International Journal of Computational Intelligence Research
ISSN 0973-1873 Volume 13, Number 1 (2017), pp. 65-78
© Research India Publications
http://www.ripublication.com

A Comprehensive Analysis for Software Fault Detection and Prediction using Computational Intelligence Techniques

Sushant Kumar
Department of Computer Science
Central University of South Bihar, Patna, India

Prabhat Ranjan
Department of Computer Science
Central University of South Bihar, Patna, India

Abstract

The computational intelligence based software engineering (CISE) is fruitful methods in recent trend for delivery of quality software. CISE is being effectively applied for achieving optimized result for all phases in software development life cycle (SDLC). This paper is more focus on computational intelligence techniques for fault handling to overcome the cost and time of testing and enhance the software quality. The fault handling is performed by fault detection and prediction, some important fault detection and prediction issues have been discussed. The different testing issues regarding fault handling has discussed here with computational intelligence techniques. Software fault prediction is used for early testing issue and test case optimization is useful for testing optimization issue. This technique will help to increase the quality of software. The result from different computational intelligence techniques has shown that they are performing very well. This paper also reflects that the fault prediction is more useful than fault detection since it will reduce the cost and time of testing. Both the methods are playing important role however the prediction will help to identify the fault in early phase.

Keywords: Computational Intelligence based Software Engineering, Test case optimization, Fault Detection, Fault Prediction.

1. INTRODUCTION

In a software development life cycle [1] the software testing and maintenance phase [2] is the most important and crucial step to ensure software quality. Increasing demand for software and chance of more dependent on software it is highly desirable for software to meet customer requirements. Software may be changed several times as per need and modification. The changes may modify the old or add new requirements for quality software. However it is extra burden on developer and engineer. The researchers are interested to find other methods that help to find the fault in early phase. It will reduce the cost of different testing technique that perform on development and maintenance phase[3-5].The scope of regression testing is always important because it has performed to make sure that modification is correct and have not adversely affected the unchanged portion of the software. Test suites that are previously available from earlier versions of software can be expensive to execute completely with modified test suites. Different techniques are proposed by researchers for reducing cost related to software testing and regression testing. They include regression test selection [6], regression test prioritization [7], and hybrid approach (selection and prioritization). The optimization will perform on available test data for detection of fault. The prediction of fault is helpful for management the test data and fault result will available in early process. Since it is the problem of NP complete [8] so it can be better solved by intelligence technique. This type of application is also useful in other field of engineering like embedded system for software engineering optimization for fault handling [9]

.

1.1 Computational Intelligence Techniques

Computational intelligence and meta-heuristic algorithms have become increasingly very popular in computer science, information technology, artificial intelligence, machine learning, engineering design, data mining, bio-Informatics, image processing, and data-intensive applications. Most algorithms in computational intelligence and meta-heuristic optimization are part of swarm intelligence. Computational Intelligence Technique is a branch of nature inspired computational methodologies and approach to handle complex real world problems. In this paper we have applied Computational Intelligence techniques for fault detection and prediction.

1.2 Recent Issues in Software Fault Detection and Prediction

The objective about this paper is map the computational intelligence in software engineering problem with more focus on complete detection and early prediction of the software fault. The handling of fault in software is main work of testing. The different issues like detect the fault through test suite and fault that is predict in advance is very crucial. This all issue will overcome the cost and time of testing. We have discussed five issues in this paper which also include relation with fault detection and predictions and computational intelligence methods.

1.2.1 What different testing approaches has solved using computational intelligence technique?

1.2.2 What different testing optimization techniques solved by computational intelligence techniques?

1.2.3 What is the merit and demerit of different computational testing techniques that are applied in software engineering?

1.2.4 What is the impact in terms of cost and time using detection and prediction technique for software fault?

1.2.5 What is the different validation technique available for detection and prediction?

The other section includes with second section is about search based computational intelligence based optimized fault detection and prediction. The comparison of computational intelligence technique is shown in third section. The third part is also including computational intelligence techniques for fault predictions. The forth section is conclusion part and the last part is future work.

