Journal of Current Research in Engineering and Science
Bi-Annual Online Journal (ISSN : 2581 - 611X)

Volume 6- Issue 1, Paper 9 ,January 2023

Software Testing using Genetic Algorithms- A review

IshaanRajora, Lakshay Sandhu,
Mohammad Yamin Bhat, Rohit Beniwal
Dept .of computer science & Engineering
Delhi Technological University

Abstract

Software testing is the process of assessing and verifying that a software or application is working in the manner it is programmed. This paper is a literature review that reflects the evolution of genetic algorithms (GA) and how they have been efficiently used in different types of test case generation during functional software testing. We have focussed on set-based GA, cluster-based GA and hyper volume genetic algorithms which have been used for automated test data generation and for optimisation of that test data for solving variouscomplexproblemsrelatingtosoftwaretesting.Thispaperhighlightstheideasoftwaretestingusingvariouskindsofgeneticalgorithmsforoptimumresults.

Keywords - software testing, test-data generation, genetic algorithm, test-case prioritisation

Introduction

1.1 Overview

The role and significance of software status has expanded in recent years, as software has become increasingly vital in the global economy and societal evolution. Imperfect software can result in not just costly maintenance, but also major asset loss and, in certain cases ,serious national security or environmental risks. Software testing is of utmost importance in software programming since it is utilised to ensure the quality of softwares. Software testing has been shown to account for more than half of project expenditures in the overall life cycle of softwares. Furthermore, referring to Boehm's studies, if an issue is found later, it takes much more money and is costlier to rectify. So, it becomes crucial to improve the softwaretesting'sefficiency.Softwaretestingapproachesareprimarilyfordeterministicsoftware.In fact many actual programmes contain various forms of uncertainty, like randomness or fuzziness, implying that their behaviour is unpredictable.When executing the programmewithuncertaintymultipletimeswiththesametestdata,itmaytakevariouspathways,wrappingdifferentstatements,orevenproducedistinctresults.Previoustestadequacyrequirementsarenolongerapplicableinthissituation.

Software testing has been one of the most important processes to develop a reliablesoftwaresystembutitcanbesignificantlytimeconsuming.Thegoalistousetheleast amount of test data to find as many faults as possible. Testing particularly manual andad-hoccouldbesufficientforsmallbuildsbutforlargerset-ups automation testing comes into play. As software complexes further and further, testing becomes more and more challenging. In recent years, genetic algorithms (GA) have proven to be highly cost effectiveand efficient for test data generation. Moreover, GA is now being preferred for solving various software optimization problems.

Regression testing, a software testing practice that makes sure the unchanged partsof software sit well with the updated ones.The overall stability and functionality of theexisting features is dependent on it. For cost reduction, TCP is utilised for scheduling theofthetestcasestoenhancetheircapabilityforrevealingfaults.Orderingthetestcasestoexecute eventually is called Test case prioritization. Prioritising test cases aids in meetingtwo significant limitations, namely cost-time cost and budget cost-in software testing, toenhancethefaultdetectionrateasearlyaspossible.

1.2 Motivation

Therehavebeenvariousstudyresultsaimedattestingaprogrammewithnondeterminismin the past, but hardly any of these studies have focused on programmes containing ran-domness.Randomness-aware programmes, on the other hand, are common in actuality.Softwares for gaming, the Windows operating system as well as network software are fewexamples where randomness-aware programmes are used for instance when a user chal-lengesasoftwareprogramtoagameofChinesechess.Ingeneral,theprogram’sexecutionis determined by a set of strategies.However, certain random decisions may be included intheplan.Asnon-deterministicoptions,thecomputerdecisionswillbekeptundetermined.As a result, research into testing a programme using randomisation is both required andimportant.The program’sstabilitywill be ensured ifthe random behaviour’simpact couldbe determined on the programme. This could be achieved if some test cases could be madeusingsomerandomvariables.

Asuiteshouldbediscoveredintheprogramme’sinputdomainundertrialforan

Awarded set of programme goals ,it should be such that in the whole test suite there should be at least one test datum which could cover each target. Various experimental objectives have distinct requirements for testing.

