TEST CASE REDUCTION USING STATEMENT COVERAGE BASED ANTCOLONY SYSTEM

Dr.C.P.Indumathi[1], Dr.N.Suguna[2], P.Subhavarshini[3], V.Dharani[4]

[1] *Assistant Professor, DCSE, BIT Campus, Anna University, Tirchirappalli, Tamil Nadu, India*
[2] *Associate Professor, Department of CSE, GCE, Sengipatti, Tanjavur, Tamilnadu, India*
[3] *Department of CSE, Jeepiar Engineering College, Chennai, Tamilnadu, India*
[4] *UCE,BIT Campus, Anna University, Tirchirappalli, Tamil Nadu,620024,India*
Email: [1]cpindumathi@aubit.edu.in, [2]sssuguna@yahoo.co.in, [3]subhavarshinimail@gmail.com, [4]dharanivenkatesantech@gmail.com

Abstract: Prioritization of test cases and ordering them for improving the performance of regression testing has become an important researchsubject in recent technology. This paper presents an appropriate method for the Ant Colony System (ACS) to address the problem ofcoverage-based test case priority setting. A tour-based heuristic mechanism and a tree-shaped updation of pheromone rule are used inthis method. The underlying logic is in their later journey, the partially built solutions of the antsare used, and their search is variedsimultaneously. To improve the algorithm's convergence speeda sorting based local search mechanism was used. The propound method is tested and the experimental results have shown that this technique will surpass other state-of-the-artmethods in terms of Average Percentage of Statement Coverage(APSC).Subsequently ,reduced significant number of test cases and help the software testers for immediate testing process.

***Keywords*: Test suite reduction, Test case prioritization, control flow graph, Ant Colony Optimization**

1. Introduction

Regression testing was time-consuming task and it is very costly. This accounts for as much as half of the total cost of maintaining apps. The quantity of regression testing has also increased with the ongoing development of the software systems. Also, the number of test cases is getting much greater. If all test cases were made tore-run,it might can take days , weeks or even months. [1]. Thus researchers have suggested numerous methods to enhance the efficiency of regression testing and make it cost-effective. Such methods are listed in the analysis carried out by Yoo and Harman as test suite minimization, test case selection and test case prioritization [2]. In this project, our aim is to eliminate the redundant test cases in the regression testing suite and thus minimizing the cost of regression testing. It is achieved by solving statement cover-age-based sequence of a control flow graph CFG (Singh, 2012) and prioritize them as per the combined pheromone level and heuristic value on the path. The CFG is a schematic representation of the program's source code. The program statements are represented by the nodes and the flow of control by the edges(Singh, 2012). Throughout our work, we have focused on decision to decision (DD) paths in CFG, a directed graph with program statements as nodes and control flow between those nodes are represented through edges. Bio-inspired algorithms are used to solve complex optimization problems in an optimal manner. The Test Case Prioritization-ANT (TCP-ANT), is used here with ant colony optimization[3] as the basis.Nonetheless ,by using pheromones the current approaches for TCP problems are relied to direct the search while dismissing the heuristic function. We designed the ACO to speed up the search on the basis of a particular problem.Therefore, to boost search performance we can use domain information but,in the existing Ant Colony Optimization(ACO) for the coverage based TCP there is a issues to use domain information.In this paper, to effectively address the TCP-related coverage issues we suggest an expanded Ant Colony Optimization (ACO) architecture, it is capable of using domain-specific expertise and historical search information to direct the finding process. To aid the suggested ACO system we develop an ACB-Heuristic(Additional Coverage-Based Heuristic) method and proved it to be a deciding factor of TCP-related coverage problems to a certain degree and developed a new local search system based on theoverhead described mechanism. The considerable findings are as follows:

i) To help artificial ants to select the states to transmit next according to problem-specific information we propound an ACB-Heuristic approach, i.e. the remaining test cases coversthe number of uncovered statements. This heuristic blends the quest for the efficiency of the state-of-the-art additional greedy approach with the global search ability of the ACO method, such that the theoretical scheme avoids the downside of existing techniques

ii)We are suggesting a sorting-based local search (SB-LS)method to refine the answers of artificial ants. Depending upon a coverage-based TCP problem's solution key property, thissearch method boosts solution consistency and the finding performance. In addition, this finding is universal and to be extended to the consistency of answers of coverage-based TCP problems produced by several other approaches.

