International Journal of Engineering, Science and Mathematics (UGC Approved)
Vol. 6 Issue 8, December 2017,
ISSN: 2320-0294 Impact Factor: 6.765
Journal Homepage: http://www.ijesm.co.in, Email: ijesmj@gmail.com
Double-Blind Peer Reviewed Refereed Open Access International Journal - Included in the International Serial Directories Indexed & Listed at: Ulrich's Periodicals Directory ©, U.S.A., Open J-Gate as well as in Cabell's Directories of Publishing Opportunities, U.S.A

ROLE OF SOFT COMPUTING IN SOFTWARE ENGINEERING

Pooja Rana*
Rajender Singh*

Abstract:
Genetic Algorithm (GA) is an evolutionary computing approach which is recurrently used in different areas of Computer Science i.e software testing, software metrics, software complexity, Management with functional areas like finance, marketing, production, planning , Physics, and Engineering etc..[1]. GA is one of the dominant research areas. In this paper the concept of GA and applications of GA in different areas are demonstrated. The study reveal that genetic algorithm is effectively used to solve the different research areas of software engineering like software testing and software metrics etc..

Key words : Software Testing, software metrics , Genetic Algorithm, Chromosome, fitness function

Correspondence Author
Pooja Rana
Department of Computer Science and Applications
Maharshi Dayanand University,Rohtak

Introduction

"Genetic Algorithm" (GA) is one of the prominent evolutionary approaches. The general conception of "Genetic Algorithm" was proposed by John Holland. He is considered as the father of "Genetic Algorithms"[1]. Genetic Algorithm (GA) is a search-based optimization technique based on the principles of **Genetics and Natural Selection**. It is frequently used to find optimal or near-optimal solutions to difficult problems which otherwise would take a lifetime to solve. It is frequently used to solve optimization problems, in research, and in machine learning.

A Genetic Algorithm has five parts viz chromosome, the initial pool of chromosomes, a fitness function, a selection function, a crossover operator and a mutation operator. Genetic Algorithms are a tool to solve optimization problems. To solve a problem using GA some parameters need to be address regarding the initial population, the probability and type of crossover, the probability and type of mutation, the stopping criteria, the type of selection operator and the fitness function to be used in order to solve the problem. The performance of GA is affected by all these parameters and operators and they are interrelated.

Genetic Algorithm

Genetic Algorithms (GAs) are search based algorithms based on the concepts of natural selection and genetics. GAs are one of the branch of computation known as **Evolutionary Computation**.

*Research Scholar, Department of Computer Science and Computer Applications, Maharshi DyanandUniversity, Rohtak

*Professor, Department of Computer Science and Computer Applications, Maharshi Dayanand University, Rohtak

GAs were developed by John Holland and his students and colleagues at the University of Michigan, most notably David E. Goldberg and has since been tried on various optimization problems with a high degree of success.

In GAs is working on **pool or a population of possible solutions not on single solution** to the given problem. The initial population is generated randomly. These solutions then undergo recombination and mutation (like in natural genetics), producing new children, and the process is repeated over various generations.

Each individual (or candidate solution) is assigned a fitness value (based on its objective function value) and the fitter individuals are given a higher chance to mate and yield more "fitter" individuals. This is in line with the Darwinian Theory of "Survival of the Fittest".

In this way we keep "evolving" better individuals or solutions over generations, till we reach a stopping criterion.

A Genetic algorithm has typically five parts , a representation of a guess called a chromosome, an initial pool of chromosomes, a fitness function, a selection function, a crossover operator and a mutation operator.[2]
Chromosome can be a binary string or a more elaborated data structure. The collection of chromosomes with their corresponding fitness values is called population. The population at a particular instance is called generation. The initial pool of chromosome (population) can be randomly produced or manually created.[2]
Fitness function is one of the major decisive parameters of "Genetic Algorithm". It defines the objective of the problem to be optimized. A pair of chromosomes based on their fitness values is used to reproduce offsprings. The fitness function measures the suitability of a chromosome to meet a specified objective. A chromosome is fitter if it corresponds to greater coverage [2].
The selection function decides which chromosome will participate in the evolution stage of the genetic algorithm made up by the crossover and mutation operators. The crossover operator exchange genes from two chromosomes and creates two new chromosomes. After crossover operation, the genetic characteristics of the generated offspring are further modified. The mutation operator changes a gene in a chromosome and creates one new chromosome. Mutation is a procedure to modify the characteristics of the generated offspring to make it more effective [2].
The basic algorithm for GA is