2.1 Literature Review

NSGA-II was performed on difficult test problems by Kalyanmoy Deb et al.[1] and concluded that it will provide better solutions and converge better when compared with Pareto-archived evolution strategy (PAES) and strength- Pareto EA (SPEA). PAES were able to converge closer to the true Pareto-optimal front only in one single case. They proposedthatNSGAIIisstatedtobethebetteramongothermethodsobservedbecauseofutilisingd iversity preserving mechanism. Although this has been a matter of ongoing research insingle-objective evolutionary algorithm studies, this study displays that epistatic difficulties may also cause problems for MOEAs.They also introduced an extension to definedominance for mannered multi-objective optimisation, which when used with the real-coded NSGA- II and with this stated definition has been presented to solve these different difficulties much better than another recent stated approach.

Christopher C. et al. [2] talks about automatic software test data generation by usinggenetic algorithm.They described the execution of a genetic based system and observed theefficiency of this method. With their previous observation of this study they also examinethe complexity problem by executing their system on a number of synthetic programswithvaryingdifficulties.Theyconcludedtheirresultsbyperformingfourexperiments with the help of dynamic test data generation. In their experiment, the analysis of randomtestgenerationforcomparativelylargerprogramsdeclinedinperformance.Accordingto them, the increase in complexity of the program causes an increasing complexity fornonrandomtestgenerationmethods.However,standardgeneticalgorithmgavethebestresults forprogramswithvaryingdifficulties.Moreovertheyfoundthemostefficientwaystogeneratetes tdatabysatisfyingmanyrequirementswhichwerehighlyunlikelyandtheirdiscovery will help in solving most of the similar test generation problems and might

Journal of Current Research in Engineering and Science
Bi-Annual Online Journal (ISSN : 2581 - 611X)

Volume 6- Issue 1, Paper 9 ,January 2023

leadtosignificantdifferencesbetweenoptimisationanddynamictestdatagenerationissues.

AnautomatedtestcasegenerationbasedonGAwasdiscussedbyYuehuaDongetal.[3]thatproposedanimprovedGAforsoftwaretestinganddatageneration.TheimprovedGAhasmoreenhancedresultsthanthebasicGAbyproficiencyandvirtueonthetestcasegeneration.Theyusedabinaryencodingmethodduetoitseasyencodinganddecoding,simpletoattaincrossoverandmutationpotency.Toimprovetheaccuracyof selection operation of GA they decided to refrain in variation and crossover operation tomaintain best solutions and also decided to use preservation and roulette wheel selectionmethod for conjunction to fasten the overall convergence rate.For the mutation operation,theyusedthebasicbitmutationi.e.toselectavariationindividualarbitrarily,thenchoose a random place for a variation point.According to them fitness function affectsstraightlytotheconvergencespeedofGAandthepotentialtofindoptimalsolutionsotheyproposedafitnessfunctionaccordingtotheirrequirementofexperimentalproblem.Intheirexperimentanalysis,theimprovedGAbasedtestdatagenerationwascomparedwith the basic GA based test data generation approach and observed dominance on timeefficiencyandsearchcapability.

PraveenR.Srivastavaetal.proposedthecuckooandtabusearchalgorithms(CSTS)forautomation of test data generation. Tabu Search reduced the general complication of thealgorithmbycuttingthenumberofiterationsandexecutiontime.TheyusedLévyflightin solving the issues of getting stuck in local optima, thereby inspecting the search space moreeffectively. They combined the strength of the cuckoo algorithm to converge in minimumtimeusingthebacktrackingtabumechanismbyLévyflight[4].Theirexperimentswere basedonrelativelysimpleexampleswherethealgorithmprovedefficientingeneratingoptimal test cases and it performed significantly improved than previous approaches andvariousothermetaheuristictechniques.