ISSN (Print): 2204-0595
ISSN (Online): 2203-1731

iii)We also perform detailed studies to test the conceptual structure for the ACO. In contrast,multiple stateoftheart approaches are introduced. The research uses a variety of comparison problems and functional issues. The findings have shown that the new ACO has high effectiveness and generality.

The remaining portion of this article is structured as follows. Section II discusses the overtures ofthe coverage-based and the ACO approach, and reviews several relevant works. Section III provides the basis for the propound ACO. The experiments are provided for in Section IV. Section V addresses threats to the study's validity. This report is eventually concluded in Section VI.

2 Related work

Throughout the last few years, we have gone across the effects and usefulness of Ant Colony Optimizatiom(ACO) and swarm-based methods to automate the different phases of software testing. N.Sethiet's.Al. in [7] used ACO in regression testing to reduce test suite. S. Yang et al. Al.[8] implemented a updated ACO approach for automatic testing of applications. We introduced new update coefficient for local pheromones and con-trasted the findings with current approaches focused on random and genetic algo-rithms. In their research they report increased performance and test coverage [9]C. Lu et.Al. by using tree-shaped function for pheromone updates for statements cover-age-based test case prioritization.Mukesh Mann et al.[10] al-so ACO to automate the program development processes by adding it to flow graph control and decision graphs.

3 Problem definition

TCP problem inregression testing are defined in Rothermel et al.[4] as follows:

Problem in Test Case Prioritization is defined as follows :

Given: A test suite T, PT, the set of permutations of T and f , a function from PT to real numbers. Problem:Find T′ ∈ PT such that (∀T″) (T″ ∈ PT) (T″ , T′) [f (T′)≥ f (T″)].

PT denotes the set of all possible ordering of T and f is a function that yields an award value when applied to any such ordering. Proposed by Li et al. [5], the Average Percentage of Statement Coverage (APSC) metric is calculated using below formula(1),

$$APSC\left(T'\right) = 1 - \frac{TS_1 + TS_2 + \cdots + TS_m}{mn} + \frac{1}{2n} \quad (1)$$

here m=total number of statements being evaluated in this program, n=number of test cases in T and TSi - is the first test case in T′ covering a statement.

4 Ant Colony Algorithm

AS algorithm used to solve a travelling salesperson problem but it does not delt with the progressive(state-of-the-art)algorithms.AS algorithm is the first algorithm,inspired by the action of real ants.AS algorithm has the benefit of implementing Ant Colony Optimization algorithms and demonstrating the promise of using artificial pheromone and ants to accelerate the search for ever fitter solutions to compounded problems of optimization. The forthcoming work was driven by two objectives:Theinitial objectiveis to enhance it's efficiency and then second was to analyze and describe its behaviour. A simpler version was transpired that retained approximately the balanced level of efficiency, calculated by sophistication of algorithms and numerical tests described in 1996 Ant Colony System (ACS)[6]. Because ACS is one of the basis of several algorithms described in following years, we concentrate on ACS rather than AS. Three major aspects varies ACS from the previous AS:

4.1 Pheromone

In ACS, after all of the ants have measured their tour (i.e. at the end of each iteration) AS updates the pheromone trail using all the solutions generated by the ant colony. Growing edge of one of the calculated solutions shall be changed by a pheromone quantity equal to its solution value. By the conclusion of this step, the pheromone of the whole system evaporates and the building and upgrading process is iterated. In the opposite, ACS uses the best solution computed after the beginning of the calculation to update pheromone globally. Like in the case of AS, global updates are meant to improve the attractiveness of the desirable path, but the ACS approach is more successful because it prevents long convergence time by specifically focusing the search in the neighborhood of the best route to the current iteration of the algorithm.

