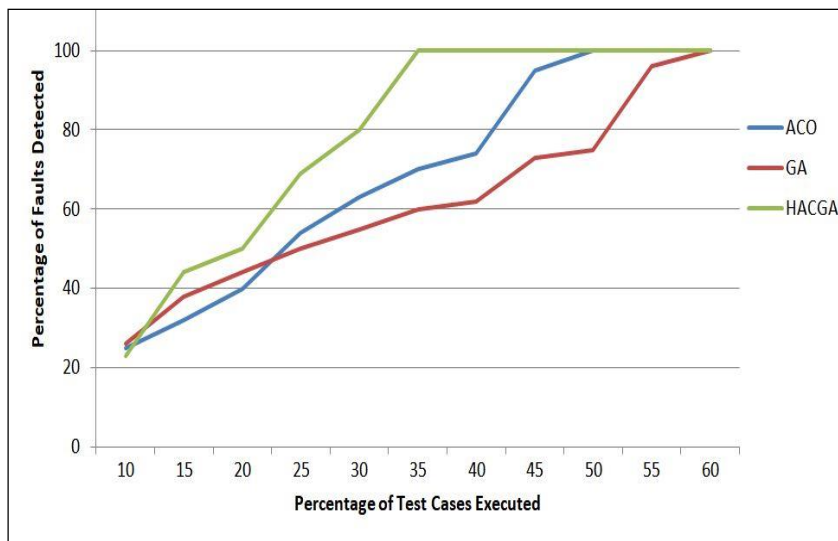


Figure 2. Results

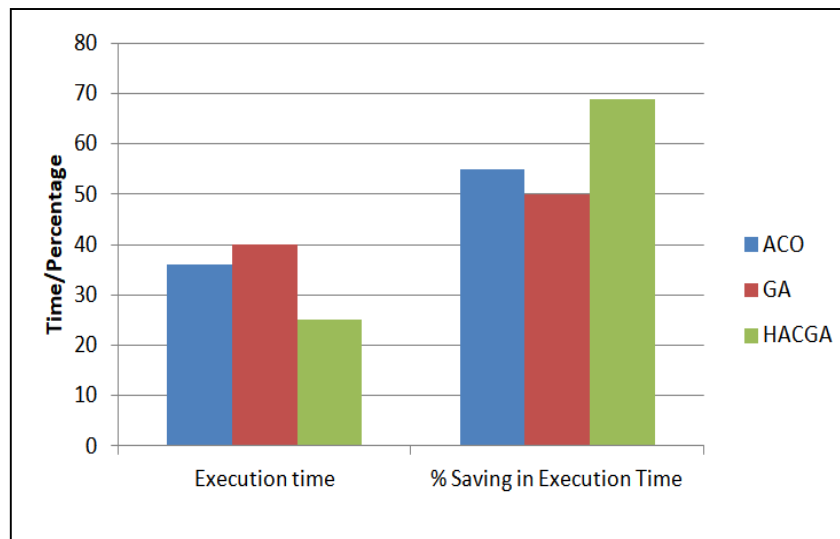


Figure 3. Percentage saving in execution time

6. CONCLUSION

Test case selection is an important activity to reduce the testing effort without compromising with the quality of the software. In recent years, search based nature inspired techniques are evolving to select optimal test suite. This paper presents a Genetic Algorithm based hybrid ACO technique HACGA which exploits the benefits of the two. The main idea is to select a subset of test cases from a large pool which are more promising for efficient testing. These test cases are selected using traditional ACO technique. To further optimize the effectiveness, we employed GA over selected test cases to get a good mix of test cases to avoid the problem of getting stuck in the local optima. It was analyzed that proposed technique performs better than traditional techniques and is capable of finding 100% errors in 33 % of the test cases. Moreover the proposed technique is also capable of reducing the overall test case execution time when compared to other two traditional techniques. In future, it will be interesting to implement the proposed model for large scale testing problems to assess its scalability.

