Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 4, Issue. 4, April 2015, pg.189 – 194

RESEARCH ARTICLE

A Novel Test Cost Reduction Approach for Test Suites

Dr. Preethi Harris, Sathya.A, Pooja.S, Shanmuga Priya.S

Information Technology, Sri Ramakrishna Engineering College, Coimbatore, India

preeharris@gmail.com, satharun720@gmail.com, poojasha1109@gmail.com, priyadazzller93@gmail.com

Abstract- Software utilization has gained momentum over the last decade. The development of such software is largely dependent on testing the product for its reliability. With software testing being the major cost driver in the development of software, optimizing the test cost is a major problem in software organization. Recently, attempts have been made through heuristics and evolutionary algorithms like genetic algorithm to minimize test cost. Inspired by these attempts in literature, in this paper, ant colony optimization algorithm has been adapted to tackle increasing test cost. Experimentation has been carried out on real time applications to determine the minimum cost along the trails of the ants. The results obtained using MACO algorithm is compared with information gain.

Keywords– Software, Software Testing, Test Cost, Heuristics, Genetic Algorithm, Ant Colony Algorithm, Minimum Cost

I. INTRODUCTION

The development of software is determined by the choice of the right technology and skillset of people. It is widely believed that a well-designed software product is characterized largely by effective testing. The NIST report [12] states that, from 40-50% of the cost when developing the software is spent on testing to release a quality product. Software organizations often have very tight time and budget constraints, which will restrict the actual testing activity. Consequently, such implications insist on test cost reduction during the software development and software organizations insist on research activities to develop effective methods for software testing. Test cost reduction should focus on identifying a minimal cost of test cases after removing redundant attributes of the original test suite.

Evolutionary computation encompasses algorithms that mimic the principles of natural evolution [1]. Evolutionary algorithms use population-based random variation and selection to search a space of candidate solutions in light of performance information. The basic types of evolutionary algorithms include:

1. Genetic Algorithm
2. Particle Swarm Optimization
3. Ant Colony Optimization
4. Bee Colony Optimization

However, there are few efforts for applying some of these novel search-based optimization techniques in the area of software testing [2],[3],[4].Genetic algorithms belong to the larger class of evolutionary algorithms (EA), which generate solutions to optimization problems using techniques inspired by natural evolution, such as inheritance, mutation, selection, and crossover [1]. The genetic algorithm has been adapted for minimal test cost reduction problem [5] and attribute reduction with test cost constraint [6].

Bee colony optimization (BCO) algorithm is inspired by the intelligent foraging behavior of honey bees. The bees find the food sites (test cases) then the best test cases is computed (fitness)with less computing resources [7].

However, the findings of the literature reveal that Bee Colony Optimization is that premature convergence is the limitation of Genetic Algorithms and Bee Colony Optimization requires in-depth knowledge of mathematics to solve problems.

Ant Colony Optimization (ACO) algorithm is a probabilistic technique for solving computational problems which can be used to find optimal paths through the graph [8]. It is depends on the behavior of ants in finding paths from their colony to food [8],[9]. ACO can be adapted for test suite optimization because in real time the ACO can adapt to changes and hence in this paper the ACO optimization technique has been proposed as Modified Ant Colony Optimization (MACO) algorithm for reduction in test cost on three real time test sets. The next section describes the proposed test cost reduction algorithm.

II. TEST COST REDUCTION

The test cost reduction (Fig 1) begins with the test sets of the application as input. It is followed by applying the MACO algorithm. The performance evaluation of the experimentation compares MACO and the state-of-the art algorithm Information Gain (IG) [10].

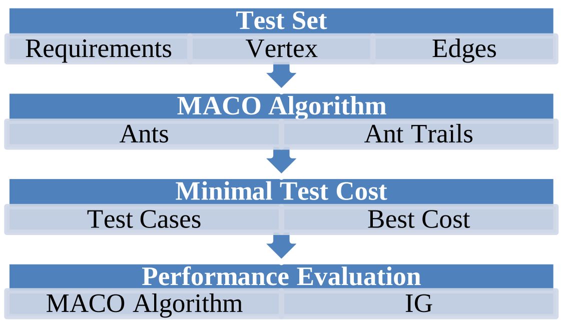


Fig 1. Block diagram of test cost reduction

The proposed MACO algorithm represents the test requirements as a complete graph. In this graph each vertex corresponds to one requirement associated with the given universal test suite. Each vertex also contains information pertaining to the test cost and each edge connecting a vertex has information about the cardinality of test cases. The test case cardinality representing the weights of each edge denotes the pheromone density. In each iteration ants are generated. The basic operations done by adapting ACO for test cost reduction is depicted in MACO algorithm.

