Automated test data generation using soft computing techniques

Deepa Chauhan, Akanksha Sehgal

Abstract— **one of the most momentous tasks during software testing is the generation of proper test data. It can also be done by manually and automation process. There are various soft computing Techniques that have been used to automate this task. Here we describe the automatic generation of test data especially for web-based application with the help of Soft computing techniques. These techniques are also called the metaheuristic techniques. The metaheuristic techniques are high -level soft computing strategies that define algorithm framework and technique to find approximate solution for optimization problems. Main purpose is to generating test data automatically with the help of metaheuristic search technique. For this, there are a number of tools available that can be used for automation of test data. But the researchers found, that we can also used some soft computing techniques like genetic algorithm (GA), particle swarm optimization(PSO), simulated annealing, evolutionary algorithm, ant colony optimization, fuzzy logic and so on to generate smart and efficient test data automatically with the help of these soft computing techniques. The existing result have the hybrid Genetic particle swarm technique algorithm(GA-PSO), Hybrid Genetic algorithm and Hill climbing techniques, genetic algorithm(GA) and ant colony algorithm(ACO) and so on. But the combine approach is giving the better result than individual. So we are taking a different combination of soft computing techniques that is GA (genetic algorithm) and fuzzy logic. And then we will compare this combine technique with the earlier proposed hybrid GA-ACO technique.**

Index Terms—**software testing, automated test data generation, metaheuristic search techniques, Genetic algorithms, ant colony**

I. INTRODUCTION

In software testing there are two types of testing, one are manual testing and the other one is automation testing. The automated software testing is a process in which software tools execute pre-scripted tests on a software application before it is released into production .In industry, test data selection is generally a manual process - the responsibility for which usually falls on the tester. However this practice is very costly, difficult and challenging. Automation in this area has been limited. Soft computing techniques belong to the family

Of evolutionary algorithms. Their goal is to obtain an approximate solution to an optimization problem, when there Is no exact method (or the solution is unknown) to solve it in a Reasonable time. Genetic algorithms use the concept of natural selection and apply it to a population of solutions to the given problem. Software testing forms an integral and necessary part of software development life cycle. Software development process invests at least 50% of the total cost in software testing process. Now, if the safety is an integral feature of the software the cost can go even higher. To improve the quality of software, extensive manual testing is useless due to the huge requirement of time and cost. At present, there are lots of testing tools available in the market in which we can observe automation of test scripts execution implemented with the capture and playback mode feature of the tools. This again depends on the human intelligence and his involvement and hence does not perform software testing automatic completely. The selection of test cases is decided by Software tester and success of testing process depends upon his expertise and intelligence of a tester. Software testing can be effectively automated to generate test data with soft computing strategies like genetic algorithm (GA) and fuzzy logic. It works by transfiguring test data generation problem to optimization problem and test goal to objective functions. Hybrid genetic algorithm is better than Genetic Algorithm. Many literatures illustrate that fuzzy logic is more skilful and efficient in local search. In this paper, we introduce genetic algorithm to fuzzy logic and form a mixed algorithm which carries on and improve the advantage of the two combining algorithms, called GAFLA. And comparing with the hybrid GA and ACO

Manuscript Received April 2015.

***Deepa Chauhan**, Department of computer science and engineering, Galgotias University, Greater noida, India, 7376644876.*

***Akanksha sehgal**, Department of computer science and engineering, Galgotias University, Greater noida, India, 9910993964.*

II. OBJECTIVE

The importance of automated web based application testing derives from the increasing assurance on these systems for business, media, social, organizational and governmental functions. Over the past ten years, internet user numbers grew by approximately 400%. In 2009, online retail sales grew by 11% compared to 2.5% for all retail sales. Amazon, the leading online retailer, increased its sales by 29.5%. One of the advantages of web applications is their continual availability. The service provided by a web application is not limited by location or time, since geographically most of the users may have concurrent access. Still, these advantages placed a demand for high availability. Web time is considered to be 5 to 7 times faster than normal time. Web technologies change more frequently and their adopters seek early acquisition of market share. This pressure on development time squeezes the testing phase, especially when it is unautomated, labour intensive and therefore time consuming. However, inadequate testing poses significant risks. Studies showed that trust and convenience are major factors affecting customer loyalty using web applications. Both recent and