REFERENCES

- [1] Szyperski, Clemens, Dominik Gruntz, and Stephan Murer. Component software: beyond object-oriented programming. Pearson Education, 2002.
- [2] Dorigo, Marco, and Gianni Di Caro. "Ant colony optimization: a new meta-heuristic." Proceedings of the 1999 congress on evolutionary computation-CEC99 (Cat. No. 99TH8406). Vol. 2. IEEE, 1999.
- [3] Mitchell, Melanie. An introduction to genetic algorithms. MIT press, 1998.
- [4] A. Singh, *et al.*, "A hybrid Approach of Genetic Algorithm and Particle Swarm Technique to Software Test Case Generation," *Int. J. Innov. Eng. Technol.*, vol. 3, pp. 208-214, 2014.
- [5] N. Sethi, "Ants Optimization for Minimal Test Case Selection and Prioritization as to Reduce the Cost of Regression Testing," vol. 100, pp. 48-54, 2014.
- [6] S. Yang, *et al.*, "Improved ant algorithms for software testing cases generation," *Sci. World J.*, vol. 2014, 2014.
- [7] E. Erturk and E. Akcapinar, "Expert Systems with Applications A comparison of some soft computing methods for software fault prediction," *Expert Syst. Appl.*, vol. 42, pp. 1872-1879, 2015.
- [8] S. K. Mohapatra and S. Prasad, "Test Case Reduction Using Ant Colony Optimization for Object Oriented Program," *Int. J. Electr. Comput. Eng.*, vol. 5, pp. 1424-1432, 2015.
- [9] M. Mounika and D. V. Reddy, "Test Case Selection For Path Testing Using Bee Colony Optimization," pp. 1-7, 2015.
- [10] S. Musa, *et al.*, "Software Regression Test Case Prioritization for Object-Oriented Programs using Genetic Algorithm with Reduced-Fitness Severity," *Indian J. Sci. Technol.*, vol. 8, 2015.
- [11] W. Rhmann, "Use of Genetic Approach for Test Case Prioritization from UML Activity Diagram," vol. 115, pp. 8-12, 2015.
- [12] C. Ping and X. Min, "Software Testing Case Generation of Ant Colony Optimization Based on Quantum Dynamic Self-Adaptation," vol. 8, pp. 95-104, 2015.
- [13] S. Agarwal, *et al.*, "Automatic Test Data Generation-Achieving Optimality Using Ant-Behaviour," *Int. J. Inf. Educ. Technol.*, vol. 6, pp. 117-121, 2016.

- [14] A. Ansari, *et al.*, "Optimized Regression Test Using Test Case Prioritization," *Procedia Comput. Sci.*, vol. 79, pp. 152-160, 2016.
- [15] M. S. Kumar and P. Srinivas, "An Ant Colony Algorithm to Prioritize the Regression Test Cases of Object-Oriented Programs," *Indian J. Sci. Technol.*, vol. 9, 2016.
- [16] E. Khanna, "Regression Testing based on Genetic Algorithms," *Int. J. Comput. Appl.*, vol. 154, pp. 43-46, 2016.
- [17] G. Kumar and P. K. Bhatia, "Software Test Case Reduction using Genetic Algorithm : A Modified Approach," *Int. J. Innov. Sci. Eng. Technol.*, vol. 3, pp. 349-354, 2016.
- [18] P. R. Srivastava, "Test case optimisation a nature inspired approach using bacteriologic algorithm," *Int. J. Bio-Inspired Comput.*, vol. 8, pp. 122-131, 2016.
- [19] K. Arora, *et al.*, "Hybrid Approach for Optimizing Test Suite Based on GA & ACO," *Int. J. Innov. Res. Technol.*, vol. 3, pp. 65-69, 2016.
- [20] P. Gulia and P. Palak, "Nature Inspired Soft Computing Based Software Testing Techniques for Reusable Software Components," *J. Theor. Appl. Inf. Technol.*, vol. 95, pp. 6996-7004, 2017.
- [21] D. Silva, *et al.*, "A Hybrid Approach for Test Case Prioritization and Selection," *CEC*, pp. 4508-4515, 2016.
- [22] S. Dalal and K. Solanki, "Performance Analysis of BCO-m-GA Technique for Test Case Selection," *Indian J. Sci. Technol.*, vol. 11, 2018.
- [23] P. Palak and P. Gulia, "Ant Colony Optimization Based Test Case Selection for Component Based Software," *Int. J. Eng. Technol.*, vol. 7, pp. 2743-2745, 2018.

BIOGRAPHIES OF AUTHORS



Palak is a research scholar at Department of Computer Science and Applications, MDU, Rohtak, India. She received B.Tech degree from GJUST, Hisar, Haryana, India and M.Tech. degree from MDU, Rohtak, Haryana, India. Her areas of research are software engineering, test case reduction and selection, ant colony optimization and other related fields.



Dr. Preeti Gulia is currently working as assistant professor in the Department of Computer Science and Applications, MDU, Rohtak. She has wide experience in various fields of science and technology such as big data, software engineering, data mining, cloud computing and many more. She has supervised many M.Tech, MCA students and Ph.D. Scholars. She is an active member of ACM, CSI and IAENG.