

Genetic Algorithm for Test Suite Optimization: An Experimental Investigation of Different Selection Methods

Chetan J. Shingadiya^a, Dr. Nitesh M. Sureja^b

^aPhd Scholar – Rai University-Ahmedabad

^b Professor-GTU-Ahmedabad

^aChetan.shingadiya20@gmail.com, ^bnmsureja@gmail.com

Article History: Received: 10 November 2020; Revised 12 January 2021 Accepted: 27 January 2021; Published online: 5 April 2021

Abstract: Software Testing is an important aspect of the real time software development process. Software testing always assures the quality of software product. As associated with software testing, there are few very important issues where there is a need to pay attention on it in the process of software development test. These issues are generation of effective test case and test suite as well as optimization of test case and suite while doing testing of software product. The important issue is that testing time of the test case and test suite. It is very much important that after development of software product effective testing should be performed. So to overcome these issues of optimization, we have proposed new approach for test suite optimization using genetic algorithm (GA). Genetic algorithm is evolutionary in nature so it is often used for optimization of problem by researcher. In this paper, our aim is to study various selections methods like tournament selection, rank selection and roulette wheel selection and then we apply this genetic algorithm (GA) on various programs which will generate optimized test suite with parameters like fitness value of test case, test suite and take minimum amount of time for execution after certain preset generation. In this paper our main objectives as per the experimental investigation, we show that tournament selection works very fine as compared to other methods with respect fitness selection of test case and test suites, testing time of test case and test suites as well as number of requirements.

Keywords: Genetic Algorithm, Test Suite Optimization, Selection

1. Introduction

Genetic Algorithm is evolutionary in nature which was proposed by John Holland [31]. From past many years, GA is used for optimization of problem in various fields like blueprint detection, robotics, artificial intelligence and many more. In order to use genetic algorithm initially it required to present the chromosome solutions. To evaluate these chromosomes as solutions we have to define fitness function for the same [31]. This genetic algorithm uses idea of population and generation to evaluate through inspired in nature. As far as genetic algorithm is concerned the population contains number of chromosomes which is used as a individual solutions. For the next generation of populations various operators are used and these operators like selection method, crossover methods and mutation methods. As per the Darwin principles of continued existence which is called “the continued existence of the fittest”, the good chromosomes are selected while the other one die away. These good chromosomes regenerate new chromosomes with fine genes which form new generation more feasible in nature. In the same way genetic algorithm have the process facsimile as an essential genetic operation for the algorithms. Presently, there are many selection techniques like tournament, roulette wheel, and tournament etc which most important for reproduction of new population. In this paper, our aim is to study various selections methods like tournament selection, rank selection and roulette wheel selection and then we apply this genetic algorithm (GA) on various programs which will generate optimized test suite with parameters like fitness value of test case, test suite and take minimum amount of time for execution after certain preset generation. [32] Simple steps of genetic algorithm is given below.

Basic Genetic Algorithm

Step1. Representation of given problem using Encoding

Step2. Generate Random Initial Populations

Step3. Loop

Define fitness function for chromosomes in a population

Generate a new population by using GA operators

Calculate fitness value of every chromosome, and then for generation of new population use selection method which is used to select best parents from the population.

Apply crossover on best selected parents for new offspring

Generate new population by apply mutation method on offspring

Step4. When the process complete we will be able to find a solution from output of genetic algorithm.

As per genetic algorithm (GA), chromosome is selected base on fitness value. Good the fitness value of chromosome have the better chance of selection as per probability. Exclusive of any kind of revolutionize preferred chromosome is simulated into next generation of the population. On the other hand other chromosome can select for the reason that GA's probability. For the selection of chromosome various methods are used like tournament selection, roulette wheel and rank selection many more. These methods rebelliously influence the presentation of the algorithm. Crossover method is executed on best selected parents. For the use of crossover, two chromosomes are selected and then crossover point is selected arbitrarily and pair of selected chromosomes go through the process of this method. It is important that chance of crossover is high. Two new offspring are generated by this process. The mutation method initiate with chromosome chance from the population then mutation is executed at random based on mutation method. It is very essential that chance of mutation is need to set low. [32]

2. Literature Survey

This literature investigation is providing information about the various operators of the genetic algorithm like selection methods, crossover methods and mutation methods. As per study also it is clearly review that these genetic algorithm operators are very much important for test case and test suite optimization. This current investigation is presented on test suite optimization problem using genetic algorithm which is a crucial challenges for test suite optimization.