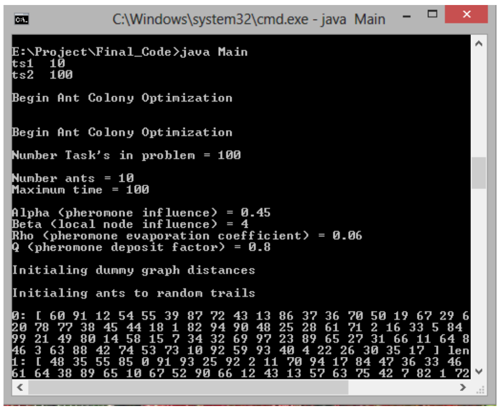
```
Algorithm MACO
Input:
Universal test suite T
Requirements R
Output:
Minimal test cost test cases, $T_{rs}$
Begin
for each test set $T_i$ in T
do
create ants $a_j$for the requirements $r_i$
end do
end for
for$a_j$in $T_i$
do
select the next vertex $v_i$ and compute the best costs $c_i$ for each test set $T_i$
if$a_j$ reaches the last vertex of last trail then
update the pheromones of each edge traveled by $a_j$
endif
end do
end for
$\lambda=\sum_{i=1}^{n}\frac{c_i}{n}$
for i in n
do
if($c_i< \lambda$) then
$T_{rs}\leftarrow T_{rs}$ U $t_i$
end if
end do
end for
End
```

The length of the ant trail in each iteration is recorded. Then the best paths are selected which represents best cost. The average costis computed for all the recorded paths as λ and the path with cost less than λ is the minimal test cost. The test cases corresponding to this condition is placed in the representative set $T_{rs.}$

III. RESULTS

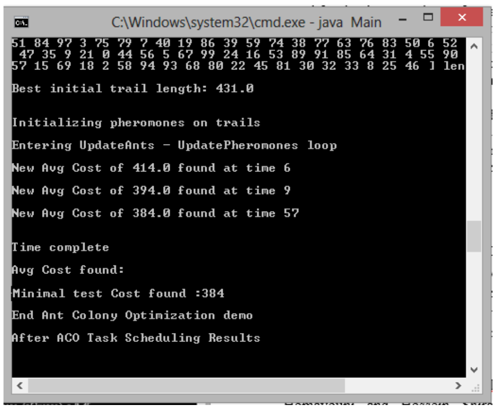
Windows 8 pro Operating system has been used, JDK 1.7 and COSER (Cost Sensitive Rough set)[12] has been used. The system requirements are: CPU: 3GHz, RAM: 6 GB, Storage 500 GB.

Experimentation has been carried out with four test suites from UCI data sets [13]. The output screenshots for generating ants and trails is depicted in Fig 2. The details of pheromone trails, cost determined in each ant trail, computation of average cost and minimal cost for the test suite considered is illustrated in Fig 3.



```
E:\Project\Final_Code>java Main
ts1  10
ts2  100

Begin Ant Colony Optimization


Begin Ant Colony Optimization

Number Task's in problem = 100

Number ants = 10
Maximum time = 100

Alpha (pheromone influence) = 0.45
Beta (local node influence) = 4
Rho (pheromone evaporation coefficient) = 0.06
Q (pheromone deposit factor) = 0.8

Initialing dummy graph distances

Initialing ants to random trails

0: [ 60 91 12 54 55 39 87 72 43 13 86 37 36 70 50 19 67 29 6
20 78 77 38 45 44 18 1 82 94 90 48 25 28 61 71 2 16 33 5 84
99 21 49 80 14 58 15 7 34 32 69 97 23 89 65 27 31 66 11 64 8
46 3 63 88 42 74 53 73 10 92 59 93 40 4 22 26 30 35 17 ] len
1: [ 48 35 55 85 0 91 93 25 92 2 11 70 94 17 84 47 36 33 46
61 64 38 89 65 10 67 52 90 66 12 43 13 57 63 75 42 7 82 1 72
```

Fig 2. Initialization of MACO algorithm



```
51 84 97 3 75 79 7 40 19 86 39 59 74 38 77 63 76 83 50 6 52
 47 35 9 21 0 44 56 5 67 99 24 16 53 89 91 85 64 31 4 55 90
57 15 69 18 2 58 94 93 68 80 22 45 81 30 32 33 8 25 46 ] len

Best initial trail length: 431.0


Initializing pheromones on trails

Entering UpdateAnts - UpdatePheromones loop

New Avg Cost of 414.0 found at time 6

New Avg Cost of 394.0 found at time 9

New Avg Cost of 384.0 found at time 57


Time complete

Avg Cost found:

Minimal test Cost found :384

End Ant Colony Optimization demo

After ACO Task Scheduling Results
```

Fig 3. Results of MACO algorithm

The real time test sets namely: Mushroom, Vote and Tic-Tac-Toe from UCI data set [13] are given as input to MACO algorithm implementation in Java. The test cost range generated for each ant trail and the minimum cost which is determined as the best cost is tabulated in table 1. The results obtained using MACO and IG is depicted in Fig 4. The results of experimentation shows that for Tic-Tac-Toe the minimal cost incurred was 402 which was higher than the minimum cost incurred for Vote and Mushroom test set.