2.2 ConceptsandTerminologies

To answer some research questions, Dario D. Nucci et al.[5] considered testing criteriawhich were distinct and widely used in previous TCP work: execution cost criterion, pastfaults coverage criterion, and statement coverage criterion [6].This study is a clear exampleofhypervolumebasedindicatorsandtheiradvantagesoversimpleAUCbasedmetrics.Thecriteria shows how the Hypervolume-based metric can satisfy any type of testing criteria.Utilising the testing models depicted over, the creators inspected two distinct

Journal of Current Research in Engineering and Science
Bi-Annual Online Journal (ISSN : 2581 - 611X)
Volume 6- Issue 1, Paper 9 ,January 2023

definitions ofthe TCP issue: Two criteria (Single- objective). The objective is to calculate an ideal orderof experiments or test cases which limits the execution cost and maximises the statementcoverage, three criteria (Two- objective).For this detailing, the authors considered thepreviousflawsinclusionasathirdmeasuretobeamplified.

2.2.1 Testadequacycriteria

Inordertotestasoftwareproductatestsuiteneedstobegeneratedaccordingtoacriteria.Thenthefaultsanderrorsareobtainedbyrunningtheprogram[7].Toguaranteetheappropriatenessofthetest,theauthorssetforwarddefiniteteststandardstorunthetestingresults.Particulartestgoalsrelatetodefinitetestmodels.Forinstance,executablestatementsareneededforstatementcoverageruleorcriteria.Asobserved generally when comparing, the coverage measure method is more complex than the branchconvergemethod.Similarlywhencomparedbranchcoveragecriteriatothestatementcoverage model,branchcoverageisobservedtobemorecomplex.

2.2.2 Multi-objectiveTestOptimisationProblems

We discuss a total of three multi-objective test optimisation problems: test suite minimi-sation(TSM),testcaseprioritisation(TCP),testcaseselection(TCS)[8].

With the evolution of a software project or application, the associated test suite contin-ues to grow alongside.Without careful maintenance of the test suite, it can easily lead toexcessively long test execution times, lowering the benefits of regression testing as bugs getdiscoveredlateindevelopmentorevenafterrelease.TSMisdesignedtosolvetheproblemoflong-runningtestsuitesbyremovingunnecessarytestcases[9].

TCP and TCS have helped software developers to get timely feedback on their productorapplicationastheyhaveimprovedregressiontestingthroughselectionandprioritisationoftestcases[10].ThegoalofTCPinsimpletermsisfindingtheorderinwhichagivensetoftestcaseswillbeexecuted,thusoptimisingagivenobjectivefunctionandsatisfyingtimeconstraintswhichhelpsinachievingtestinggoals[11].

TCS includes selection of a subset from a test suite which is used to check the changesmade in the software i.e. to check whether the changes made to the software affects theperformance of the unmodified parts [12]. The identity of the modified parts of softwareprogrammaybecompletedbytheusageofuniquetechniques.Thedetailsofthedifferentselecting approaches differ on how a selected approach defines, seeks and identifies adjustments within the application under test [13]. After the identification of test cases for theunmodified parts through a particular technique, we can use an optimisation algorithm,forinstancetheadditionalgreedy,forselectingaminimalsetofthosetestcasesinrelatio ntoacertaintestingcriteria[14].

2.2.3 Hypervolume

Thereisadevelopingpatternofsolvingmany-objectiveissuesutilisingqualityscalarmark-ers or indicators to consolidate various objectives into a solitary one [15].In this manner,rather than optimising the objective functions first, indicator-based algorithms discover asolutionsetthataugmentstheunderlyingqualityindicatortothemaximum[15].Perhaps

the most famous indicator is the hypervolume. It observes the nature and standard of thesolutionsuiteasthecompleteobjectivespace,whichinturniscontrolledby(atleastone)of such arrangements (combinatorial union [15]).For two objective and three objectiveproblems,thehypervolumereferstotheareaunderthecurveandthevolumerespectively.

3. ComparisonsofGeneticAlgorithms

3.1 Setbased genetic algorithm

Xiangjuan Yao et al.[7] introduced a special software testing generation approach forsoftwares with randomness and uncertainty, while previous practices of test data generationfrequently drop in efficiency.An algebraic model followed by a novel test adequacy criterionisputforwardtoprovidevitalitytothetestingsoftwares,accordingtothisanewapproachf ordecipheringtheoptimizationmodelbysetbasedGAisset.