Genetic Algorithm (GA) is used for optimization problems where population evolution is looking for a promising solution. [4] Tournament and ranking selection method is work well in terms of maintaining steady pressure toward divergence where tournament selection method is more suitable for more efficient with respect to time. [46] In this paper researcher Studied the rank based selection and tournament selection method. In which it investigated that tournament selection method is performing good than rank based selection method. [48] The quality of the last best algorithm solution can be further improved by an efficient and unfair search. [11] As per review of this paper after investigation it shows that comparison of roulette wheel selection method and tournament selection method they prove that tournament selection is good method for performance in terms of convergence rate. [49]

To select optimized set of test cases with ability to get all the faults in lesser time this takes more time to run entire test suite. This paper introduces novel test case reduction combine technique based on genetic algorithm and bee ant colony optimization. They have given new test case selection approach from a large test suite using hybrid technique based on genetic algorithms and bee colony optimization. [20] By dropping no of test cases and test suite we can considerably decrease testing efforts and time. [1] Testing of large difficult system might be specially time tough within the application and more complex to test each combination. [18] The decision making for what to test or reduce tests then it causes to miss many things. [18] Very high expectation in cost, time. [6, 19] The optimized test suite generated is much more helpful in regression testing which otherwise carried out with a superior number of test cases. [4] In testing time allocation is very limited which normally 60% of time for

development phase while time for testing is less than 40%. [20] Running entire test suite for entire software products takes nearly seven weeks. [1] For every change, new test cases are written and are added to the existing test suite grows and becomes less efficient because of broken, redundant, obsolete test case. It's become more costly and time tough to run entire test suite. This requires optimization of test suite to takeover with time and cost constraints. [2] Most of the existing methods are static and single objective therefore, selected test suite may be skipped many important test case and test suite is not optimized, suitable, safe for use. [2] The potential work is to estimation the drop in every cost and compares the outcome with other test suite reduction techniques. [1] Partial accessibility of resources and Redundancy of test cases. Ineffective and incorrect test checking. [6] The important things about optimization of the test suite are that get better quality software product which requires very lees amount testing and efforts. It is an expensive activity and takes a lot of time and effort. Genetic algorithm is used to generate effective test case and test suite which highly important for the testing of software product. Genetic algorithm (GA) is a nature inspired algorithm and it is evolutionary in nature. The purpose of use this algorithm is in optimization and minimization of various problems. Optimization is a process that finds the best solution for a problem. Optimization is the process of making something better. In any process, we have a set of inputs and a set of outputs. Optimization refers to finding the input values so that we get the "best" output values. The definition of "best" varies from problem to problem. The goal of optimization is to find that point or set of points in the search space. [26] Optimization is needed because of the large test size suites, limited budgets and resource constraints; in that running a whole set of tests is impossible even for a small one modification. [2] In the creation of a test suite, the variable input domain is the search space and its purpose is to generate test data for branch coverage criteria using evolutionary testing techniques to save computational resources. Global search techniques serve two competing goals such as exploration and exploitation. Exploration is essential to ensure that each part of the solution domain is searchable enough to provide a reliable estimate of the global optimal. Exploitation is also important to focus the search effort around the best solution found so far by searching their environment to reach a better solution. [4] Test suite optimization is the process of effectively generation of test cases and test suite while doing testing of computer software. In this paper we have proposed improved genetic algorithm (GA) with parameters like tournament selection, multipoint crossover method and bit flipped mutation method for test suite optimization.

A. Tournament Selection

In GA, tournament selection method is used for select fittest individuals from the generation. These fittest individuals are forward to the next generation. In an N-way tournament selection, we select n individuals and execute a tournament among all individuals. Simply fittest individual is selected from selected individuals and pass to the next generation. This is the way other tournaments are performed and we get ultimate selection of individual who goes to the next generation. One of the important things in tournament selection is selection pressure which preferential best individuals. Higher the selection pressure more good individuals are preferential. With these GA will get better population in upcoming generation also convergence of a genetic algorithm is achieved by selection pressure. In case of low pressure then convergence rate is also low and similarly is high pressure then convergence is high. Tournament selection is affording selection pressure between individuals. The winner of the tournament is then passing to the next generation. [52]