ISSN: 2278 – 1323

historical studies have shown that online shoppers exhibit impulsive purchasing habits, indicating that downtime can prove costly. For example, downtime was estimated to cost Amazon $25k per minute even as early as 2001. Search based testing has been used widely as a way to automate test data generation for traditional, stand alone applications, thereby making testing less reliant on slow laborious processes. Search based test data generation has also proved to be effective and complementary to other techniques. However, of 399 research papers on SBST, only one mentions web application testing issues and none applies search based test data generation to automate web application testing. Popular web development languages such as PHP and Python have characteristics that pose a challenge when applying search based techniques such as dynamic typing and identifying the input vector. Moreover, the unique and rich nature of a web application's output can be exploited to aid the test generation process and potentially improve effectiveness and efficiency.

III. SURVEY OF RELATED WORK

Genetic Algorithm has low search efficiency in later period of revolution. In order to overcome such shortcoming of Genetic Algorithm, researchers introduce some algorithms which have better effect on search, such as Annealing algorithm and Ant Colony Optimization [4, 13-16]. As a result, they produce some new algorithms such as Mixed annealing genetic algorithm and Hybrid genetic ant colony algorithm. In field of test, researchers have already used Mixed annealing genetic algorithm and Hybrid genetic ant colony algorithm to test the result of experimental data. Generally speaking, Hybrid genetic ant colony algorithm has a good effect of restraining local convergence and improving the search efficiency. In fact, Hybrid genetic ant colony algorithm itself also has such characteristics, early maturity in local convergence and low effect in searching. Anyhow, Hybrid genetic ant colony algorithm is better than Genetic Algorithm. Ant colony system algorithm is put forward to overcome the shortcoming of ant Colony Optimization. Many literatures illustrate that ant colony system algorithm is more competent and efficient in local search. In this paper, we introduce genetic algorithm to ant colony system algorithm and form a mixed algorithm which carries on and improve the advantage of the two combining algorithms, called ACSGA. In order to test its adaptability, we choose classical triangle discrimination problem used frequently in the experiments of path-oriented software testing to verify the efficiency of ACSGA. GA has started getting competition from other heuristic search techniques, just like the particle swarm optimization (PSO). Like GA, PSO is set with a population of random solutions. The development was based on survey of the social behaviour of animals such as bird flocking and swarm theory. Each one in PSO is assigned with a randomized velocity according to its own and its companions‟ flying experiences, and the individuals, that are called particles, are then flown by hyperspace. On Comparing with GA, PSO has few attractive characteristics. It has memory, so the knowledge of right solutions is retained by all particles. Since in GA, above knowledge of the problem is destroyed once the population changes. It has useful cooperation between particles, the particles in the swarm share information between them [30]. Various works [16]–[20] show that particle swarm optimization is equally well suited or even better than genetic algorithms for solving a number of test problems [21]. The GPSCA is a combination of genetic and particle swarm algorithm to automatic test-data generation for searching test data to satisfy the data-flow coverage criteria with using fitness function that evaluates the fitness of test data based on Its relation, through dominance, to the definition and use in the data-flow requirement. The paper also presents a set of empirical studies that show the effective of our test-data generation technique in achieving coverage (generating test data that cover the test requirements), reducing the number of Test cases. The Genetic Algorithms have been used for several

Optimization problems. The genetic algorithms have been used to generate test plans for functionality testing [3] i.e. is to verify if the software satisfies design and functional suitability criteria. [4] More work has been done using Genetic Algorithms in formal concept analysis to generate branch coverage test data automatically, which supports automatic test data generation. [5] Ant Colony Optimization has been used similarly used for test data generation. The "all state testing coverage" requirement is commonly used in state based software testing. A test suite is said to achieve all states coverage if every state is accessed at least once by a test case within. On the basis of this simple fact Ant Colony Optimization has been used with several variations to cover various problems [2].3 Ant Colony Optimization has been used to solve several other optimization problems like

