

# International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: [www.ijirset.com](http://www.ijirset.com)

Vol. 7, Issue 4, April 2018

## Application of Hybrid BF-PSO Technique in Multi-objective Test Suite Optimization

Renu<sup>1</sup>

M.Tech, Department of Computer Science, DCRUST Murthal, India

**ABSTRACT:** In order to optimize the large test suites produced by the existing approaches, in this paper hybrid BF-PSO Technique is introduced for multi-objective test optimization. The algorithm for hybrid BF-PSO is defined for the problem. The BF-PSO combines both algorithms BF and PSO. The aim is to make use of PSO ability to exchange social information and BF ability in finding a new solution by elimination and dispersal. In the BF-PSO, the unit length random direction of tumble behaviour can be decided by the global best position and the best position of each bacteria.

**KEYWORDS:** Test Suite Optimization, Bacterial Foraging, Particle Swarm Optimization.

### I. INTRODUCTION

Most approaches generate a large number of test cases for a particular model until a test adequacy criterion is achieved. The cost of executing, storing, and maintaining these large test suites can be reduced through test suite optimization techniques. Utilizing the available resources as much as possible is called optimization. Optimization process is an incremental problem. At each and every evolution the solution leads to the target function[5]. The test suite optimization produces a subset of the initial test suite that preserves the original test adequacy criterion by removing the redundant test cases with respect to the considered criterion. However, in real testing problems, there exist multiple test criteria, because a single ideal criterion is simply impossible to be formulated and achieved. Single-objective test suite optimization is not useful in practical, because testers typically have many different objectives. For example, a frequently optimization problem is to produce a minimal test suite which achieves maximal coverage of the model entities with a minimal execution cost.

This paper introduces hybrid bacterial foraging[8] and particle swarm optimization[7] for solving the multi-objective test suite optimization problem. Due to the complexity of this problem (it is exponentially related to the original test suite size), we chose the Multi-Objective Evolutionary Algorithms for solving them. Evolutionary algorithms (EAs) are population-based meta-heuristic optimization algorithms that use biology-inspired mechanisms like mutation, crossover, natural selection, and survival of the fittest in order to refine a set of solution candidates iteratively[6].

### II. RELATED WORK

Greedy Technique[1]: Given the original test suite T and test requirements R, the greedy technique iteratively selects a test case that satisfies the maximum number of unsatisfied test requirements until all the test requirements satisfied by T have been satisfied.

Harrold et al.'s proposed heuristic[2] algorithm is to find the minimum size of representative set of the test suite. The basis of this algorithm is to find essential test cases, which are defined as those test cases that when removed, some test requirements can never be satisfied. The algorithm repeatedly selects the most essential test case from the test suite, until the selected test cases can satisfy all the requirements of the original test suite.

Nachiyappan, Vimaladevi and SelvaLakshmi [3] proposed genetic algorithm for test suite reduction. Their approach used mathematical model for test suite reduction. The model built the initial population based on test history. The fitness value of test cases was calculated based on the block based coverage value and execution time of the test cases.

# International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: [www.ijirset.com](http://www.ijirset.com)

Vol. 7, Issue 4, April 2018

The test cases with optimum fitness were selected. Test cases which violate fitness constraint were rejected. Their approach reduced the test suite size with same coverage as original test suite.

A Kaur, D. Bhatt[4] proposed a hybrid algorithm of Particle swarm optimization and mutation. Tests with maximum fault coverage and lesser time execution are selected. The main focus of this approach is to find a test suite that finds maximum faults in minimum time.

### III. HYBRID BF-PSO ALGORITHM

The BF-PSO combines both algorithms BF and PSO. The aim is to make use of PSO ability to exchange social information and BF ability in finding a new solution by elimination and dispersal. In BFOA, a unit length direction of tumble behaviour is randomly generated. Random direction may lead to delay in reaching the global solution. In the BF-PSO, the unit length random direction of tumble behaviour can be decided by the global best position and the best position of each bacteria.

Flow Chart for Hybrid BF-PSO algorithm is shown below in fig 3.1. This algorithm consists of four prime steps:

1. **Chemo taxis:** Here, swimming and tumbling are the two prime ways which define the manner in which bacteria search for food. Swimming means moving in a pre-specified direction. Tumbling means moving in a completely new direction.
2. **Swarming:** For the algorithm to converge at the optimal solution, it is required that the optimum bacteria attract other bacteria so that together they converge at the solution point quickly. To achieve this, a penalty function is added to the original cost function on the basis of relative distances of each bacterium from the fittest one. Penalty function becomes zero when all the bacteria have reached to the solution point.
3. **Reproduction:** Here, the fittest bacteria are divided into two groups. The weaker set of bacteria are replaced by other more fit set of bacteria. This keeps the population of bacteria constant throughout the evolution process.
4. **Elimination and Dispersal:** Because of changes in environment some bacteria may be killed or may be dispersed to a new place. In BFOA, this phenomenon is simulated by liquidating some bacteria and initializing new replacements randomly in the search space. It helps in reducing the probability of being trapped in pre-mature solution point.