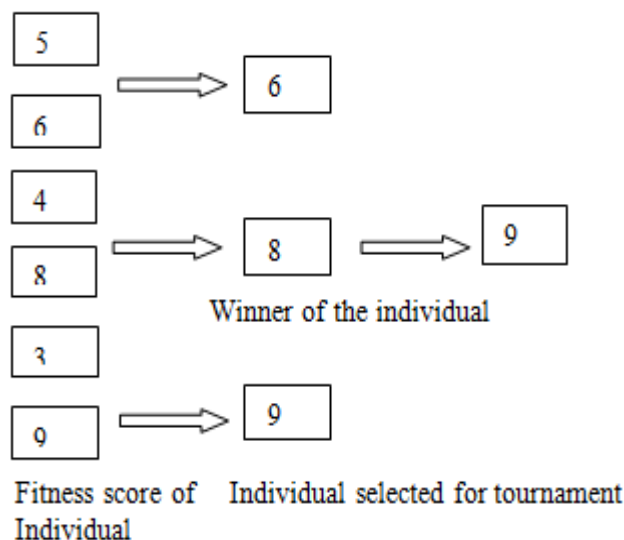


Figure 1: Tournament Selction

Roulette Wheel

The roulette wheel selection method usually use reproduction machinist. In which individuals are selected based a chance to the proportion to the fitness. The chance of selection of section in a roulette wheel is in amount to the quantity of the essential angle of the segment. In GA entire population is partition into wheel and each section represents a chromosome.

The individual is forward to the next generation when proportion of individuals fitness to the total fitness values of an entire population. So for that *i*th individual in the population is selected with chance relative to F_i where F_i is the fitness value for every individual while keeping population size is fixed. Then the chance of *i*th individual is selected is [32, 31]

$$P_i = F_i / \sum_{j=1}^n F_j \tag{1}$$

Where ‘n’ is the population size.

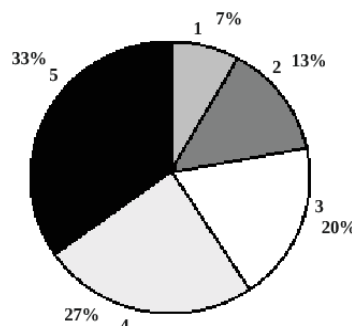


Figure 2: Roulette Wheel Selection

Rank Selection

As per the study show that roulette wheel has some issue when fitness value of individual varies quite large. For sake of simplicity consider given below figure, if best individual fitness is 75% , its occupies 75% of that circumference of the wheel so we can say that other individual have less chance to get selected for next generation. So to overcome this issue rank selection method is useful. While in rank selection, ranks are given to every individual and then taken each individual, receives fitness from ranking. According to the rank selection method the worst individual will be having rank 1, then next 2 ...and so on. So it is obvious that good individual will have N fitness. This approach may be slow down the performance of genetic algorithm because fit individual may not differ from other individual. [31]

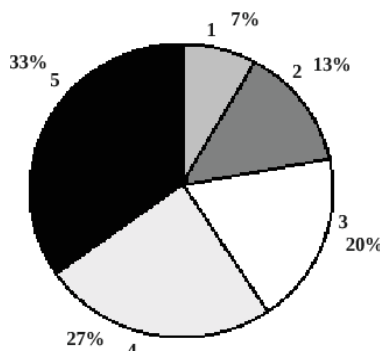


Figure 3: Rank Selection

3. Methodology

We are using genetic algorithm (GA) with its parameters like initial population, fitness function/objective function, selection methods, crossover methods and mutation method for test suite optimization. This algorithm is

used to calculate fitness value for the each chromosome which is defined in our problem. Based on these fitness values we will also calculate total fitness value of the test suite as per the problem. After calculating fitness value and testing time for the each test suite.

Initial Population

The procedure starts with a set of individuals which is called a Population. Each individual is a result. An individual is recognized by a set of parameters known as Genes. Genes are joined into a string to form a Chromosome. Using genetic algorithm these set of genes of an individual is identified using a string in terms binary values are used. For sake simplicity we dictate here for generation of the initial population for 4-input variables student scale program.