Travelling Salesman Problem to obtain better results than

IV. OUR PROPOSED METHOD

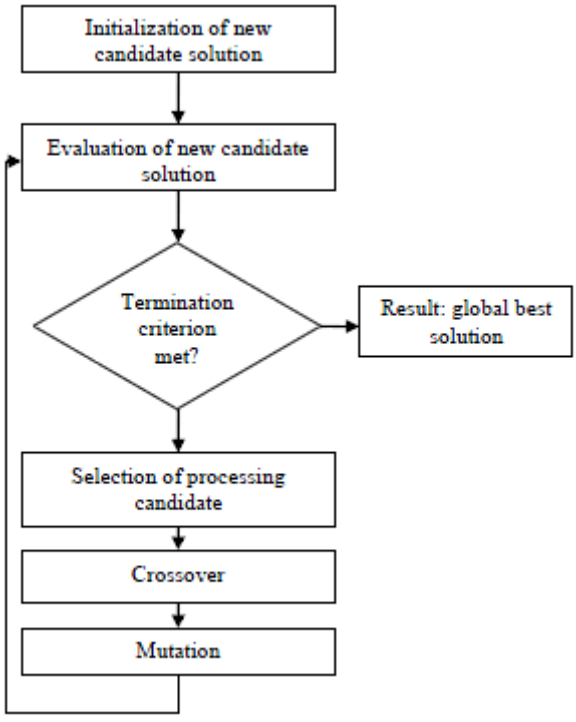
A. Introduction to GA and fuzzy logic

Genetic Algorithm can be used to find out optimized solution. Genetic algorithms work on the basis of „genes‟, which are created randomly. They are then subjected to some task. Genes demonstrating good performance are kept for next phases, while others are discarded. As far as testing is concerned, Genetic algorithm searches for optimal test parameter combinations that satisfy a predefined test criterion. This test criterion is represented through a "coverage function" that measures how many of the automatically generated optimization parameters satisfy the given test criterion. Genes that optimize the coverage function will survive while others will be discarded; the process is repeated with optimized genes being replicated and further random genes replacing of discarded genes. Ultimately one gene (or a small group of genes) will be left in the set and this would logically be the best fit for coverage function. The salient features of our genetic algorithms are as follows: It's a representation of a guess called chromosome. GA is an initial pool of chromosome and also a fitness function. A crossover operator and a mutation operator. The GA uses basically three operators (Reproduction, Crossover and Mutation) to Handel the genetic composition of a population. Reproduction is a procedure by which people rated the current generation are

Reproduced in the new generation. The crossover operates two-off springs by recombination of the information from both parents. The mutation is random charge value of the same of the genes in an individual.

Selection: A selection scheme is applied to determine how individuals are chosen for mating based on their fitness. Fitness can be defined as a capability of an individual to survive and reproduce in an environment. Here the Selection generates the new population from the old population, thus starting a new generation. Each chromosome is evaluated in present generation to determine its fitness value. This fitness value is used to select the better chromosomes from the population for the next generation.
Crossover or Recombination: After selection, the crossover operation is applied to the selected chromosomes. It involves swapping of genes or sequence of bits in the string between Two individuals. This process is repeated with different parent individuals until the next generation has sufficient individuals. After crossover, the mutation operator is applied to a randomly selected subset of the population.
Mutation: Mutation alters chromosomes in small ways to introduce new good traits. It is applied to bring diversity in the population.
Algorithm 1 GA (Genetic Algorithm)
Input: random population and appropriate fitness function
Output: optimized solution
Initialize (population);
Evaluate (population);
While not satisfied **do**
Selection (population);
Crossover (population);
Mutate (population);
End

Flow chart of GA-



The genetic algorithm is a model of machine learning which derives its behaviour from a metaphor of the processes of