Atestdatagenerationapproachformulti-pathcoverage,basedonageneticalgorithmwasintroducedwithlocalevolutiontoensuretheadequ acybyfindingerrorsbyrunningaprogramofthetestdata.Theydescribedthetraditionaltestingade quacycriterionfor a given software with set of test target and stated them valid for (softwares withoutuncertainties) test details awning a target with probability 0, 1 and for softwares withuncertainties and randomness coverage of test details for a target is not determined i.e.softwaremaybedistinctwhileexecutingthesametestdatum.

Followedbythestatedcriterionforsoftwareswithrandomness,tosolvetheoptimiza-tion model formulated by them, branch coverage criterion is seen as an instance to buildoptimised structure for softwares with uncertainties and a set based genetic algorithm isproposedandformulatedaccordingly.

Firsttheygovernthevaluesofcontrolledparameters,likethenumberoftestdata,thresholdetc,followedbycreatingandgeneratingarandominitialpopulationcontaininga number of individuals. Although the chances to be selected to the next generation isgreaterifthefitnessvalueofanindividualissubstantialandifthisconditionissatisfiedthan performing the genetic operation with selection, crossover and mutation operation isdone.

Their experiment analysis is based on ten C programs with random numbers.Theexperimental outcomes portray that the stated approach can resolve the difficulty of testdataforsoftwarewithuncertainnumbers.

3.2 Clusterbasedgeneticalgorithm

Dipesh Pradhan et al.proposed a CBGA-ES+ in addition to the previous CBGA-ESalgorithm for Multi-Objective Test Optimization [8].The design of CBGA-ES+ is to selectnon-dominated elite solutions from a group of clusters of the population that is where itdiffers from CBGA-ES as it includes only dominant elite solutions. These solutions will beused to generate the offspring solutions which will form the next generation. The clustersare sorted with the cluster dominance strategy and then the non-dominant solutions areselected among these clusters. The cluster dominance strategy has been used to draw adominance relation between two clusters.Each of the two clusters has a centre i.e.themean fitness of solutions of the cluster such that the cluster with a lower value of centredominatestheonewithahighervalue.

CBGA-ES+ is intended to compute a variety of multi objective test optimization chal-lenges. As a result, the inputs for CBGA-ES+ contains the initial test suite to be optimisedas well as a collection of parameters to be adjusted, such as population size, cluster size,and elite population minimum size.The minimal size of the elite population was set in order to avoid the algorithm from converging prematurely as a function of the added elitist selection. The algorithm initialises a random population of a given size. Solutions that are having similar fitness value are clustered using Lloyd’s algorithm. Lloyd’s algorithm selects one solution from Pt at random for each cluster and labels the

Journal of Current Research in Engineering and Science
Bi-Annual Online Journal (ISSN : 2581 - 611X)

Volume 6- Issue 1, Paper 9 ,January 2023

objectives as the cluster centres, respectively .It is important to note that the solutions chosen for each cluster have to be distinct, such that there aren't two clusters having the same centres.

Onceeverysolutiongetspartitionedintoclusters,theclustercentresareupdatedusingthemeanofsolutions.Whenthereisachangeinthevaluesofclustercentres,everysolutionisclusteredbycalculatingtheEuclideandistancebetweenthesolutionsandthecentresofthenewclusters.Followingthat,alloftheclustercentresareupdatedagain,andtheprocedureiscontinueduntilthevaluesofthecluster centredonot change for two consecutive iterations.Eventually,the clusters that we regenerated ae delivered,with each cluster consisting of a group of related solutions with regard to the pre-defined objectives.Following that,theLloyd's algorithm clusters are organized using the cluster dominance strategy,and the elite population is initialized with non-dominated solutions, which the nuses the algorithm dominance comparator.In particular,the non-dominated solutions from the best clusters are added to the elite population by comparing every solution with the solutions in the cluster.