2. Computational Intelligence Techniques for Optimized Fault Detection

Algorithms has classified as deterministic or meta-heuristic. If an algorithm works for mechanically deterministic manner, it is called deterministic. Generally it is provide same solution if our initial point is same. It will desirable to provide in randomness in solution and it will give better result. Nature-inspired algorithm is newly popular used algorithms for optimization in many engineering field of application. Here Meta - heuristic has applied for test case optimization problem. Testing optimization may be performed in black box testing and white box testing. Tester work to optimize the test case for both the testing type and found the fault as early time [10].The different algorithm is used for test case optimization like genetic algorithm, particle swarm optimization, artificial bee colony, ant colony and many more. The most popular algorithm for nature inspired is genetic algorithm. Most of the papers are using Genetic algorithm to solve the test case optimization problem for fault detection purpose.

2.1 Genetic Algorithm

Genetic algorithms are inspired by Darwin's theory about evolution. Genetic Algorithms (GAs) are adaptive search techniques premised on the evolutionary ideas of natural and genetic selection. Algorithm is started with a set of solutions (represented by chromosomes) called population. Solutions from one population are taken and used to form a new population. This is motivated by a hope, that the new population will be better than the old one. Solutions which are selected to form new solutions (offspring) are selected according to their fitness. Crossover and mutation operator used in genetic algorithm for new population and diversity .The genetic algorithm is widely applied for test case optimization. The test case optimization includes the generation, selection and prioritization [11-15]. The result is also validated with different data sets.

2.2 Particle Swarm Optimization

Particle swarm optimization is compare to genetic is new technique to use for optimization problem. A complex problem is solved using PSO for optimization like traveling sales man problem, Job sequencing problem. In software testing PSO is also used because testing is complex problem and generation of test case is NP-complete problem. PSO can be used for solving either single objective or multiple objectives [16].PSO is important optimization technique to optimize different complex problem. It is also simple for optimization it has only two basic equation-1 is for velocity and equation-2 is for new position.

v[] = wv[] + c1 * r1() * (pbest[] - present[]) + c2 * r2() * (gbest[] - present[]) (1)

present[] = present[] + v[] (2)

Here, w, c1, c2, r1, r2 are constants, pbest are local best and gbest are global best of particle. For software testing problem particle is represented the test case and test case is managed by objective function to focus on satisfy the requirement. If we compare with genetic algorithm here particle is chromosome. PSO has producing good result compare than genetic algorithm in terms convergence speed. Test case optimization performs very effective through PSO for fault detection in less time [16-19].

2.3 Ant Colony Optimization

Ant colony optimization (ACO) is a meta-heuristic technique for solving computational problems. It is best effective approach for reduce the path and find optimal path. Marco Dorigo has proposed ant colony optimization. The real power of ACO is the behavior of ants to find a path between their colony and source of food. In ACO ants are blind and communicate with their colony using chemical pheromone. Ant colony optimization is applied for test case optimization is to find the maximum fault in minimum time. The ant is solution here. Collection of ant represents the test suite. Authors have performed ACO on optimization for good result [20-23].

2.4 Artificial bee colony

Artificial bee colony (ABC) has proposed by karaboga in 2005. It is an optimization algorithm based on the intelligent behavior of honey bee. In ABC, the colony consists of three groups of bees 1) Employed bees 2) Onlooker bees 3) scouts bees. One artificial bee for each food source i.e. the number of employed bees equal to the number of food sources near hive. Soma sekhara et.al.[24] used the artificial bee colony optimization for generation of path. Ruchika Malhotra et.al[25] has proposed a new and implemented a new approach for test suite optimization using Mutated Artificial Bee Colony. It used both local search and global search. Author also compared different Meta-Heuristic algorithm for test case optimization. Author verified and validates the algorithm by running 10 benchmark C++ programs

2.5 Memetic algorithm

Memetic algorithm is advanced hybrid approach for genetic algorithm. It supports local and global search. Gordon Fraser et.al.[26] has extended the work of previous paper "Whole test generation". Author has used memetic algorithm for branch coverage. The evosuite tool is used for implementation of paper. This paper has more focused on local search operators.