Fitness Function

In genetic algorithm (GA), fitness function or objective is used to find how fit the particular chromosome. Fitness provides fitness value to each chromosome. The chance of that an individual or chromosome will be selected for reproduction is depends on fitness value. Our proposed fitness function or objective function for the genetic algorithm (GA) is given below.

In our proposed approach, the fitness function is used to calculate fitness value based on summation of all bits each bit string value in the given chromosome. The fitness function or objective function is given below.

$$F(X1) = \sum_{i=1}^n (Tcri) \quad (2)$$

Where, Tcri is the numbers of requirements are available in the test cases (Chromosome) or individual and n is gene length. The fitness function for calculation of test suites fitness is given below.

$$F(X2) = \sum_{i=1}^n (Tci) \quad (3)$$

Where, Tci is the numbers of test case are available in the test suites (population) and here n is the number of test cases (population). The fitness function for the calculation of testing time of test suite is given below.

$$F(X3) = EETS(TS) - SETTS(TS) \quad (4)$$

Where, EETS = End Execution Time for Test Suite, SETTS = Start Execution Time for Test Suite TS stand for Test Suite. Above all three functions like F(X1), F(X2) and F(X3) are the fitness function or objective functions. In which F(X1) will be calculated fitness value of the chromosomes (Test cases) and F(X2) will be calculated fitness value of the test suites and F(X3) will be calculating testing time for the each test suite which are defined in the given problem.

Selection Method

In genetic algorithm (GA) so many selection methods are available for selecting best parents from available test suite. Therefore, selection is the most important method for optimization of any problem. For reproduction or new population we will be using tournament selection technique.

Crossover Method

Similarly like selection method, there many crossover methods are available in genetic algorithm (GA). So Crossover method is the important operator in a genetic algorithm. For each pair of parents to be mated for that a crossover point is chosen at random from within the genes. In our approach we will examined various crossover methods like single point and multi point. We have used multi crossover methods in our proposed genetic algorithm (GA).

Mutation Method

Mutation is also important operator to form new offspring formation, some of their genes can be subjected to mutate with a low down casual probability. Here we have used bit flipped mutation method that implies that bits in the bit string in the chromosome can be flipped using with mutation.

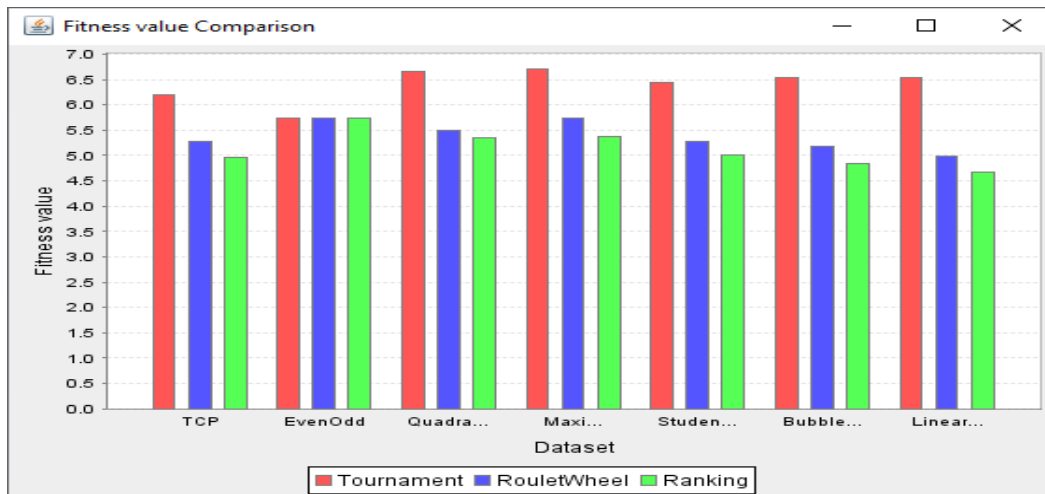
4. Results and discussion

As per the experimental investigation, we have taken gene length is 10, number of populations are 20 and this process continue till 100th generation. For this we have examined our proposed approach on various programs

likes two input variable problem it is odd or even example and three input variable problems it is triangle classifier, quadratic equation and maximum of three and four input variable it is student scale program and five input variable problem it is bubble sort and six input variable problem it is liner equation.