The algorithm of dominance comparator takes the values of two solutions and checks whether one is dominated by the other, and then returns the outcome. The addition of solution is done to the elite population only when it's not dominant for any solution in the given cluster. This process is continued until either all of the solutions in the cluster are compared to one another or the size of the elite population equals the required population size. When the structure of the returned elite population is less than the given mini elitepopulationsize,computationsfromthenextdominatingclusterarepickedfortheelitepopulationusingthesameupdatedelitepopulationalgorithmuntilthesizeoftheelitepopulationequalsrequiredpopulationsize.

3.3 Hyper-volumebasedgeneticalgorithm

Test case prioritization is just generation of test cases to reveal specific faults in software. So it is a special case of test case generation.Ordering the test cases to execute eventually is called Test case prioritization.Prioritising test cases aids in meeting two significant limitations, namely cost-time cost and budget cost-in software testing, to enhance the fault detection rate as early as possible. In this paper, the case prioritised by Dario DiNucci et al.[5] is regression faults in software. Regression testing—a software testingpractice;thatmakessuretheunchangedpartsofsoftwaresitwellwiththeupdatedones.Since the capability of fault detection is not known before executing tests, the majority of the methodologies that are put forth for TCP use substitutes like coverage criteria with the possibility that experiments with better code coverage will reveal faults at a higherprobability.Once the coverage criterion is decided, search algorithms execute in a way that they find the order of maximizing that criterion.

Legitimate fitness functions are chosen and developed. Now, each of these fitness functions measures AUC addressed by the combined coverage and cost scores acquired aftersteadilyperformingtheexperiments(executingtestcases)asindicatedbyadistinctorder(prioritization). Numerous points in the cost-coverage space are consolidated into a soli-tary scalar value and utilised as a fitness fn for meta-heuristics, like single-objective GAs.Later work on search-based TCP likewise utilised multi- objective GAs, taking differentAUC-basedmetricsasvariousobjectivesforoptimization.

Hypervolume,usedinmany-objectiveoptimizationproblems,isjustanadvancedformof the AUC metric. Thus, A. Panichella et al. proposed HGA, which is a genetic algorithmbased on Hypervolume, to address the issue of TCP when multiple test coverage criteriaare used.They consider that it can deal with both— single cumulative code coveragecriteriaandmultipletestingcriteriainasinglescalarvalue.

Threeseparatecasestudieswerecarriedouttoaddresstheresearchquestion—IsHGAa lot quicker than GA and NSGA-II regarding efficiency.Furthermore, concerning Addi-tional Greedy, when the size of the program and the test suite increase, the effectivenessstaysunaffected.

While contrasting HGA and many-objective search based algorithms (e.g., MOEA/D-DEandGDE3),it was noted that it is more or just as effective,and the efficiency was 3times ber.

4. Results

CBGA-ES+algorithm has conclusivel yout performedallthealgorithmsfromTable4.1, 4.2 and 4.3 for TSM, TCP and TCS respectively [8]. HGA, on an average, is 1.89 times faster than GA. We already know number of test cases adversely affects the performance of GA. Indeed, with increasing number of test cases the ratio between the time required by GA and HGA increases.

HGA performs better than Additional Greedy in most cases—cost-effectiveness wise but efficiency is lesst.On the two-criteria formulation,HGA and GA perform the same interms of fault detectionability. However,the former has better efficiency than the latter thanks to our algorithm for the fast computation of the hyper volume. On the three-cri-teria, HGA is often has higher effectiveness and always has a higher efficiency than NSGA-II.