1. Start
2. Initialize the set of n randomly generated solutions called chromosomes
3. Evaluate the fitness value of each chromosome.
4. Create new offsprings by applying genetic operators like selection, crossover and mutation etc.
5. Evaluate the new population generated
6. If the solution is satisfied with desired solution then stop and return the solution
7. Stop

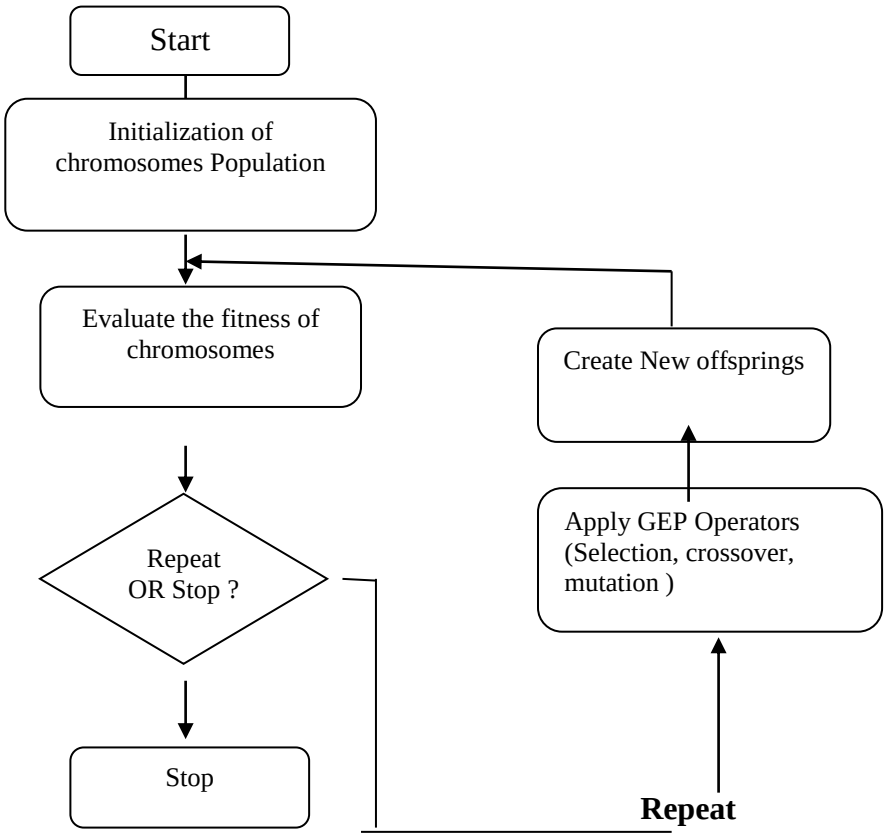
The algorithm will iterate until the population has evolved to form a solution to the problem or until a maximum number of iteration have taken place. [2]
It has been documented that if the initial population of solutions to the GA are good then the algorithm will perform better and opportunity of discovering a good solution is more [3]. If

the sample size of the population is not large enough then it is very difficult to find a good solution [4] .Some time a problem is quite difficult and has a limited solution available then the initial population is planted with some of possible solution and partial solutions of the problem [5].

Role of GA in Software Engineering:

Software engineering is one of the leading research areas. Software engineering is a methodical approach to develop and maintain the automated system. Some of the areas of software engineering are software testing, software metrics, software quality assurance, robotics etc.. To find out the bugs or error form the software module software testing is to be used. To find out the complexity of the software component or software module software metrics are used. Software quality assurance is used to judge and improve the quality of the software [6].