Evolution in nature. And it is done by the creation within a machine of a population of individuals represented by chromosomes, in quintessence's a set of character strings that are analogous to the base-4 chromosomes that we see in Our own DNA. Every individual in the population then go through

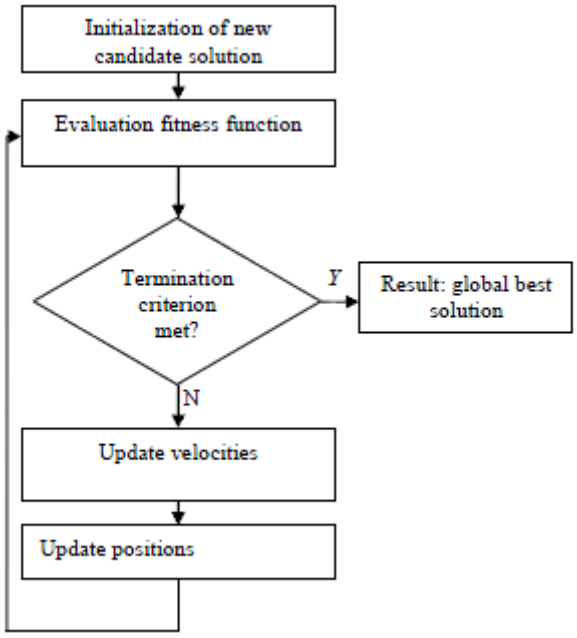
B. Fuzzy logic

Main objective of software industry is to customer satisfaction. So the Bug free software needs to be produced with minimum testing cost and in minimum time. Automatic testing is necessary for adapting fast development of software industry as well as cut down cost and time. Testing including execution of a program by some sets of test data and compare the results with expected outcomes is called software testing. Generating of automated test data is very difficult task in object oriented program because inheritance, method overriding, polymorphism, templates shows many binding anomalies due to dynamic behaviour of objects. Class is the basic building block of object oriented programming. Traditional testing like structural testing, functional specification- based testing and heuristics testing approaches are used for it. Structural testing is essential because it's located the bugs in codes by control flow testing, path coverage testing, data flow testing. Functional testing meets the requirements and specification of software. Heuristics testing technique test the abstract classes. Automatic test data is generated in object oriented programming from traditionally technique. In code- based technique, test data is generated after analyzed the change impact between source code and instrumented code. In model-based technique test data is generated after analysis of two different versions of models. Model is drawn during system requirement analysis phase. Hitesh et al [12] has explained automated test data generation technique for soft ware testing in their paper. Some Researchers explained automated test data generation based on UML designing. But nowadays researcher has concentrated on search based [2] technique for automated test data generation of OOP. They have utilized evolutionary algorithm like genetic algorithm (GA), genetic programming (GP). Some researchers made comments on the using of GA with Fuzzy logic as well as neural networks [10] for various type of software testing. In this paper automated test data is generated on the new model based approach. Advantage of this approach is that [3] test data can be available earlier during software requirement analysis phases. Dynamic behaviour [1] of objects has been utilized throughout the model. Information is extracted from .UML file by java parser. Tree structure of objects is formed by the extracted information. The tree is then converted to optimized tree structures by Genetic Programming in association with Fuzzy logic control. The optimized tree is then converted to binary trees. Test data generation, Validity checking, Termination, all are done from binary trees by using depth first search algorithms. The fuzzy based extension of GA (FAexGA) approach for test case generation. The aim is to find minimal set of test cases that are likely to expose faults using mutated versions of the original program. In FAexGA approach, crossover probability varies according to the age intervals assigned during lifetime. The crossover probability of young and old individuals is assigned low while for other age interval this probability is high. The very young offspring's crossover probability is low thus enabling exploration capability. Old offspring have also less crossover probability and eventually dying out would help avoiding a local optimum or premature convergence. On the other hand, middle-age offspring's are frequently used for crossover operation. Include the age and lifetime of chromosomes (parents). The emphasis of this work is on the exploration and