2.6 Cuckoo search

Cuckoo search is comparatively new search technique in nature inspired computing. It has local search ability and also global search ability. But the main advantage of cuckoo search is for global optimization problem. Test case prioritization is solved by the cuckoo search and author shows the result that it also provide good result for global optimization problem [31]. Cuckoo search is also simple to solve, we can solve the cuckoo search by following three steps.

- Each cuckoo lays one egg at a time and dumps it in a randomly chosen nest.
- The best nests with high-quality eggs will be carried over to the next generations.
- The number of available host nests is fixed , and the egg laid by a cuckoo is discovered by the host bird with a probability either 0 or .The host bird can either get rid of the egg or simply abandon the nest and build a completely new nest. [27]

2.7 Harmony search

Harmonic search is a music inspired algorithm. It is given by Zong W. Geem in 2001 on first time, he was inspired by improvisation of musicians. Harmony search also uses gradient search. Many engineering problem has been solved using harmony search. It is also used in test case optimization. Ming huang .et.al [28] has improved the HS algorithm for test case selection.

2.8 Other computational intelligence techniques

Apart from genetic algorithm ,particle swarm optimization, Ant colony optimization ,Artificial bee colony optimization and Memetic algorithm the other algorithm that applied on test case optimization are tabu search , Hill Climbing, Simulated Annealing , Artificial Immune system and more, some of the algorithm defined in Table-1. Meta-heuristic algorithm has applied in test case optimization in 3 categories like generation, selection and prioritization, table 2 shows the different category. Fuzzy logic and artificial neural network is also used for test case optimization. Ali M. Alakeel (2014)[29] has used Fuzzy logic for Test case Prioritization for regression Testing. Assertion has imposed using fuzzy logic for test case prioritization. The developed approach is estimate the effectiveness of test case. Abraham et.al [30] has used hybrid approach ABC, FCM (Fuzzy C Means) and PSO for test case optimization to minimize the cost and time in regression testing. The experiment is done for test case prioritization. Chintala et.al. [31] has used neural network test effort estimation .Author estimate the testing effort by pre-coding and post-coding phases using neural network.

3. Computational intelligence techniques for software fault handling

The five issues that are discussed here will play important role to overcome the cost and time of the software. The fault detection and prediction has provided the detail about all fault in software. The manual and conventional method is not much effective. So the application of computational method is very important for handling the fault. The Computational Intelligence Techniques have applied for testing approach to cover the maximum fault. The selection of computational techniques is also important for maximum fault handling. Cost unit analysis performs for detection and prediction of fault.

3.1 Analysis of different testing approaches using computational Intelligence techniques.

The analysis of different testing approach using computational intelligence has used to find the testing coverage technique, like path testing, mutation testing and branch coverage for solving the optimization of test case. The path testing is popular coverage method of program and validation of test case. Other approach branch coverage, mutation testing, slice based testing etc. is also used for fault handling in software. In table 1, different testing approaches have used for fault handling in software. This result is useful for choosing the useful testing approach with good computational techniques. The testing approach has used the computational technique (CIT) for full coverage of program like path testing, mutation testing, functional testing etc. The different CIT methods are applied in testing approach for better result like Genetic Programming is effectively used in mutation testing for quality test case. The fuzzy logic is also applied for optimization of test case.