In ACS, the final evaporation phase is substituted by a local updation of the pheromone applied during the construction phase. Every time an ant moves from the current to the next, the edge-associated pheromone is changed as follows(2):

$$\tau(r,s) \leftarrow (1-\rho)\cdot\tau(r,s) + \rho\cdot\tau_0 \quad (2)$$

Here, $0 \leq \rho \leq 1$ is a parameter (set at 0.9) and the starting pheromone value is $\tau 0$.Theresult of local updation is to dynamically adjust the eligibility of the edges.When each time an ant is use an edge, this becomes moderately less preferable and pheromone remains $\tau 0$ just for the edges that were never part of a global best tour. A significant trait of local updating and global updating process at every edge's pheromone $\tau ij(t)$ is inferior constrained by $\tau 0$.

4.2 State transition rule

ISSN (Print): 2204-0595
ISSN (Online): 2203-1731

This rule was used during the development ofuptodateanswer,each ant agrees on the state to transit next by state transition rule step. A new state transition rule is introduced in ACS called pseudo-random-proportional.Byusing the random-proportional rule of AS, the next state is automatically chosen with a distribution of probabilities based on ηij and τij. Then theupcoming state is randomly chosenbased on ηij and τij weighted by α (equal to 1) and β (equal to 2) with a probability distribution.

5. Proposed approach

The proposed algorithm is constructed based on the Ant Colony System(ACS)[3] and is called the Coverage Based ACS(CB-ACS). A test suite T = {t1, t2, ..., tn} containing n test cases covering a total of m statements in a program tested is translated to a graph G = {V, E}. V is the set of vertices with n real vertices corresponding to the n test cases and one virtual vertex v-1.All ants start their traversal from virtual vertex. Ants are aimed at stopping when complete coverage of statements is obtained and are discouraged from unnecessary traversals to all vertices. Start index was one virtual vertex v-1 and n real vertices, in graph G we transform the n unordered test cases. The heuristic of CRP is extended to discover the first history better solution TH0. The quantity of τ0= APSC(TH0) pheromone is deposited on each common edge and also on the virtual edge in G, then all the artificial ants are initialized at v-1.

S is a set consisting of all statements. Sak denotes the set of statements this statement was already covered by an artificial ant k. The heuristic function is denoted by η, From vertex vr to vs, the heuristic value for ant k defined below using the formula(3):

$$\eta k(r,s)=|(S\backslash Sak)\cap Svs| \quad (3)$$

Where Svs is the covered statement sets of vertex vs, S\Sak obtains the complementary set of Sak within S. The above formulation returns the input set elements ηk(s) and determines the cost of moving ant k to s. k will consider and transfers to s at the stage of a candidate vertex s contains statements that was never covered by k at that iteration. Otherwise, ηk(s) becomes zero if Sakcontain all the elements in Svs, and k was constant. A vertex yields a larger heuristic if k covers more additional statements by moving to it, and yields smaller heuristics if it provides comparatively fewer additional statements for k. At the end of each transition, Sak value is updated by by using formula(4):

$$Sak \leftarrow Sak(Svs\backslash Sak) \quad (4)$$

computes the difference of Svs and Sak ,WhereSvs is the set of statements covered by the selected vertex s.

Using the PRP rule [11] defined below(5),

$$P_k(r,s)=\begin{cases}\dfrac{[\tau(r,s)]\cdot[\eta(r,s)]^{\beta}}{\sum_{u\in J_k(r)}[\tau(r,u)]\cdot[\eta(r,u)]^{\beta}}, & \text{if } s\in J_k(r)\\ 0, & \text{otherwise}\end{cases} \quad (5)$$