For test suite optimization using genetic algorithm (GA), we will be examined total twenty test suites. As per our problem definition these twenty test suites mean populations in which each test suite contain four test cases. For the comparative analysis of fitness value we have used three selection techniques like tournament selection, roulette wheel selection and rank selection. Results are clearly shows that fitness value of each individual in tournament selection is far better than other two techniques similarly; test sites fitness value is also very fine in tournament selection then other two methods.

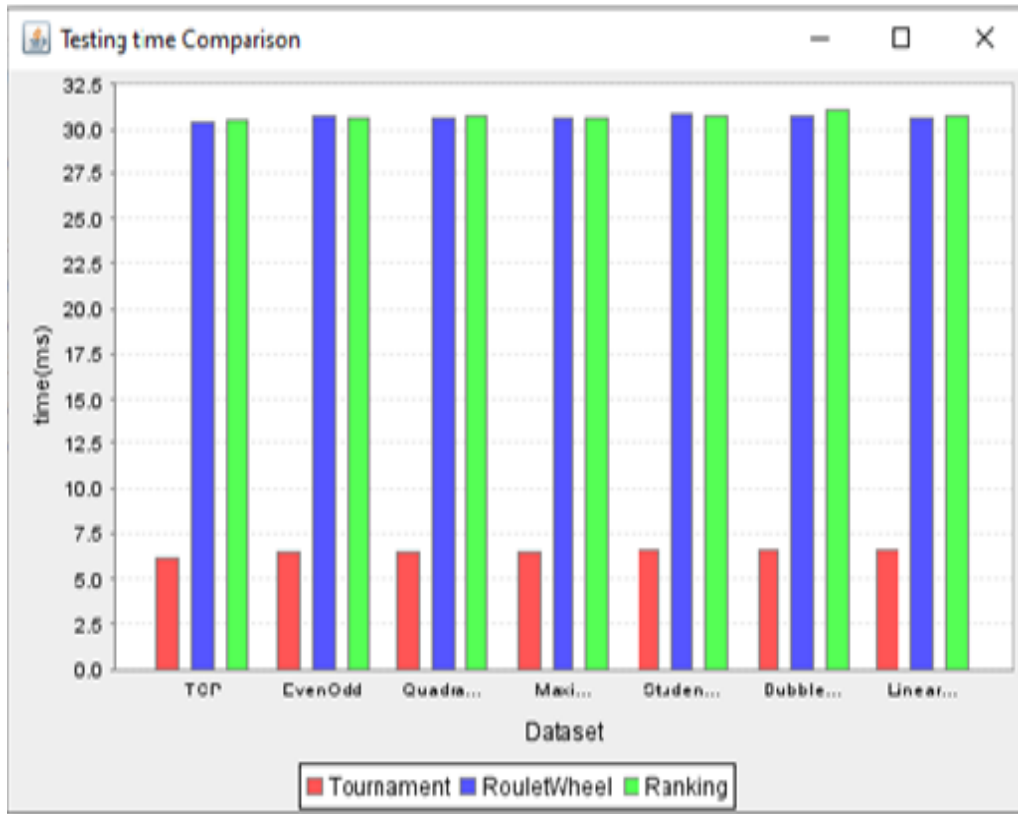
In our experiment we use multi point crossover method because it gives better output of individuals. Finally, at last we have examined testing time of test suites with different selection techniques. As per the standard problem result tournament selection method give far better values of testing time of test suites than other two methods like rank selection method and roulette wheel selection method. For simplicity table result are shown only for student scale program. Here RWFV stands for Roulette wheel selection fitness value, RFV stands for Rank selection fitness value and TFV stands for Tournament selection fitness value. Similarly RWTT stands for Roulette wheel testing time for test suite, RTT stands for Rank selection testing for test suite and TTT stands for Tournament selection testing for test suite.



Graph 1: Comparison of fitness value

Program – Student Scale		
RWFV	RFV	TFV
3	6	6
4	3	7
8	5	8
7	6	7
4	5	5
4	4	5
6	7	7
5	6	6
7	5	7
7	8	8
7	7	7
5	6	6
4	4	6
6	2	6
4	5	5
5	5	8
7	6	7
6	6	6
7	5	7
5	4	5