Software testing is one of the crucial phase of the software development life cycle. It helps to find out the errors and helps for removal of errors. Software testing phase takes lot of efforts, time and cost and the need to find appropriate test case generation technique to effectively improve the testing process. Genetic algorithm is used to improve the efficiency of testing tools used to generate test cases. Using genetic algorithm one can find most critical path in the software. Even for small scale problems GA outperforms



Flow Chart of Genetic Algorithm

To control software quality assurance genetic algorithm gives an appropriate solution. The objective is to generate optimal test data. Genetic algorithm gives the best solution as compared to others.

Role of GA in software Testing:

Software testing is a process in which the runtime quality and quantity of a software is tested to maximum limit. Software testing is the process of executing a program to find out the bugs. Software testing is performing a major role in software development life cycle. Software testing is laborious and time consuming work. The key problem of software testing is to generate test cases. The automated test generation improves the efficiency and effectiveness of software testing and helps to reduce the cost of software testing. The random symbolic and dynamic approach to generate test data is not enough to generate optimal amount of test data. Evolutionary testing uses a kind of Meta heuristic search technique, the Genetic Algorithm (GA), to convert the task of test case generation into an optimal problem [7][8][9].

Akshat and Ashish et. al. applied genetic algorithm in software testing. The result shows that by increasing the number of test cases GA becomes efficient. Genetic algorithm is used to increase the efficiency and process time of software testing. GA provides a mean of an automatic test case generator. Here the evolutionary generation of test cases can be applied and proves to be efficient and cost effective than random testing [10].

Praveen Ranjan and Tai-hoon et. al. authors compare random test data generation with a new approach using genetic algorithm. It exhibit that to improve software testing efficiency first is to search the most critical path by applying genetic algorithm. Author concludes that by examining the most critical path, a more effective way to approach testing is obtained, which helps to refine efforts and cost estimation in the testing phase [11].

J.McCart et. al. used genetic algorithm to improve the testing technique process by reducing the execution time as well as improving the ability to the technique in detecting more error [12].

The application of genetic algorithm in software testing is a new area of research that brings about the cross fertilization of ideas across two domains. Genetic algorithm is used to generate test cases while ensuring that the generated test cases are not redundant. In order to carry out the effectiveness of the test cases and test data the quantification, measurement and the perfect modeling is required which is done by using the accurate suite of software test metrics.

Role of GA in Software metrics:

Software metrics are numeric value related to software development. Metrics have traditionally been consisting through the definition of an equation, but this technique is limited by the fact that all the interrelationships among all the parameters be fully understood. The aim of research is to find the alternative methods for generating software metrics. Deriving a metrics using a GAs has several advantages.

Pedro A. Diaz-Gomez and Hougen et. al. they suggest the center of mass as an alternative metric to measurement diversity at the population level. Diversity becomes independent when the population size and chromosome length increases. For big population sizes, diversity could be measured in small portion or generate chromosome uniformly distributed in the fitness landscape. Author concludes that the computation time invested in calculating diversity and analyzing the initial population is going to be rewarded in the quality of the solution and number of iterations to get it [13].

Yogesh Singh, Bhatia and Sangwan et. al. proposed a model based on four parameter Changeability, Interface Complexity, Understandability of software and Documentation Quality for accessing software reusability levels using soft computing techniques like

Fuzzy Logic , Neural Networks and Neuro-Fuzzy. Authors proposed a soft computing technique to automatically predict software reusability levels i.e. very low, low, medium, high and very high. Combination of the neuro-fuzzy with the data sets generated by the fuzzy logic is to be taken for the advantage of some of the desirable features of neuro-fuzzy approach such as learning ability and good interpretability. Paper concludes that the neuro-fuzzy technique allows the integration of numerical data and expert knowledge [14].

Seyed Mohammad and Jalili et.al. proposed a novel GA-based algorithm SCI-GA (software Component Identification using Genetic algorithm. To define fitness function SCI-GA uses coupling, cohesion and complexity measurements. Authors had taken three real life cases to evaluate SCI-GA. The result reveals that SCI-GA can identify correct suboptimal software components, and performs far better than alternative heuristics like k-means and FCA-Based methods [15].

D M Thakore et.al. proposed a Secure Coupling Measurement Tool(SCMT) to generate all possible solutions by applying Genetic Algorithm(GA). SCMT uses coupling also feature of object oriented design to determine the security at design level. It takes input as a UML class diagram with basic constraints and generates alternate solutions. Tool also provides metrics at code level to compute the security at code level. The result of both the metrics gives proof of secure design [16].