Ants attempt to transit vertices both with a larger number of statements and belong to the initial answer, with a biased probability escaping local optima. After each transition by LPU rule, on G all ants deposit the pheromone. Finally, all artificial ants are stop at a vertex, the full statement coverage is attained place.The sorting based local search(SB-LS) mechanism was applied by which the constructed solutions are fine-tuned and their fitness values are calculated. The GBT tree was revised by comparing the iteration-best fitness value and the historybest one accordingly. Finally,to update the amount of pheromone on G we use the proposed GBT tree was used which uses the GPU rule.

```
#include<stdio.h>
#include<math.h>
1.  void main()
2.  {
3.  double a,b,c;
4.  double a1,a2,a3;
5.  int valid=0;
6.  clrscr();
7.  printf("enter first side of triangle");
8.  scanf("%f",&a);
9.  printf("enter second side of triangle");
10. scanf("%f",&b);
11. printf("enter third side of ttriangle");
12. scanf("%f",&c);
13. if(a>0&&a<100&&b>0&&b<=100&&c>
    0&&c<100){
14. if((a+b)>c&&(b+c)>a&&(c+a)>b){
15. valid=1;
16. }
17. else{
18. valid=-1;
19. }
20. }
21. if(valid==1){
22. a1=(a*a+b*b)/(c*c);
23. a2=(b*b+c*c)/(a*a);
24. a3=(c*c+a*a)/(b*b);
25. if(a1<=1||a2<=1||a3<=1){
26. printf("obtuse angled triangle");
27. }
28. else if(a1==1||a2==1||a3==1){
29. printf("right angled triangle");
30. }
31. else{
32. printf("acute angled triangle");
33. }
34. }
35. else if(valid==-1){
36. printf("ninvalid triangle");
37. }
38. else{
39. printf("ninput values are out of range");
40. }
41. getch();
42. }
```

Fig.1.Triangle program

As in Fig 1,Triangle classification program[12] is used and we generated a CFG with each node covering different number of statements shown in Fig 2. Initially the program consists of 25 test cases each with different statement

ISSN (Print): 2204-0595
ISSN (Online): 2203-1731

coverage represented graphically in(). Applying our proposed approach using Conditional Random Push method we can obtain an initial solution with which τ0 is calculated [].