Table - 1 Software Testing problem and Computational Intelligence Techniques

Testing approaches	**Computational Intelligence Techniques**	**Author**
Path testing	Artificial Bee Colony, Genetic Algorithm, Genetic Algorithm, Firefly Algorithm, Genetic Algorithm, Firefly Algorithm	Soma Sekhara babu et.al.[24] Ahmed S.Ghiduk [12] P.R.Srivastava et.al. [15] Praveen R.S. et.al.[19] Lodha G.M. [10] P.R.Srivastava et.al.[32]
Mutation testing	Genetic Programming	William B. Langdon et.al.[11]
Functional testing	Genetic Algorithm,ACO	Javier Ferrer et.al.[22]
Branch coverage	Genetic Algorithm Memetic Algorithm	Gordan Fraser et.al.[14] Gordon Fraser et.al. [26]

Test case optimization	ACO	Chengying Mao et.al.[21]
	RAPSO	Shujaun Jiang et.al.[17]
	Fuzzy Logic	Ali M. alakeel [29]
	ACO	Elanthiraiyan, N.[33]
	ABC	Mala D. et.al.[43]
	Fuzzy Logic	Chin-Yu Huang et. al.[47]

3.2 Analysis of different test case optimization methods using computational intelligence techniques

Test case optimization is used for reducing, minimizing and prioritizing the test case so that we can reduce the cost and time. For optimization of test case prioritization is one of the important and crucial tasks, because it decides the time bound for testing. In the given time slot all the critical test case must be execute to find the fault. So need the good prioritization technique to search the all test suite (collection of test case) for finding the critical test case. In table-2, we have defined the different test case optimization approach. The test case optimization method is selected according to available cost and time. If we have very less time to execute the test case test case prioritization method is very effective. The test case generation is the first step for optimization of testing process. The different Nature-Inspired algorithm is used for test case generation to produce good test case. The Genetic algorithm is most popular used for generation. The test case selection is used for only selecting from the large available good test suite. It is selected through some objective function like path coverage, code coverage etc. The details are also available in different testing approaches. The last and important method is test case prioritization. This method is very effective for regression testing. The older method is also used for test case optimization like model based optimization is popular in UML to model the requirement.

Table - 2 Test case optimization methods

Test case optimization	**Author**
Test case generation	Bestoum et.al.[18] ,Ahmed s. Ghiduk et.al.[],Soma sekhara et.al.[24],Gordon Fraser et.al.[14],Javier Fraser et.al.[22],Gordon Fraser et.al.[26],C.Mao et.al. et.al.[21],P.R.Srivastava et.al. [19],Shujuan Jiang et.al. [17],Shujuan Jiang et.al. [34],P.R.Srivastava et.al. [32],Gordan Fraser et.al.[35] R.Malhotra et.al.[42]
Test case selection	Luciano et.al.[36],Luciano et.al. [16], Lucianoet.al. [6], Swarendu biswas et.al.[37]

Test case prioritization	Yogesh singh [38] , Md. Saeed Siddik et.al.[39], AliM.Alakeel[29],Elanthiraiyan et.al.[33],G.Rothermal et.al.[7] ,Khin hla et.al.[9]

3.3 Analysis for fault detection and prediction convergence result using computational intelligence algorithms

The search algorithms to be used for software engineering are from two main groups. The first group includes classical and old techniques like branch and bound algorithm, and linear programming. The classical algorithms are deterministic, and they are generally determining only one solution. The other important group includes the Meta – heuristics methods such as Evolutionary Algorithms, Particle Swarm Optimization, Ant Colony Optimization, and many others. The Meta -heuristics are the most preferred in the SBSE field. A reason for this is the nature of the Software Engineering problems are real world problems, and generally related to objectives that cannot be characterized by a set of linear equations, and they are not tractable by deterministic methods [49]. In our analysis we cover only Meta heuristics type algorithm for comparison. In Table 3, based on the analysis of different meta-heuristic algorithm apply in software testing we have done the comparison of algorithms. The pros and cons are provided in the table form. This analysis help to select which algorithm is more reliable and suitable for test case optimization problem. Meta-heuristic algorithm which has easy to implement, less parameters to adjust and have local and global search ability is more reliable and near to solutions. The complexity of algorithm is also decided that the algorithm is effective for both type of testing i.e. black box testing and white box testing. Since test case optimization is NP-Complete problem so select the best algorithm that can solve NP problem to solve optimization problem for software testing.