Table 1: Comparison of fitness value



Graph 2: Comparison of testing time of selected fitness

Program – Student Scale		
RWTT	RTT	TTT
31	31	8.7
32	31	7.3
32	30	6.9
31	30	6.8
31	32	6.7
30	30	6.6
31	31	6.5
30	31	6.6
30	31	6.6
30	31	6.6
31	31	6.6
31	31	6.6
31	31	6.6
31	31	6.6
31	31	6.8
30	30	6.6
31	31	7.1
31	30	6.9
30	33	6.7
30	31	6.5
30	30	6.6

Table 2: Comparison of testing time of selected fitness

Student Scale Program	
Population P1	
TSF	TST
21	8
18	7
22	6
19	7
15	7
16	5
22	6
21	6
25	7
24	7
22	7
21	7
17	7
18	6
15	6
21	6
21	6
21	6
20	7
18	7

Table 3: Comparison of Test Suite fitness value and testing time

5. Conclusion

We have carried out experimental investigation on various standard programs like even or odd program, maximum of three number, triangle classifier problem, quadratic equation, student scale program, and bubble sort program and linear search program. We conclude that the result with this approach of genetic algorithm with tournament selection method works very fine. As we know that crossover operator and mutation method are the most important parameter. Finally we conclude that genetic algorithm (GA) works fine with tournament selection with operators like multi point crossover operator and bit flipped mutation method. As per the experimental investigation after 100 generation, 20 population and each chromosome have gene length is 10, we conclude that

1. While Testing of test suite, time taken by tournament selection method very less as compared to other methods.
2. The fitness value of test suites is also optimized.
3. As the number of generation grows more it is clearly shown that more number of requirements is covered in test suite with our proposed approach in genetic algorithm.

References

- ShrividyaJayaprakash, Dr. K. Algarsamy , A Distinctive Genetic algorithm for Test-Suite Optimization. ELISVER, The 2015 International Conference on Soft Computing and Software Engineering (SCSE 2015). 427-434. <https://doi.org/10.1016/j.procs.2015.08.437>. 2015
- Zeesan Anwar, Ali Ahsan , Multi-objective regression test suite optimization with Fuzzy logic. IEEE, International Multi topic Conference (INMIC). 95-100 DOI: 10.1109/INMIC.2013.6731331. 2013
- Pavneet Singh Kochhar; FerdianThung, David Lo , Code coverage and test suite effectiveness: Empirical study with real bugs in large systems. IEEE, International Conference on Software Analysis, Evolution and Reengineering (SANER). 560-564 DOI: 10.1109/SANER.2015.7081877. 2015
- M. S. GeethaDevasena, G. Gopu, M. L. Valarmathi, Automated and Optimized Software Test Suite Generation Technique for Structural Testing. World Scientific, International Journal of Software Engineering and Knowledge Engineering. Vol-26, No:01, 1-13. <https://doi.org/10.1142/S0218194016500017>. 2016
- ManjuKhari, Prabhat Kumar, Denial Burgos, Ruben Gonzalez Crespo, Optimized test suites for automated testing using different optimization techniques.Springer. 8341-8352. DOI: 10.1007/s00500-017-2780-7. 2017
- Akshat Sharma, RishonPatani, AshishAggrawal, Software Testing using Genetic Algorithms.International Journal of Computer Science and Engineering Survey.Vol-7. No.2. 21-33 DOI: 10.5121/ijcses.2016.7203. 2016
- GeethaDevsena M. S., Lavarmathi M. L, Optimized test suite generation using tabu search technique. International Journal of Computational Intelligence Techniques.Vol.1. Issue.2. 10-14. 2010