ISSN: 2278 – 1323

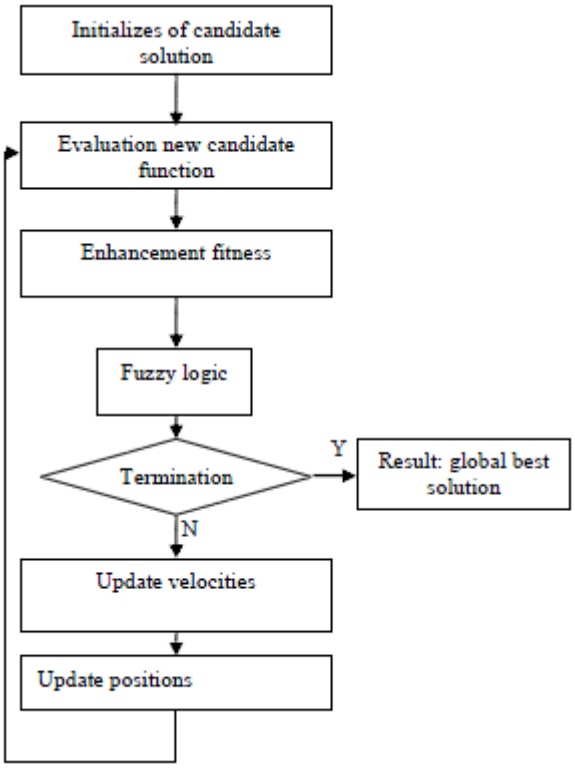
exploitation of individuals. The fuzzification interface of FLC includes variables that determine the age of an offspring. FLC assigns every parent values Young or Middle-age or Old. These values determine the membership for each rule in FLC rule base. The fuzzification interface of FLC defines for each parents the truth value of being Young, middle-age and old as shown in Table 1. The fuzzy rule base used in this experiment is presented in Table1. Each cell defines a single fuzzy rule. Example, "If the Parent 1 is old and Parent 2 is older than crossover probability is Low". The centre of gravity (COG) is used as a defuzzification method which computes crisp value for the crossover probability based on values of the linguistic labels as shown in Table 2.

Flow chart for fuzzy logic



The test cases relate to the inputs of tested software and are represented as a vector of binary or continuous values. The test cases are generated randomly in the search space of possible input values. Genetic operators are applied and the test cases are evaluated based on the fault – exposing - capability using mutated versions of original program Fuzzy Logic is a form of approximate reasoning, which can be used to elected variation or approximation in logic, by making use of natural language (NL) in logic. It is a software testing technique, semi or fully automated, that demand for providing invalid, unanticipated, or random data to the inputs of a computer program. The simplest form of fuzzing technique is sending a stream of random bits to the software, such as command line options and randomly mutated protocol packets, or such as events. In order to remain a realistic approach for the broad field of applications, fuzzy logic solutions need to adjust to the requirements of modern software. Fuzzy logic is a complex topic which requires proper software based solution to achieve efficient usage in real life use cases. In some cases, hiding this complexity is the key to success. In other cases by embracing its possibilities and advantages modern software engineering problems can be conquered elegantly and efficiently. Fuzzy logic can be employed in regression and other types of testing providing better and reliable results. Fuzzy logic is used to develop expert systems which can select feasible test cases retaining the efficiency and effectiveness and also helps to reduce the number of test cases, saving development cost, effort and time.

Proposed approach



V. CONCLUSION AND FUTURE WORK

Developed an algorithm for generating test data using combining the power of GA and Fuzzy logic, GAFLA– Genetic Fuzzy algorithm Combined with a new multiobjective fitness function. Demonstration has been bringing out to show the effectiveness of the proposed GP-FLA compared to the GA-ACO techniques. The results of our new approach GAFLA is better than GA-ACO as in some cases, it has higher coverage ratio % than the GA-ACO. Our experiment also demonstrated the effectiveness of our proposed approach in case of number of generations, as GA-FLA require less generation than GA-ACO.

The future work will be to generate the test data using hybrid Fuzzy logic and ACO (Ant colony optimization) and compare its effectiveness with our GA-FLA approach for web based application.