Table - 3 Comparison of Computational Intelligence Techniques for convergence

Algorithm	**Advantages**	**Disadvantages**
Genetic Algorithm	Answer gets better with time Do not break easily	Give slow results Weak local search ability
Artificial Bee colony	Good quality solutions of difficult optimization problem can be discovered	Extra overhead
Hill climbing	Simple Give Fast results	Leads to local optima, Highly dependent on stating solutions
Simulated annealing	Less dependent on the starting solutions, Less restrictions	Stuck in local optima, Cooling problem
Memetic algorithm	Strong local search ability, Hybrid approach	Gives slow result ,Hard to implement

Ant colony optimization	Record memory to entire colony, Less affected by initial poor solutions	Convergence is guaranteed Time of convergence uncertain
Particle swarm optimization	Easy to perform, Efficient in global search, Few parameters to adjust	Weak local search ability, In refined stage convergence is slow
Fuzzy Logic	Best for fuzziness data, Easy to construct and understand	Huge Number of Rule, Accuracy Problem

3.4 Analysis for cost units in phase wise impact for fault Detection and Prediction.

The detailed study perform by different paper shows that [3-5] the impact of prediction is more valuable than fault detection. If we know the fault knowledge in early phase then system engineers have sufficient time to plan about the fault. Kumar et.al[4] have defined the cost of early phase of removal of cost is very low compare than in last phase. The cost is exponentially high in later phase and handling of fault is difficult. The average result for a system with 20 K LOC (lines of code) need 7 weeks to run all test cases[47].The table 5 clearly shows that it is very costly in maintenance phase. In the last phase or deployment phase we perform regressing testing for any modification

Table 4. Cost in Development and maintenance Phase wise

Phase Name	**Cost In Units**
Requirement	1
Design	10
Code	100
Testing	1000
Maintenance	10000

The data coming from organization motivate the researcher to think about fault handling in less cost of units. The prediction of fault in early phase is not easy task, it will need user expert, human intelligence and many intelligence technique related to automation are needed. Since manual it is not possible we require a good computational intelligence technique for predict it. Table no 5 has given the clear view for prediction to reduce the burden of testing and maintenance. Tracy Hall [44] says that the fault prediction is very important in today environment to take the decision very fast and less time. It will benefit the organization as well as end user for safety and quality of the product. The 50 to 80 percent cost [45] is very problematic for any organization. The latest report publish in SEI/CMU [46] is also justify that the cost of prediction is very less .The fault detection and correction is also important for handling of fault in real time[48]. Since there are many version of software is release to compete with other similar product.

3.5 Analysis for performance of validation methods for Detection and Prediction.

The performance evaluation for detection and prediction is in different way. The detection of fault is focus on find the fault with minimum time and minimum test case. Different optimization technique like Test case prioritization, minimization are used. For the test case prioritization APFD methods are widely used. It is the average percentage fault detection. Other methods like test case selection based upon objective function are used. The prediction of fault is related to early working of optimization. It works on every phase to find the faulty module or non-faulty module. Classification methods are widely used in prediction method. In table -5 we have defined different evaluation technique for fault detection and prediction. The validation of result produced by CIT methods is very crucial since it produce the level of fault in program. The fault prediction approach is widely used the MMRe and BMMRe method for level of fault.

