International Journal of Hybrid Information Technology
Vol.8, No.4 (2015), pp.35-40
http://dx.doi.org/10.14257/ijhit.2015.8.4.05

A Hybrid Meta-heuristics Technique for Finding Optimal Path by Software Test Case Reduction

Deepti Arora[1] and Anurag Singh Baghel[2]

[1] Gautam Buddha University, UP, India
[2] CS Faculty, Gautam Buddha University, UP, India deepti31arora@gmail.com, anuragsbaghel@gmail.com

Abstract

Testing is the one of the crucial phase that is performed during software development. It is done to validate the quality of the software testing using minimal cost, efforts and to generate high quality test cases. It is impossible to test software with all the possible inputs. Hence there is a need to select and reduce test cases that will find as many faults as possible. This paper will describe a method for optimizing software testing Using hybrid meta-heuristics that is by the combination of particle Swarm optimization technique and genetic algorithm, an algorithm has been proposed to find faults in software with reduced number of test cases in minimum time.

Keywords: *Software Testing, Meta-heuristics, Genetic Algorithms, Particle Swarm Optimization, Software test Cases*

1. Introduction

[1] Verification and validation are the most important phases of the SDLC (software development life cycle), Software testing aims to verify that software verifies its specifications mentioned in SRS. Software Testing is done to make sure that product meets the requirement of the product. It is done to check if the behavior of the system is same as that of the desired behavior. Testing does not aim to prove that the product is defect free but to find defects in the software, to provide quality product to customers. It is very time consuming and laborious activity. It is costly and consumes about 50% of the cost of the software development. Test cases are like pesticide in the software which find the pest in form of defects in the software. it is difficult to prove correctness of the program ,software testing is interested in how well test cases test the program code and to find as many faults as possible with minimum number of test cases. A better test case is one which has potential to uncover defects. Regression testing is the most expensive testing technique as in this technique whole set of test cases are executed again and again when modifications are done. It is very time consuming and it is not practically possible to retest system software every time with the same complete set of test cases. [2] Due to this reason researchers have considered various techniques like test case selection, test case prioritization for reducing the cost of regression testing. In test case selection technique subset of test cases are selected that are necessary to validate the software. In test case prioritization, test cases are prioritized that are most essential to run in early phase of testing. Prioritization is done in terms of fault coverage, path coverage, branch coverage. As it is impossible to test the software unit with all the combinations of inputs so there is a need to design minimum no. of test cases that will discover as many faults as possible [3]. Hence there is a need to select and prioritize the test cases. In this paper a hybrid algorithm has been introduced to reduce no. of test cases that is a meta-heuristic algorithm which is a combination of genetic algorithm and particle swarm optimization Technique.

2. Genetic Algorithm

Genetic Algorithm (GA) is a meta-heuristic technique which exploits the search space to find better solution that is optimal solution. GA starts with randomly generated population of individuals. After the initialization of population Fitness of the individual is calculated. Based on the problem fitness function is defined [4]. After the fitness function has been defined individuals are selected based on the Roulette –Wheel Selection method. Second operator is Cross over operator which is done to introduce new off-springs in the population which can have better characteristics than their parents. Mutation is done to introduce new individual which may not have been explored so far .it is done by flipping the bit of the individuals.

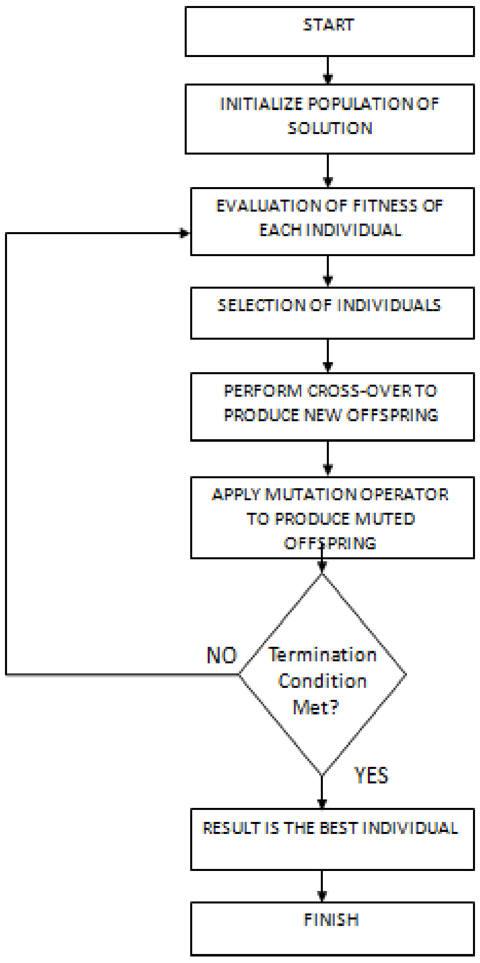
START
INITIALIZE POPULATION OF SOLUTION
EVALUATION OF FITNESS OF EACH INDIVIDUAL
SELECTION OF INDIVIDUALS
PERFORM CROSS-OVER TO PRODUCE NEW OFFSPRING
APPLY MUTATION OPERATOR TO PRODUCE MUTED OFFSPRING
NO
Termination Condition Met?
YES
RESULT IS THE BEST INDIVIDUAL
FINISH