# International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: [www.ijirset.com](http://www.ijirset.com)

Vol. 7, Issue 4, April 2018

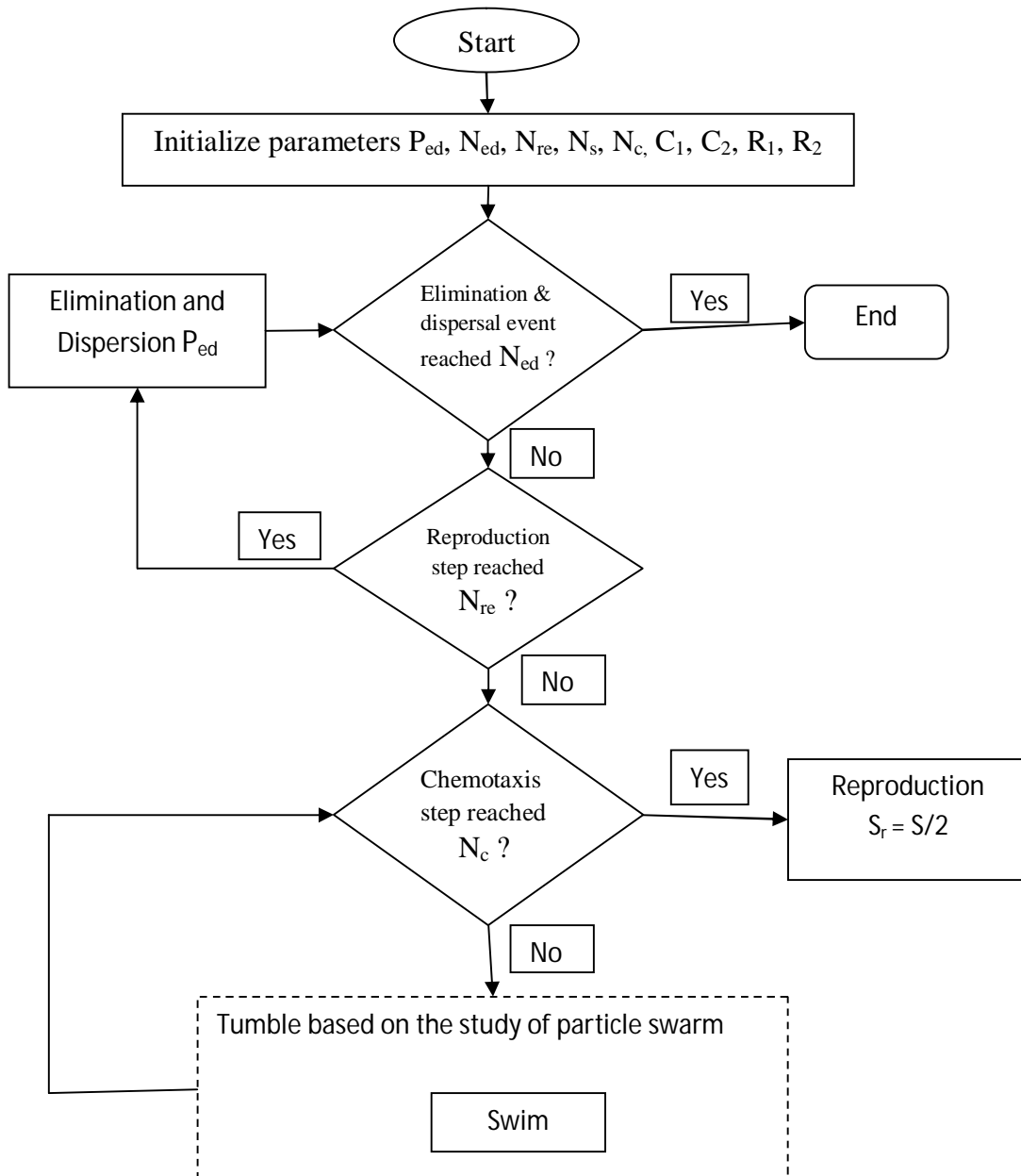


Fig 3.1: Flow Chart for Hybrid BF-PSO Algorithm.

## IV. EXPERIMENTAL RESULTS

In Proposed work we used the combination of bacterial foraging and particle swarm optimization(BF-PSO) algorithms to reduce the size of test suite. The major steps of the approach are:

# International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: [www.ijirset.com](http://www.ijirset.com)

Vol. 7, Issue 4, April 2018

Test Case	No. of faults covered	Execution Time(ET)
T1	4	7
T2	2	3
T3	3	5
T4	4	5
T5	3	3
T6	3	6
T7	3	3
T8	2	2

Table 4.1: Test cases, no. of faults covered, their execution time

STEP 1: Initialize all the parameters of the Hybrid Algorithms and give the Test case Requirement matrix as input to the program.