Table 5: Computational Intelligence Performance evaluation method for fault Detection and Prediction

Software Fault Detection/prediction	**Evaluation Method**
Fault Detection	• APFD-Average Percentage Fault Prediction • Branch Coverage • Code Coverage • Execution Time • Fault Detection Rate
Fault Prediction	• MMRE – Mean magnitude of relative error • BMMRE – Balanced mean magnitude of relative error • R2 - coefficient of determination • MdMRE - Median magnitude of relative error • Accuracy rate, ROC, Precision and Recall

4. CONCLUSION

The application of computational intelligence techniques for solving software engineering problems is part of a relatively new area. This is also defined through artificial intelligence of software engineering called search based software engineering (SBSE). The search based software engineering is using meta-heuristic approach for solving test case optimization problem. The test optimization techniques have been applied by many researchers for automatic test case generation, selection and prioritization but not any one could achieve the best performance for every piece of code to select the test case and also to find early fault in testing process. Due to large test case, the test case optimization has become a complex problem. So the scope remains open to apply some new computational intelligence techniques to achieve better results. The other important methods for fault handling are software fault prediction. This will predict the fault in early phase sometimes without start of the project. This method gives time to software engineer and tester to plan about their phases early. Since the handling of fault in later phase is exponentially high. Due to increase the agility in software delivery everyone is facing the shortage of time. The prediction technique will also play important role for guide the fault detection methods for which part is needed more testing.

5. FUTURE WORK

It has found from the different issues that prediction is more valuable for organization to reduce the failure of software. If fault prone modules discover in advance it well helpful in SDLC phases for software quality. In our future work we have to propose a model for the software fault prediction using computation intelligence techniques or meta-heuristic algorithm. The proposed model will help to reduce the cost and times of software testing for enhance the software quality.

REFERENCES

[1] Ian Somerville, “Software Engineering Eight edition”, Pearson Edition India, 2009

[2] Aditya P.Mathur, Foundation of software Testing,(Pearson Education, India,2007)

[3] Sushant Kumar ,Prabhat Ranjan and R.Rajesh, “An Overview of Test Case Optimization using Meta-Heuristic Approch”, Recent Advances in mathematics, Statics and Computer Science, (2015)

[4] Sushant Kumar , Prabhat Ranjan and R.Rajesh “A Concept for Test Case Prioritization Based Upon the Priority Information of Early Phase”, Lecture Notes in Electrical Engineering, (2016)

[5] Yogesh Singh, Ruchika Malhotra, “Object-Oriented Software Engineering”, PHI India (2012)

[6] Luciano S.de Souza, Ricardo B.C.Prudencio,Flavia de A. Barros, “Search Based constrained test case selection using execution effort”,Expert Systems with Applications40(2013)

[7] Gregg Rothermel,Roland H. Untch, Mary Jean Harrold, “Prioritizing Test Cases For Regression Testing”, IEEE TRANSACTIONS ON SOFTWARE ENGINEERING,27 (2001)

[8] Manoj kumar, Arun Sharma,Rajesh kumar, “Towards Multi-Faceted Test Cases Optimization”, Journal of Software Engineering and Applications,4, 550 (2011)

[9] Khinhla, YoungSik choi, Jong Sou Park, “Applying Particle Swarm Optimization to Prioritizing Test Cases for Embedded Real Time Software Retesting”, IEEE International Computer and Information Technology Workshops,(2008)

[10] Lodha, G.M., Gaikward, R.S., “Search Based Software Testing with Genetic using fitness function”, IEEE conference on Computational Intelligence on Power,Energy and Controls with their impact on Humanity CIPECH)(2014)

[11] William B.Langdon ,Mark Harman “Optimizing Existing Software with Genetic Programming”, IEEE Transaction On Evolutionary Computation,19,(2015)

[12] Ahmed S. Ghiduk, Automatic Generation of basis test paths using variable length genetic algorithm , “Information Processing Letters”114, 304 (2014)

[13] Samatha K. shreesha Chokkadi, Yogananda Jeppu, “A Genetic Algorithm Approach for test case optimization of safety critical control”, InternationalConference on Modeling, Optimization and Computing(ICMOC 2012) Procedia Engineering38 647 (2012)