Figure 1. Work Flow GA

3. Particle Swarm Optimization Technique

Particle Swarm Optimization (PSO) [5-7] is another heuristic search algorithm. PSO is generally used to solve those problems whose solution can be represented as a point in an n-dimensional space. In PSO potential solution is called particle. A number of particles are randomly set into motion through this space. Each particle posses its current position, current velocity, and its pbest position .Pbest is the personal best position explored so far. It also incorporates Gbest that global best position achieved by all its individuals. Process repeats until termination criteria are met or the optimal solution is found.

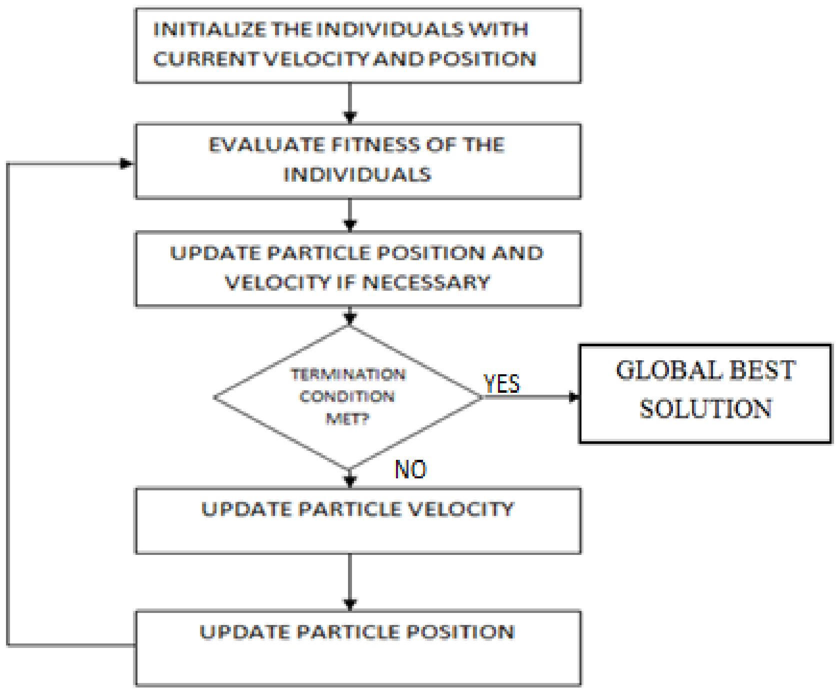
INITIALIZE THE INDIVIDUALS WITH CURRENT VELOCITY AND POSITION
EVALUATE FITNESS OF THE INDIVIDUALS
UPDATE PARTICLE POSITION AND VELOCITY IF NECESSARY
TERMINATION CONDITION MET?
YES
GLOBAL BEST SOLUTION
NO
UPDATE PARTICLE VELOCITY
UPDATE PARTICLE POSITION

Figure 2. Work Flow Of PSO

4. Proposed Algorithm

Hybrid Algorithm has been proposed which is a combination of Genetic and Particle Swarm Optimization Technique [8]. GA suffers from the drawback that it traps in local optima, that are it does not know how to sacrifice short term fitness for long term fitness. It does not keep in account the global best position of the individual. This drawback is overcome by Particle Swarm optimization technique which tracks the particle personal best position as well as global best position, hence it moves toward global optima without getting trapped in local optima. GA has been popular because of its parallel nature of search, and essentially because it can solve nonlinear, multi model problems. But in some cases PSO too suffers from this drawback that during search of best solution, particles can converge to point between particle best and neighbor best in the search space, thus particle converge towards single point, ignoring global best conditions. To overcome this limitation GA mutation function has been used .By combining the power of both the algorithm that is Genetic Algorithm and Particle Swarm Optimization Algorithm a new Hybrid algorithm has been proposed. Algorithm starts with Generation of population of Particles and Test Cases. Test Cases will cover Faults in the program. Randomly generated faults are assigned to test cases .Initialize each Particle for each test case with its Fitness value, Here fitness value is execution time, particle having minimum execution time in comparison to all the particles is considered as gbest position that is global best position. If the particle is having the minimum execution time than previous then this is considered as pbest position that is personal best position of the particle. Sort each particle for each test case depending on their fitness function. Calculate the velocity of each particle .Particle will traverse the test cases Sequence of test cases having minimum execution time will be considered as optimal solution, GA mutation function proposed is used for updating test case sequence and generating new sequences for particle traverse to get best optimal solution. Stopping criteria is total faults covered in minimum time.