- AllesandroMarcheto, Mahfizul Islam, Anjalo Susi, GuiseppeScanniello, A Multi Objective for Test Suite Reduction. ICSEA 2013, The Eighth International Conference on Software Engineering Advance. 18-24. 2013
- Gordan Fraser, Andrea Arcuri, Whole Test Suite Generation. IEEE, Vol. 39. Iss. 2. 276-291. DOI: 10.1109/TSE.2013.6. 2013
- Gordan Fraser, Andrea Arcuri, Phil McMinn, A Memetic Algorithm for Whole Test suite Generation, ELISVER, The Journal of System Software, 1-17. <https://doi.org/10.1016/j.jss.2014.05.032>. 2014
- S Prayudani¹, A Hizriadi², E B Nababan² and S Suwilo , Traveling Salesman Problem (TSP), ICCAI 2019 Journal of Physics: Conference Series, 1-8, doi:10.1088/1742-6596/1566/1/012131. 2019
- Soma SekharaBabu Lam, M L Hari Prasad Raju, UdayKiran M, Swaraj Ch, PravinRanjanSrivastav, Automated Generation of Independent Paths and Test Suite Optimization Using Artificial Bee Colony, International Conference on Communication Technology and System Design-2011, 1877-7058, 191-200. DOI: 10.1016/j.proeng.2012.01.851. 2011
- Ankita A Mundade, T M Pattewar, Test Suite Generation using Memetic Algorithm DOI: 10.1109/ICCSP.2015.7322564. 2015
- FlormPopentiu-Vladicescu, GrigoreAlbeanu, Nature Inspired Approaches in Software faults identification and debugging”, ELESVIER, 2nd International Conference on Intelligence Computing, Communication & Convergence, 6-12. <https://doi.org/10.1016/j.procs.2016.07.315>. 2016
- G. Manikandan, A Comparative Analysis of Various Mutation Types for Achieving Privacy in Medical Data Mining. Research Journal of Pharmacy Technology. 2451-2455. 2017
- ManjuKhari, Prabhat Kumar, An Effective Meta-Heuristic Cuckoo Search Algorithm for Test Suite Optimization”, Informatica 41. 363-377. 2017
- Wang Lijuan, ZhaiYue, HouHongteng, Genetic Algorithms and its Application in Software Test Data Generation, IEEE, 617-620. DOI: 10.1109/ICCSEE.2012.36
- Muhammad AbidJamil, Muhammad Arfi, Normi Sham AwangAbubakar, Akhlaq Ahmed (2016), Software Testing Techniques: A Literature Survey, IEEE, 177-182. DOI: 10.1109/ICT4M.2016.045. 2012
- Jyoti, KamnaSolanki, A Comparative Study of Five Regression Testing Techniques: A Survey, International Journal of Scientific & Technology Research. Vol. 3.Iss. 8. 76-80. DOI: 10.1109/ASWEC.1997.623769. 2014
- BhartiSuri, IshaMangal, VarunSrivastava, Regression Test Suite Reduction using an Hybrid Technique Based on BCO and Genetic Algorithm. International Journal of Computer Science & Informatics.vol. 2.Iss. 1. 165-172. DOI: 10.47893/IJCSI.2013.1113. 2017
- MohitSaifi, Shahid Sagar, Application of Software Testing using Genetic Algorithm International Journal of Engineering Research & Management Technology, Vol. 2, Iss. 3, 231-236. DOI: 10.5121/ijcses.2016.7203. 2015
- Vaibhav E. Pawar, Amol K. Kadam, Dr. S D Joshi, Analysis of Software Reliability using Testing Time and Testing Coverage. International Journal of Advance Research Computer Science and Management. Vol. 3.Iss. 5. 143-148. 2015
- Gan Kim Soon, Tan Tse Guan, Chin Kim On, Rayner Alfred, Patricia Anthony, A comparison on the performance of crossover techniques on video games.IEEE International Conference on Control System, Computing and Engineering (ICCSCE). 493-497. 2013
- Saunya Dixit, PradeepTomar, Applying Computational Intelligence in Software Testing. Journal of Artificial Intelligence Research & Advances.Vol. 2.Iss. 2. 1-5. 2015
- Manoj Kumar, Aruna Sharma, Rajesh Kumar, Optimization of Test Cases using Soft Computing Techniques: A Critical Review. WSEAS Transactions on Information Science and Applications.Vol. 8.Iss. 11. 440-452. 2011
- Ahmad Hasanat, V B Surya Prashath, Choosing Mutation and Crossover ratios for Genetic Algorithm – A Review with new dynamic Approach. Information Journal MDPI. 1-36. 2019
- R. Uma Maheswari, and D. Jeya Mala, Combined Genetic and Simulated Annealing Approach for Test Case Prioritization. International Journal of Science and Technology.Vol. 8. 1-6. DOI: 10.17485/ijst/2015/v8i35/81102. 2015
- VinayHarsora, Dr. Apurva Shah (2011), A Modified Genetic Algorithm for Process Scheduling in Distributed System. Artificial Intelligence Techniques – Novel approaches & Practical Application” 36-40. https://www.tutorialspoint.com/software_testing_dictionary/test_suite.htm
<http://softwaretestingbooks.com/objectives-of-software-testing>
- Neural Network, Fuzzy Logic and Genetic Algorithms – Synthesis and Applications
- AnupriyaShukla, Hari Mohan Pandey, DeepriMahrotra, Comparative Review of Selection Techniques in Genetic Algorithm. IEEE 1st International Conference on Futuristic trend in Computational Analysis and Knowledge Management.515-519. DOI:10.1109/ABLAZE.2015.7154916. 2015