[14] Gordon Fraser ,Andrea arcuri ,Phil Mcminn, “A Memetic Algorithm for hole test suite Generation”, The Journal of System and software, Elsevier (2014)

[15] Praveen Ranjan Srivastava, Tai-hoon Kim, “Application of Genetic Algorithm in Software Testing”, International Journal of Software Engineering and its application3, 4 (2009)

[16] Luciano S. de Souza,Ricardo B.C. Prudencio,Flavia de A. Barros, “Multi-Objective Test Case Selection : A study of the influence of the Catfish effect on PSO based strategies”, Anaios do XV Workshop de Testes e Tolerancia a Falha –WTF (2014)

[17] Shujuan Jiang, jiaojiaoShi,Yanmei Zhang, Han, “Automatic Test Data generation based on reduced adaptive particle swarm optimization”, Neurocomputing,Elsevier (2015)

[18] BestounS.Ahmed, Mouayad A.Sahib,Moyad Y.Potrus, “Generating combinational test cases using Simplified Swarm Optimization (SSO) algorithm for automated GUI functional testing”, Engineering Science and Technology, An International Journal(2014)

[19] Praveen Ranjan Sriatsava, B. Mallikarjun, Xin -She Yang, “Optimal Test Sequence Generation using firefly algorithm,Swarm and Evolutionary Computation”, Elsevier,8, 44 (2013)

[20] B suri,shweta singhal, “Analyzing test case selection and prioritization using ACO” ,ACM SIGSOFT Software Engineering Notes,(2011)

[21] Chengying Mao , lichuan Xio, Xixin , Jinfu Chen , “Adapting Ant Colony Optimization to generate test data for software structural testing”, Swarm and Evolutionary Computation,Elsevier (2014)

[22] Javier Ferrer, Peter M.Kruse ,Francisco Chicano, Enrique Alba, “Search Based Algorithm For Test Sequence Generation In Functional Testing”, Information and Software Technology(Elsevier) 58, 419 (2015)

[23] Sushant Kumar ,Prabhat Ranjan and R.Rajesh, “An Overview of Test Case Optimization using Meta-Heuristic Approch”, Recent Advances in mathematics, Statics and Computer Science, (2015)

[24] Soma Sekhara Baba Lam, ML Hari Prasad Raju, Uday Kiran M, Swarj Ch,Praveen Ranajn Srivastav, “Automated Generation of Independent Paths and Test Suite Optimization Using Artificial Bee Colony”, Procedia engineering, Elsevier30,191(2012)

[25] R. Malhotra and M. Khari, Test Suite Optimization using Mutated Artificial Bee Colony, “Proc. of Int. Conf. on advances in Communication, Network and Computing”, CNC, Elsevier,(2014

[26] Gordon Fraser, Andrea Arcuri, “Whole Test Suite Generation”, IEEE Transaction On Software Engineering, 39,2 (2013)

[27] Xin-She Yang, Suash Deb, “Cuckoo search :Recent advances and Applications”, Neural Computer Application(springer), 24, 169,2014

[28] Ming Huang,Shujie guo, Xu Liang,Xuan jiao, “Application of Improved harmony search algorithm in test case selection”, journal of software, 9 5 (2014)

[29] Ali M. Alakeel, “Using Fuzzy Logic in Test case prioritization”, The Scientific World Journal,Hindwai Publishing Crop. (2014)

[30] Abraham Kiran Joseph and G.Radhamani, “Fuzzy C means Clustering Based hybrid Swarm Intelligence algorithm for Test Case optimization”, Research journal of Applied sciences,engineering and Technology8 76 (2014)

[31] Chintala Abhisek, Veginati Pavan kumar,Harish Vitta, Parveen Ranjan Srivastava, “Software Engineering and application”, 3, 331 (2010)