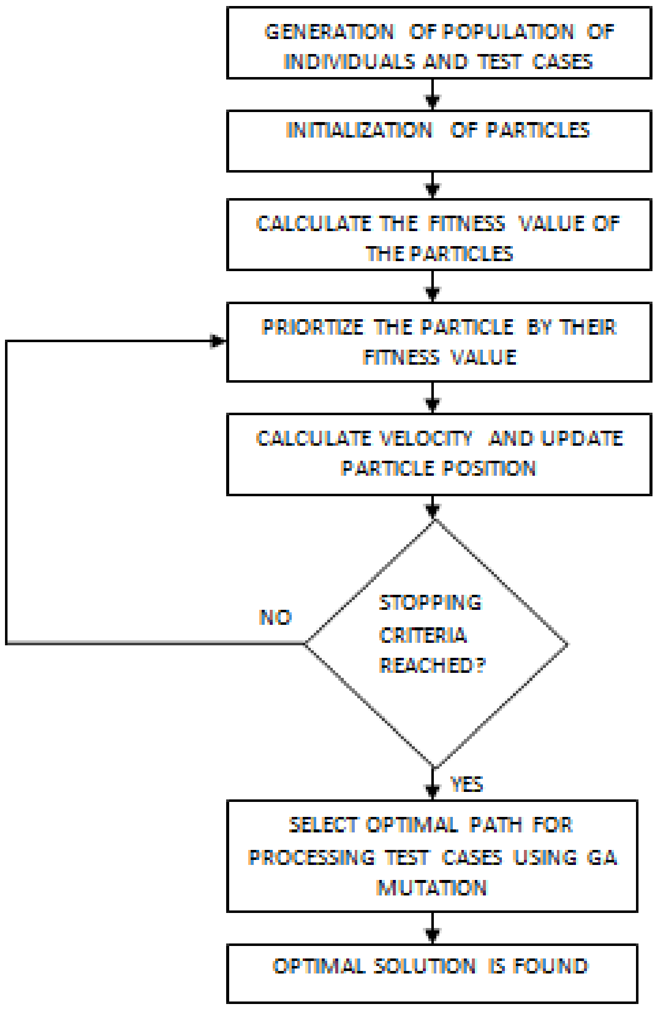
GENERATION OF POPULATION OF INDIVIDUALS AND TEST CASES
INITIALIZATION OF PARTICLES
CALCULATE THE FITNESS VALUE OF THE PARTICLES
PRIORTIZE THE PARTICLE BY THEIR FITNESS VALUE
CALCULATE VELOCITY AND UPDATE PARTICLE POSITION
STOPPING CRITERIA REACHED?
NO
YES
SELECT OPTIMAL PATH FOR PROCESSING TEST CASES USING GA MUTATION
OPTIMAL SOLUTION IS FOUND

Figure 3. Hybrid Algorithm

5. Implementation

Input is no. of test cases which will cover faults, no. of faults that is maximum number of fault to be detected, no of particle which will traverse through sequence of test cases to find faults in the program. Values of the three variables are 6, 7 and 5 respectively. Randomly generate fault for each test case. Randomly generated faults are 53, 12, 40, 114, 85, 47. Faluts are represented in Binary format. No of 1's in a Fault is the fault count.

Table 1

Test Cases	Fault Representation	Fault Count
Tc0	110101	4
Tc1	1100	2
Tc2	101000	2
Tc3	1110010	4
Tc4	1010101	4
Tc5	101111	5

Table 2

Route of the particle	Execution Time
Tc3-Tc1-Tc5-Tc4-Tc2-Tc0	122
Tc5-Tc4-Tc1-Tc0-Tc2-Tc3	99
Tc1-Tc3-Tc4-Tc0-Tc5-Tc2	117
Tc5-Tc1-Tc2-Tc3-Tc0-Tc4	108
Tc1-Tc3-Tc5-Tc2-Tc4-Tc0	126

Table 2 will be obtained after first iteration. Execution time is round off to its nearest integer. Result (Table 3) will be obtained after sixth iteration .That is shortest path will be obtained having minimum execution time.