Test Case	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10
T1	x		x			x			x	
T2		x						x		
T3		x			x		x			
T4				x		x		x	x	
T5	x		x			x				x
T6				x	x			x		
T7	x						x	x		
T8			x							x

Table 4.2: Sample Data

STEP 2: Apply the BF-PSO algorithm which starts foraging with two particles initially.

Particles	B0	B1
Test Case Selected	T1	T4
Total Faults Covered	4	4
Total Execution Time	7	5

Table 4.3: Initial Selection of Test Cases

The test cases which covers maximum number of faults will be selected first.

STEP 3: In next iteration, the particles further choose the test cases as shown in table below :

Particles	B0	B1	B2	B3
Test Case Selected	T1, T3	T4, T3	T5, T3	T7, T3
Total Faults Covered	7	7	5	5
Total Execution Time	12	10	8	8

Table 4.4: Selection of Test Cases in Iteration II

# International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: [www.ijirset.com](http://www.ijirset.com)

Vol. 7, Issue 4, April 2018

STEP 4: In next iteration, particles choose next test cases that further covers the uncovered faults till now.

Particles	B0	B1	B2	B3	B4	B5
Test Case Selected	T1,T3,T7	T4,T3,T5	T6,T3,T8	T7,T3,T5	T1,T3,T6	T1,T3,T8
Total Faults Covered	8	10	7	7	8	9
Total Execution Time	15	13	13	11	18	14

Table 4.5: Selection of Test Cases in Iteration III

In iteration III, it is found that particle B1 has detected one path in 13 seconds that can cover all the faults.

STEP 5: Stopping Criteria- Algorithm will stop when either it has found a test suite which contains minimum number of test cases which covers all the faults or number of iterations are completed.

## Percentage of Test Suite Size Reduction

If „n“ is size of a test suite and „s“ be the selected test suite using hybrid approach, then the value for percentage reduction in test suite has been calculated using:

$$\% \text{ reduction in test suite} = (n-s)/n * 100$$

## Percentage of Execution Time Reduction

Let „T“ be the total execution time of the original test suite, „Tb“ be the running time of HBG\_Tool and „Ts“ be the total time for executing the selected test suite. The percentage reduction in execution time using hybrid approach is computed as:

$$\% \text{ reduction in Execution Time} = (T-(Tb+Ts))/T * 100$$

## V. CONCLUSION

In recent years, biological intelligent heuristic optimization algorithms have become one of the mainstream methods to solve the non-linear, non-differential, multi-peak and complex problems. In this dissertation we have used the hybrid BF-PSO optimization algorithm to minimize the number of test cases in a test suite. This algorithm is capable of producing an optimal sized test-suite taking execution time and coverage factors into account that can be inferred from the experimentation and analysis section. The results shows that coverage is not compensated in the reduced test set. Also, the reduced test-suite has the test cases that cover all the requirement of the program under test at least once.

## REFERENCES

- [1] Lingming Zhang, Darko Marinov, Lu Zhang, Sarfraz Khurshid, "An Empirical Study of JUnit Test-Suite Reduction", 22nd IEEE International Symposium on Software Reliability Engineering, 2011.
- [2] D. Binkley, "Semantics guided regression test cost reduction", IEEE Transactions on Software Engineering 23 (8), 498–516, 1997.
- [3] S.Nachiyappan,A.Vimaladevi, C.B.Selva Lakshmi, "An Evolutionary Algorithm for Regression Test Suite Reduction", Proceedings of the International Conference on Communication and Computational Intelligence – 2010, pp.503-508, ©2010 IEEE.
- [4] A. Kaur, D. Bhatt "Hybrid Particle Swarm Optimization for Regression Testing", International Journal on Computer Science and Engineering, Vol. 3 No. 5 (May-11), pp1815~1824, ISSN: 0975-3397
- [5] Siba Prasada Tripathy, Debananda Kanhar, "Optimization of Software Testing for Discrete Test suite using Genetic Algorithm and Sampling Technique", International Journal of Computer Applications (0975 – 8887), Volume 63– No.7, February 2013.
- [6] Thomas Wiese, Global Optimization Algorithms-Theory and Applications, Version-2009-06-26.
- [7] Qinghai Bai, "Analysis of Particle Swarm Optimization Algorithm", Computer and Information Science, Vol. 3, No. 1, February 2010.
- [8] Swagatam Das, Arijit Biswas, Sambarta Dasgupta, and Ajith Abraham. "Bacterial Foraging Optimization Algorithm: Theoretical Foundations, Analysis, and Applications".