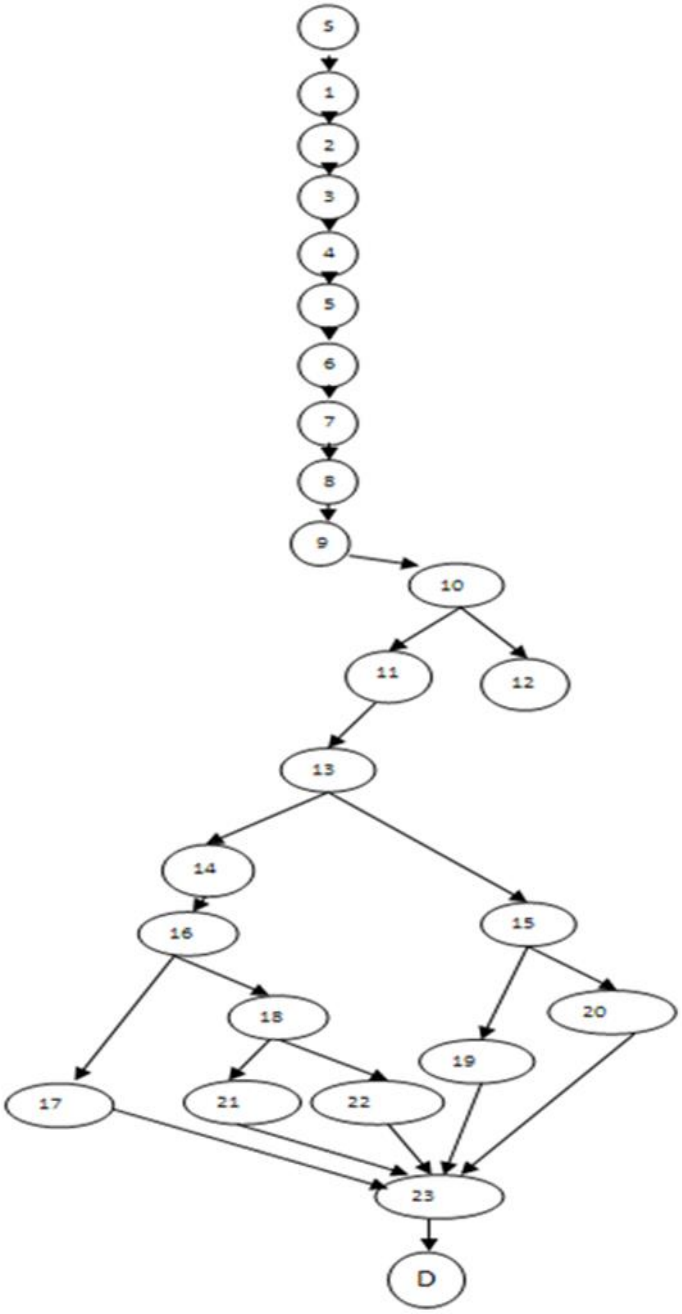

Fig. 2.CFG for Triangle program

According to our algorithm, ants traverse to each node with highest statement coverage and pheromone strength thereby achieving prioritization.The statement covered by each node is shown in table 1.The statement covered by each test case is shown in table 2.The highest priority is given to the node having the maximum combined strength of pheromone and heuristic. The traversal of nodes ends when full statement coverage is achieved that results in the reduction of the test suite size drastically.The graphical representation of test suite and reduced test suite shown in Fig 4 and Fig 5.. The resulting test suite size for the considered program is 5.

Table.1.Statements covered by each node

S.NO	NODES	STATEMENTS COVERED
1	S	1
2	1	2
3	2	3
4	3	4
5	4	5
6	5	6
7	6	7,8
8	7	9,10
9	8	11,12
10	9	13
11	10	14
12	11	15,16
13	12	17,18,19,20
14	13	21
15	14	22,23,24
16	15	35
17	16	25
18	17	28
19	18	26,27
20	19	36,37
21	20	38,39,40
22	21	29,30
23	22	31,32,33,34
24	23	41
25	D	42