[32] Parveen Ranjan Srivatsava, B.Malikarjun,Xin-She Yang, “Optimal test sequence generation using firefly algorithm”, Swarm and Evolutionary computation8 44 (2013)

[33] Elanthiraiyan, N, Arumugam, C.,Parallelized, “ACO algorithm for regression testing prioritization in hadoop framework”, IEEE International Conference on Advanced Communication Control and Computing Technologies (ICACCCT), (2014)

[34] Shujuan Jiang, Yanmei Zhang, Dandan Yi,Test, “Data Generation Approach for Basis Path Coverage”, ACM SIGSOFT Software Engineering Notes37,3 (2012)

[35] Gordon Fraser,Andrea Arcuri, Phil McMinn, “A Memetic Algorithm for whole test suite generation”, Journal of System and software,Elsevier (2014)

[36] Luciano S.de Souza ,Ricardo B.C.Prudencio, Flavia de A. Barros, “A Multi objective Particle Swarm Optimization for Test Case Selection Based on Functional Requirements Coverage and Execution Effort”, IEEE International Conference on Tools with Artificial Intelligence,(2011)

[37] Swarendu Biswar and Rajib Mall, “Regression test selection Techniques : A Survey”.Informatica35 289 (2011)

[38] Yogesh Singh, “Systematic Literature Review on Regression Test Prioritization techniques”, Informatica 36 379 (2012)

[39] Md.Saeed Siddik, Kazi Sakib, “RDCC : An Effective Test Case Prioritization Framework using Software Requirements”, IEEE conferenceon Design and Source Code Collaboration(2014)

[40] Rajib mall , “Fundamentals of software engineering” , PHI,(2009)

[41] B.Beizer,software testing Techniques,7th edition (van Nostrand Reinhold, New York,1990)

[42] Ruchika Malhotra, Manju Khari, Heuristic search-based approach for automated test data generation : A Survey, Int. J. Bio-Inspired Computation, 5, 1, (2013)

[43] Mala D. J., Kamalapriya M., Mohan V, Automated Software Test Optimisation Framework an Artificial Bee Colony Optimization-Based Approach, IET Software4, 334(2010)

[44] Tracy Hall, Sarah Beecham, David Bowes, David Gray, and Steve Counsell, "A Systematic Literature Review on Fault Prediction Performance in Software Engineering" IEEE Transaction on Software engineering, Vol. 38, No. 6 (2012)

[45] Arvinder Kaur, M.S. Inderpeet Kaur, "An empirical evaluation of classification algorithms for fault prediction in open source projects", Journal of King Saud University – Computer and Information Sciences(2016)

[46] Julien Delange , John Hudak, William Nichols, James McHale, Min-Young Nam, "Evaluating and Mitigating the Impact of Complexity in Software Models", Software Engineering Institute, Carnegie Mellon University (2015)

[47] Chin-Yu Huang, Chung- Sheng Chen, Chin-En Lai, "Evaluation and analysis of incorporating Fuzzy Expert System approach into test suite reduction", Information and Software Technology, Vol 79, pp 79-105, (2016)

[48] Jianfeng Yang a , Yu Liu b , Min Xie b and Ming Zhao c, "Modeling and analysis of reliability of multi-release open source software incorporating both fault detection and correction processes", The Journal of system and software , vol 115 ,pp 102-110 (2016)

[49] Thelma Elita Colanzia, Silvia Regina Vergilioa, Wesley Klewerton Guez Assunc and Aurora Pozo, "Search Based Software Engineering: Review and analysis of the field in Brazil", The Journal of Systems and Software 86, 970– 984, (2013)

[50] Vahid Garousi, Kai Petersen, and Baris Ozkan, "Challenges and best practices in industry-academia collaborations in software engineering: A systematic literature review" , Information and Software Technology 79 106–127, (2016)

[51] Santosh S. Rathore and Sandeep Kumar, "An empirical study of some software fault prediction techniques for the number of faults prediction", Soft Comput (2016).