REFERENCES

[1] Dr. Velur Rajappa, Arun Biradar, Satanik Panda "Efficient software Test case generation Using Genetic algorithm based Graph theory" International conference on emerging trends in Engineering and Technology, pp. 298-303, IEEE (2008)

[2] Bipin Pandey, Rituraj Jain, "Soft Computing Based Approaches for Software Testing: A Survey"International Journal of Soft

Computing and Engineering (IJSCE), ISSN: 2231-2307, May 2011.

[3] Pradeep Kumar and Yogesh Singh (2012) "Assessment of software testing time using soft computing techniques" published in ACMSIGSOFT Software Engineering Notes, January Issue 2012. DOI: 10.1145/2088883.2088895."

[4] Nirmal Kumar Gupta, Mukesh Kumar Rohil "Improving GA based Automated Test Data Generation Technique for Object Oriented Software" 3rd IEEE International Advance Computing Conference (IACC) (2013).

[5] Phil McMinn, "Search-based Software Test Data Generation: A Survey", University of Sheffield Regent Court, 211 Portobello Street Sheffield, S1 4DP, UK (2004).

[6] Praveen Ranjan Srivastava ET. al., "Generation of test data using Meta heuristic approach" IEEE, 2008, pp.19 - 21.

[7] Bo Zhang, Chen Wang, "Automatic generation of test data for path testing by adaptive genetic simulated annealing algorithm", IEEE (2011), pp. 38 – 42

[8] Abdul Rauf ET. al., "Automated GUI test coverage analysis using GA", Seventh international conference on information technology. IEEE (2010), pp. 1057-1063.

[9] Priyanka CHAWLA 1, Inderveer CHANA1, Ajay RANA2 "A novel Strategy for automatic test data generation using soft computing techniques" Higher Education Press and Springer-Verlag Berlin Heidelberg 2014Received December 14, 2013; accepted September 21, 2014.

[10] B. Beizer, "Software testing techniques", Second Edition, Ran Nostran Reinhold, NewYork, 1990.

[11] Windisch, S. Wappler and J. Wegener, "Applying paricle Optimization to software testing", ACM, GECCO, United Kingdom, pp. 1121-1128, 2007.

[12] C. Juang, "A hybrid of genetic algorithm and particle swarm optimization for Recurrent network design", IEEE Transactions on Systems Management and Cybernetics, Vol. 34, No. 2, pp. 997-1006, 2004.

[13] K. Li, Z. Zhang and W. Liu, "Automatic test data generation based on ant colony optimization", Proc. of Fifth International Conference on Natural Computation, IEEE Press, pp. 216-219, 2009.

[14] P. R. Srivastava, K. Baby and G Raghurama, "An approach of optimal path generation using Ant Colony Optimization", Proc. of TENCON, IEEE press, pp.1-6, 2009.

[15] Karaboga D. and Bahriye B. (2008), „On the Performance Of Artificial Bee Colony (ABC) Algorithm", Applied Soft Computing, Vol. 8, No. 1, pp. 687-697.

[16] C. Ramamoorthy, S. Ho and W. Chen, "On the automated generation of Program test data", IEEE Trans. Software Eng., Vol. SE-2, no. 4. pp. 293-300, Dec. 1976.

[17] M. R. Girgis, "Automatic test data generation for data flow testing using Genetic algorithm", Journal of Universal Computer Science, Vol. 11, No.5, pp. 898–915, 2005

Ms. Deepa Chauhan is pursuing masters program (M-tech) at Computer Science and Engineering (CSE) Department of Galgotias University, India. Her qualification includes B.Tech (CSE), M.Tech (CSE). She is a dedicated researcher in the field of soft computing, software testing and software engineering. She has authored various research papers and book chapters in journals of good repute.

Ms. Akanksha sehgal is M-Tech in Computer Science, India and B-tech in Computer Science and Engineering. She is presently serving as associate professor in the Computer Science and Engineering Department of Galgotias University, India.

ISSN: 2278 – 1323