Conclusion

The work reviews various applications of Genetic Algorithm in software engineering. It shows that GA helps to solve the various problem of software engineering. To get the optimal result from the software, GA is applied to various phases of software development like requirement and analysis phase and software testing, software reliability. It is also used for developing new metrics. GA has a big role in software testing phase, it is used for automatic generation of test cases. We can conclude that the application of Genetic Algorithm as an optimization technique to automatically generate test case helps detect more errors in program as it uses the best possible test case. GAs community helps the software professional to take the software development one step ahead. The studies must be carried out with large data sets to improve technique of test case generation. Moreover we can say that GAs is emerging field in software engineering.

References:

[1] Manik Sharma (2013), "Role and Working of Genetic Algorithm in Computer Science", International Journal of Computer Applications & Information Technology, Vol II Issue I, January 2013, (ISSN: 2278-7720)

[2] Shrivastva, Kim(2009), " Application of Genetic Algorithm in Software Testing", International Journal of Software Engineering and its Applications, Vol. 3, No 4, October 2009

[3] E.K. Burks,S.Gustafson and G.Kendall, " Diversity in Genetic Programming: An analysis of measures and correlation with fitness", IEEE Transactions on Evolutionary Computation, vol. 8, no 1 pp 47-62, 2004

[4] F.G. Lobo and C.F. Lima, " A review of adaptive population sizing schemes in genetic algorithms," in Proceedings of the Genetic and Evolutionary Computation Conference, 2005, pp 228-234

[5] D.A Casella and W.D. Potter," New lower bounds for the snake-in-box problem: Using Evolutionary techniques to hunt for snakes," in Proceedings of the Florida Artifical Intelligence

[6] Samriti, " Applications of Genetic Algorithm in Software Engineering, Distributed Computing and Machine Learning", International Journal of Computer Applications and Information Technology, Vol 9, Issue 2, July-Aug 2016

[7] D C Koboldt, K M Stienberg, D E Larson, "The next-generation sequencing revolution and its impact on genomics", Cell 155, No 1, 2013 pp 27-38

[8] S M Mohi-Aldeen, RMohamad and S Deris, " Automatic Test Case Generation for Structural Testing Using Negative Selection Algorithm", 2009

[9] V Rajappa, ABiradar, and S Panda,"Efficient software test case generation using genetic algorithm based graph theory", in 2008 First International Conference on Emerging Trends in Engineering and Technology , IEEE, 2008, PP 298-303

[10] Akshat Sharma, RishonPatani and Ashish Aggarwal, " Software Testing Using Genetic Algorithms", International Journal of Computer Science and Engineering Survey (IJCSES), Vol 7, No 2, April 2016

[11] Praveen Ranjan Srivastva and Tai-hoon kim, " Application of Genetic Algorithm in Software Testing", International Journal of Software Engineering and its Applicatons, Vol 3, No 4 october 2009

[12] J.McCart, D Berndt and A. Watkins, " Using Genetic Algorithms for software testing: Performance improvement tecniques", AMCIS 2007 proceedings, P 222, 2007

[13] Pedro A. Diaz-Gomez and Dean F. Hougen, " Initial Population for Genetic Algorithms: A Metric Approach" in GEM, 2007, PP 43-49

[14] Yogesh Singh, Pradeep Bhatia and Sangwan, " Software Reusability Assessment Using Soft Computing Techniques", ACM SIGSOFT Software Engineering Notes, January 2011, Volume 36 Number 1

[15] Seyed Mohammad Hossien and Saeed Jalili," SCI-GA: Software Component Identification Using Genetic Algorithm" , Journal of Object Technology , vol 12 no 2, 2013 pages 31-34

[16] D.M.Thakore, Torana N. Kamble, "Use of Genetic Algorithm inQuality Measurement", International Journal of Computer Applications, Volume 60- No8 December 2012

[17] Reena and Pradeep kr. Bhatia, "Application of Genetic Algorithm in Software Engineering: A Review", International Referred Journal of Engineering and Science (IRJES), vol 6, Issue 2, February 2017, pp 63-69.