- Dr. Preethi Harris, Sathya A., ShanmugaPriya S, A Novel Test Cost Reduction Approach for Test Suites. International Journal of Computer Science and Mobile Computing. Vol 4. Issue 4. 189-194. DOI: 10.1145/1363686.1363782. 2015
- Kulvinder Sing, Rakesh Kumar, Optimization of Functional Testing using Genetic Algorithms. International Journal of Innovation, Management and Technology. Vol 1. No 1. 43-46. 2010
- Ahlam S A Ansari, Prof. Kailash K. Devadkar, Dr. PrachiGharpure, Optimization of Test Suit- Test Case in Regression Test. IEEE, International Conference on Computational Intelligence and Computing Research. 1-4 DOI: 10.1109/ICCIC.2013.6724206. 2013
- DebasisMGeneticohapatra, PrachetBuyan, Durga P. Mohapatra, Automated Test Case Generation and Its Optimization in Path Testing using Genetic Algorithm and Sampling. WASE, International Conference on Information Engineering. 643-646. DOI: 10.1109/ICIE.2009.22. 2009
- Subham Kothari, Anand Rajawat, Minimize the size of test suite using genetic algorithm for object oriented program. IEEE, International Conference on ICT in Business Industry & Government (ICTBIG). 1-5. DOI: 10.1109/ICTBIG.2016.7892703. 2016
- Yunqi Du, YA Pan, HaiyangAo, O. ALEX, Yong Fan, Automatic Test Case Generation and Optimization Based on Mutation Testing. IEEE International Conference on Software Quality, Reliability and Security Companion (QRS-C). 522-523. DOI: 10.1109/QRS-C.2019.00105. 2019
- Dharmlingum JAYA MALA, Elizabeth Ruby, VAsudev Mohan, A Hybrid Test Optimization Framework Coupling Genetic Algorithm with local search technique. Computing and Informatics. Vol 29. 133-164. 2010
- Sara Amjad, KuldeepJaiswal, Survey of Optimization Techniques for Test Suite Optimization in Regression Testing. IJIRCCE. Vol 2. Issue 4. 1981-1983. DOI: 10.1016/j.procs.2016.03.020. 2016
- Gan Kim Soon, Tan Tse Guan, Chin Kim On, Rayner Alfred, Patricia Anthony, A comparison on the performance of crossover techniques on video games. IEEE International Conference on Control System, Computing and Engineering (ICCSCE). 493-497. DOI: 10.1109/ICCSCE.2013.6720015. 2013
- Hongian Jang, Suzan Jang, Kevin Hapesi, A Review of Nature Inspired Algorithms. Journal of Bionic Engineering. 232-237 [https://doi.org/10.1016/S1672-6529\(09\)60240-7](https://doi.org/10.1016/S1672-6529(09)60240-7). 2010
- Wei Fang, Xiodong Li, Mangjie Zhang, Mengquihu ,Nature Inspired Algorithms for Real World Optimization Problem. Hindawi, Journal of Applied Mathematics. 1-2 DOI: 10.1155/2015/359203 2015
- Xin She Yang, Nature Inspired Algorithms: Challenges and Open Problems, Journal of Computational Science. 1-27. <https://doi.org/10.1016/j.jocs.2020.101104>. 2020
- Noirani Mohd Rajli, John Geraghty, Genetic Algorithm Performance with different selection techniques in solving TSP. WCE, Proceeding of World Congress Conference. Vol 2. 2011
- AnitKumar , Encoding System in Genetic Algorithm. International Journal of Advance Research in IT and Engineering (2013). Vol2. No.3. 1-7. 2013
- Ronald Pinho, FillipeSaraiva x A Comparison of Crossover Operators in Genetic Algorithms for Switch Allocation Problem in Power Systems. IEEE, Congress on Evolutionary Computation (CEC - 2020). 1-8. DOI: 10.1109/CEC48606.2020.9185795 (2020) .