TABLE 1

Results of experimentation for real time application test suites using MACO algorithm

S.No	Test Suites	Test cost range	Minimum Test cost
1.	Mushroom	380-420	384
2.	Vote	385-410	399
3.	Tic-tac-toe	390-420	402

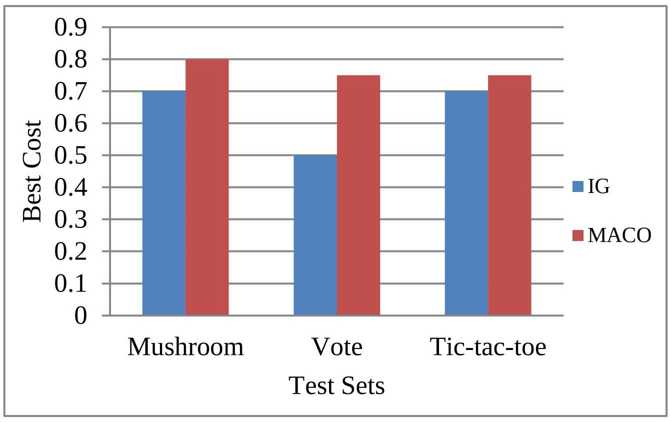


Fig 4. Best Cost obtained for the test sets using MACO and IG

IV. CONCLUSION

In the proposed approach, the ant colony optimization algorithm has been adapted to determine the minimal test cost for four real time application test suites is kept as the constraint. The results show that an average of 80% optimization in test cost is obtained using the proposed approach, in comparison to the information gain-based λ-weighted reduction algorithm which is 75%.

The results obtained for information is more formative in ACO when compared with information gain. Thus the results of experimentation show that when MACO algorithm is used the test cost is less than that obtained through IG.

REFERENCES

[1] Thomas Back, "Evolutionary algorithms in theory and practice," Oxford Press, New York,1996

[2] A. Bouchachia, "An immune genetic algorithm for software test data generation," Proc. of 7th International Conference on Hybrid Intelligent Systems (HIS'07), pp. 84-89. 2007

[3] Suriya, S, Deepalakshmi, R, Suresh Kannan, S &Shantharajah,"Enhanced Bee Colony algorithm for complex optimization problems,"International Journal on Computer Science and Engineering, vol. 4, no. 1, pp. 72-78, 2012.

[4] Bharti, S, Isha, M &Varun, "Regression test suite reduction using an hybrid technique based on BCO and genetic algorithm," Special Issue of International Journal of Computer Science and Informatics, vol.2, no.1& 2, pp. 165-172, 2010.

[5] P. D. Turney, "Cost-sensitive classification: empirical evaluation of a hybrid genetic decision tree induction algorithm," Journal of Artificial Intelligence Research, vol. 2, pp. 369–409, 1995.

[6] J. Liu, F. Min, S. Liao, and W. Zhu, "A genetic algorithm to attribute reduction with test cost constraint," in Proceedings of the 6th IEEE International Conference on Computer Sciences and Convergence Information Technology (ICCIT '11), pp. 751–754, 2011.

[7]NavdeepKoundal ,Ankur Shukla, Mohsin Rashid Mir, "Test Case Selection Using Bee Colony Optimization," International Journal of Science and Research (IJSR), vol. 3 no. 5, pp. 1432-1436, 2014.

[8] M. Dorigo and C. Blum, "Ant colony optimization theory: A survey,"Theoretical Computer Science, vol. 344, no. 2&3, pp. 243-278, 2005.

[9] ZilongXu, Fan Min, Jiabin Liu and WilliamZhu, "Ant colony optimization to minimal test cost reduction," Granular Computing(GrC), IEEE International Conference,2012.

[10] F. Min, H. He, Y. Qian, and W. Zhu, "Test-cost-sensitive attribute reduction," Information Sciences, vol. 181, pp. 4928–4942, 2011.

[11] NIST report 2010,"Free NIST Software Tool Boosts Detection of Software Bugs". Available from:
http://www.commerce.gov/blog/2010/11/09/free-nist-software-tool-boosts-detection-software-bugs, 2002.

[12]Coser:Cost-senstive rough sets,Available from: http://grc.fjzs.edu.cnrfmin/coser/" [2011].

[13] C. L. Blake and C. J. Merz, "UCIRepository of machine learning databases,Available from: http://www.ics.uci.edu/˜mlearn/mlrepository.html," 1998.