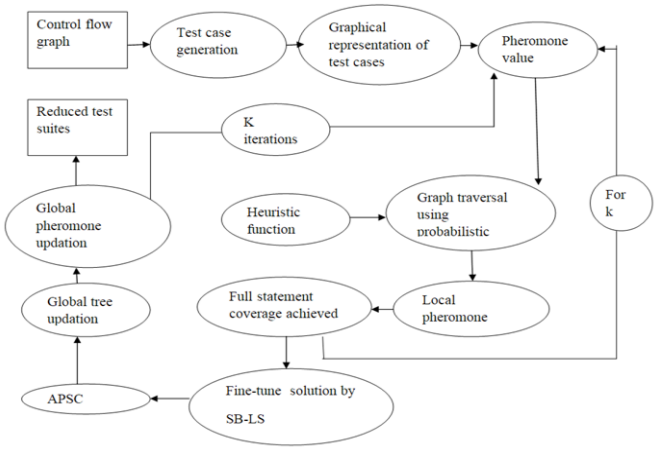

Fig 3.System Architecture Design

ISSN (Print): 2204-0595
ISSN (Online): 2203-1731

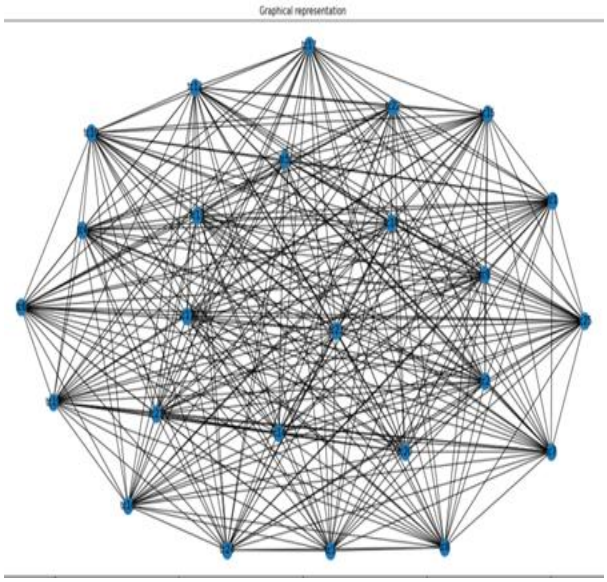

Fig4. Graphical representation of Test Suite

Table.2.Statements covered by each Test case

¡TEST CASES	INPUT	OUTPUT	NODES COVERED	STATEMENTS COVERED
T1	(101,88,63)	Input value out of range	1-9,20,23,D	(1-13,38-42)
T2	(3,4,6)	Obtuse angled	1-11,13,14,16,18,23,D	(1-15,21-27,41,42)
T3	(3,4,5)	Right angled	1-11,13,14,16,17,21,23,D	(1-15,21-25,28-30,41,42
T4	(2,3,4)	Obtuse angled	1-11,13,14,16,18,23,D	(1-15,21-27,41,42)
T5	(3,3,6)	Acute angled	1-10,22,23,D	(1-14,31-34,41,42)
T6	(10,3,4)	Invalid triangle	1-10,12,15,19,23,D	(1-14,17-20,35-37,41,42)
T7	(4,2,2)	Acute angled	1-10,22,23,D	(1-14,31-34,41,42)
T8	(98,108,201)	Input value out of range	(1-9,20,23,D)	(1-13,38-42)
T9	(0,0,0)	Invalid triangle	1-10,12,15,19,23,D	(1-14,17-20,35-37,41,42)
T10	(2,0,2)	Invalid triangle	1-10,12,15,19,23,D	(1-14,17-20,35-37,41,42)
T11	(7,2,2)	Invalid triangle	1-10,12,15,19,23,D	(1-14,17-20,35-37,41,42)
T12	(5,6,7)	Right angled	1-11,13,14,16,17,21,23,D	(1-15,21-25,28-30,41,42
T13	(5,12,13)	Right angled	1-11,13,14,16,17,21,23,D	(1-15,21-25,28-30,41,42
T14	(8,15,8)	Obtuse angled	1-11,13,14,16,18,23,D	(1-15,21-27,41,42)
T15	(11,18,15)	Obtuse angled	1-11,13,14,16,18,23,D	(1-15,21-27,41,42)
T16	(10,12,12)	Acute angled	1-10,22,23,D	(1-14,31-34,41,42)
T19	(15,20,8)	Obtuse angled	1-11,13,14,16,18,23,D	(1-15,21-27,41,42)
T20	(7,2,3)	Invalid triangle	1-10,12,15,19,23,D	(1-14,17-20,35-37,41,42)
T21	(10,5,5)	Acute angled	1-10,22,23,D	(1-14,31-34,41,42)
T22	(8,9,15)	Obtuse angled	1-11,13,14,16,18,23,D	(1-15,21-27,41,42)
T23	(4,1,1)	Invalid triangle	1-10,12,15,19,23,D	(1-14,17-20,35-37,41,42)
T24	(2,4,201)	Input value out of range	(1-9,20,23,D)	(1-13,38-42)
T25	(20,10,10)	Acute angled	1-10,22,23,D	(1-14,31-34,41,42)