Table 3

Test Cases	Fault Representation	Fault Count
Tc0	110101	4
Tc1	1100	2
Tc2	101000	2
Tc3	1110010	4
Tc4	1010101	4
Tc5	101111	5

Hence the shortest traversing sequence of test cases is Tc0-Tc1-Tc2-Tc3-Tc4-Tc5. Fault covered is 1111111.Execution time is 80.16.Shortest Route for Processing Test cases that is minimum number of test cases required are Tc0-Tc3-Tc5, Faults covered 1111111, Execution Time is 47.59, that means this is a optimal path having minimum execution time to uncover all the faults.

Graph depicts the number of test cases required to find faults in the program are less in comparison to GA and PSO. Algorithm has been tested on various programs, x axis denotes the serial number of programs.

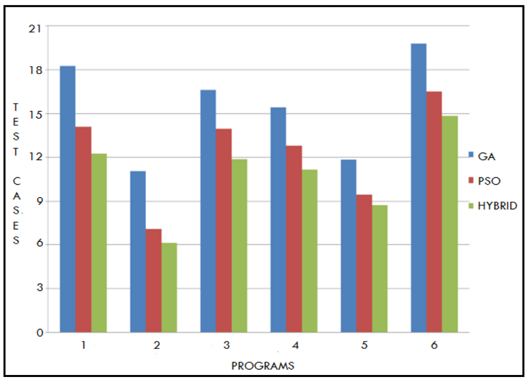

Figure 4. Comparison of Hybrid Algorithm with GA and PSO

parameters and it is easy to implement .Calculation of PSO Algorithm is very simple though it is difficult to implement with problem of non-coordinate system .It is problem dependent, Problem having continuous variables PSO is superior to GA .Many researchers have compared both the technologies. PSO outperforms GA, and is more effective in general however superiority of PSO is problem dependent.

6. Conclusion

A hybrid algorithm PSO-GA has been proposed to reduce the number of test cases and to find all the faults in the program. This problem has been implemented on various problems. One of the problem has been described in this paper. With this algorithm test cases have reduced to half, that is earlier six test cases are required to cover all the faults but now only three test cases have covered all the faults with the minimum execution time. Here is not necessary to traverse all the paths to cover all the faults as we have seen with three test cases also we can cover all the faults. This algorithm can be implemented on various complex problems in future

References

[1] Sudhir, "Analysis of Hybrid Test Case Prioritization Approach Based on BCO and GA", International Journal of Advanced Engineering and Global Technology, vol. 2, no. 10, (**2011)** October.
[2] B. Beizer, "Software testing techniques", Second Edition, Van Nostrand Reinhold, New York, (**1990**).
[3] S. Desikan and G. Ramesh, "Software testing principles & practices", Pearson Education, (**2007**).
[4] K. F. Man, K. S. Tang, and S. Kwong, "Genetic Algorithms: Concepts and Applications", IEEE Transaction on Industrial Electronics, vol. 43, no. 5, (**1996**) October.
[5] J. Kennedy and R. Eberhart, "particle swarm Optimization", Proceedings of the IEEE International Conference on Neural Network, Perth, Australia, (**1995**).
[6] J. Kennedy and R. C. Eberhart, "A discrete binary version of the particle swarm algorithm", in Proc. Conf. on Systems, Man, and Cybernetics, Piscataway, NJ: IEEE Service Center, (**1997**).
[7] R. Poli, J. Kennedy and T. Blackwell, "Particle Swarm Optimization", Swarm Intelligence, Springer, LLC (**2007**).
[8] Y. Liang, L. Liu, D. Wang and R. Wu, "Optimizing Particle Swarm Optimization to Solve Knapsack Problem", ICICA, CICIS, vol. 105, (**2010**), Springer, Berlin.

Authors

Deepti Arora, she is a student in department of information and communication technology pursuing Intrgrated Btech+Mtech (2010-2015) in computer Science, Gautam Buddha University, Greater Noida, Uttar Pradesh, India.

Anurag Singh Baghel, he is working as an Assistant Professor Department of Information and Communication Technology, Gautam Buddha University, Greater Noida,Uttar Pradesh. His research interests include Soft Computing, Image Processing, Embedded Systems Design, Digital Electronics, Computer Architecture and Microprocessor.