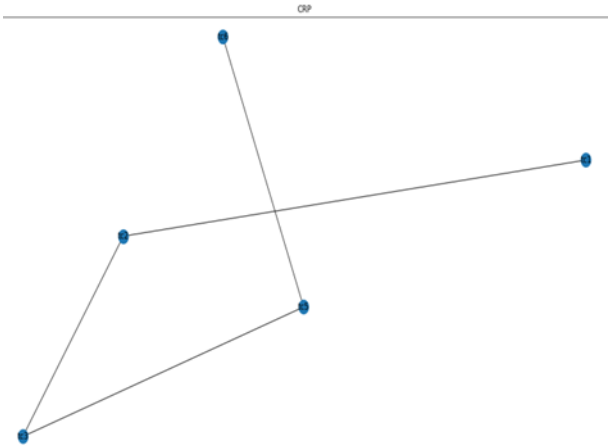

Fig 5. Graphical Representation of Reduced Test Suite

6. Results and conclusion

This paper used ACS for reducing the test suite, which automatically selects the optimum test path in a control flow graph by making use of CB-ACS. An Ant can effectively discover the CFG states and reduces the test suite for testing automatically. CB-ACS prioritize and the number of test cases for execution will be reduce according to the normal foraging action of ants and thus helps software testers to a great degree. In Ant Colony Optimization(ACO), Artificial Bee Colony Optimization (ABC), Particle Swarm Optimization (PSO), Genetic Algorithm, Simulated Annealing and are only a few other meta heuristic methods on which work can be done to harness species' natural intelligence and solve NP problems**.**

References

[1] R.Wang, B.Qu, Y.Lu.(2015).“Empirical study of the effects of different profiles on regression test case reduction”.IETSoftw, 9(2), pp.29–38.

[2] Yoo, S., & Harman, M.(2012). Regression testing minimization, selection and prioritization: a survey. Software Testing, Verification and Reliability, 22(2), 67-120. https://doi.org/10.1002/stv.430

[3] S.Karma, Y.Singh, S.Dalal and P.R.Srivastava, Test case prioritization: A approach based on modified ant colony optimization, In Proc.Int.Conf,Emerg.Res.Comput(2016)

[4] Y.Singh, a.Kaur and B.Suri, Test case prioritization using ant colony optimization, ACM SIGSOFT Eng.(2010)

[5] Dorigo, M., Gambardella, L.: Ant colony system: a cooperative learning approach to the traveling salesmanproblem. IEEE Trans. Evol. Comput. 1(1), 53–66 (1997)

[6] G. Rothermel; R. H. Untch; Chengyun Chu; M. J. Harrold. 2001. Prioritizing Test Cases for Regression Testing. IEEE Trans. on Soft. Eng. 27, 10 (2001), 929–948.

[7] Zheng Li; Mark Harman; Robert M. Hierons. 2007. Search Algorithms for Regression Test Case

ISSN (Print): 2204-0595
ISSN (Online): 2203-1731

Prioritization. IEEE Trans. on Soft. Eng. 33, 4 (2007), 225–237

[8] Dorigo, M., Maniezzo, V., und Colorni, A.: The Ant System: Optimization by a colony of cooperating agents. IEEE Transactions on Systems, Man, and Cybernetics Part B: Cybernetics. 26(1):29–41. 1996.

[9] Mr. Dharmesh Dhabliya, Mr. Rahul Sharma. (2012). Efficient Cluster Formation Protocol in WSN. International Journal of New Practices in Management and Engineering, 1(03), 08 - 17.

[10] N. Sethi, S. Rani, and P. Singh, "Ants Optimization for Minimal Test Case Selection and Prioritization as to Reduce the Cost of Regression Testing," Int. J. Comput. Appl., vol. 100, no. 17, pp. 48– 54, 2014.

[11] S. Yang, T. Man, and J. Xu, "Improved ant algorithms for software testing cases generation," Sci. World J., vol. 2014, 2014.

[12] C. Lu, J. Zhong, and C. Author, "An Efficient Ant Colony System For Coverage Based Test Case Prioritization," in GECCO '18 Companion, 2018, pp. 91–92.

[13] M. Mann and O. P. Sangwan, "Generating optimization and prioritizing optimal paths using ant colony," vol. 5, no. 1, pp. 1–15, 2015.

[14] O.Dhaiya and K.Solanki, "A Systematic literature study of regression test case prioritization approaches," Int.J.eng,Technol

[15] D.Yamuna, Dr.N.Sasirekha, An overview on test case reduction methods for data mining techniques, International Journal of Contemporary Research in Computer Science and Technology(IJCRCST)(2017)

[16] Marwah Allah, Dima Suleiman, Adnan S, Test case reduction Technique Survey(IJACSA)International Journal of Advanced Computer Science and Applications(2016).

ISSN (Print): 2204-0595
ISSN (Online): 2203-1731