Table 4.1:CBGA-ES+ comparative performance for TSM

ComparedWith	$\hat{A}12$
CBGA-ES	0.71
MOCell	0.77
NSGA-II	0.79
PAES	1.00
SPEA	0.66

Table4.2:CBGA-ES+comparativeperformanceforTCP

ComparedWith	$\hat{A}12$
CBGA-ES	0.80
MOCell	1.00
NSGA-II	1.00
PAES	1.00
SPEA	0.62

Table4.3:CBGA-ES+comparativeperformanceforTCS

ComparedWith	Aˆ12

CBGA-ES	0.67
MOCell	1.00
NSGA-II	1.00
PAES	1.00
SPEA	0.99

5 .Conclusion

CBGA-ES+ performed better compared to its predecessor algorithms (CBGA, MOCell, NSGA-II, PAES and SPEA) for multi-objective test optimization problems. More of these optimization problems can be applied to test the CBGA-ES+ algorithm. In terms of cost effectiveness, HGA is better than Additional Greedy.HGA generated solution is not dominated by NSGA-II generated solution. Statistically, Additional Greedy is more efficient than NSGA-II and HGA, while HGA is faster than GA and NSGA-II.

Reference

[1] K.Deb,A.Pratap,S.Agarwal,andT.Meyarivan,“Afastandelit-istmultiobjectivegenetic algorithm: NSGA-II,” IEEE Trans. Evol. Comput., vol. 6, no. 2, pp. 182–197,Apr.2002.

[2] C. C. Michael, G. McGraw and M. A. Schatz, ”Generating software test data byevolution,” in IEEE Transactions on Software Engineering, vol. 27, no. 12, pp. 1085-1110,Dec.2001

[3] Dong, YuehuaPeng, Jidong. (2011). Automatic generation of software test casesbasedonimprovedgeneticalgorithm.10.1109/ICMT.2011.6002999.

[4] Srivastava, Dr. PraveenKhandelwal, RahulKhandelwal, ShobhitKumar, SanjayRanganatha, Suhas. (2012). Automated Test Data Generation Using Cuckoo

Journal of Current Research in Engineering and Science
Bi-Annual Online Journal (ISSN : 2581 - 611X)

Volume 6- Issue 1, Paper 9 ,January 2023

SearchandTabuSearch(CSTS)Algorithm.JournalofIntelligentSystems

[5] D. Di Nucci, A. Panichella, A. Zaidman and A. De Lucia, ”A Test Case PrioritizationGeneticAlgorithmGuidedbytheHypervolumeIndicator,”inIEEETransactionsonSoftwareEngineering,vol.46,no.6,pp.674-696,1June2020

[6] Z. Li, M. Harman and R. M. Hierons, ”Search Algorithms for Regression Test CasePrioritization,” in IEEE Transactions on Software Engineering, vol. 33, no. 4, pp.225-237,April2007

[7] X. Yao, D. Gong, B. Li, X. Dang and G. Zhang, ”Testing Method for Software WithRandomness Using GeneticAlgorithm,” inIEEE Access,vol. 8,pp. 61999-62010,2020

[8] D.Pradhan,S.Wang,S.Ali, T. Yue and M. Liaaen, ”CBGA-ES+: A Cluster-Based Genetic Algorithm with Non-Dominated Elitist Selection for Supporting Multi-Objective Test Optimization,” in IEEE Transactions on Software Engineering, vol. 47,no.1,pp.86-107,1Jan.2021

[9] Noemmer, R., Haas, R. (2020). An Evaluation of Test Suite Minimization Techniques.In:Winkler,D.,Biffl,S.,Mendez,D.,Bergsmann,J.(eds)SoftwareQuality:Qual-ity Intelligence in Software and Systems Engineering. SWQD 2020. Lecture Notes inBusinessInformationProcessing,vol371.Springer,Cham.

[10] Pan, RongqiBagherzadeh, MojtabaGhaleb, TaherBriand, Lionel. (2022). TestCase Selection and Prioritization Using Machine Learning:A Systematic LiteratureReview.EmpiricalSoftwareEngineering

[11] D. Marijan, A. Gotlieb and S. Sen, ”Test Case Prioritization for Continuous RegressionTesting: An Industrial Case Study,” 2013 IEEE International Conference on SoftwareMaintenance,2013,pp.540-543

[12] G.RothermelandM.J.Harrold,“Analyzingregressiontestselectiontechniques,” IEEETransactionsSoftwareEngineering,vol.22,no.8,pp.529-551,Aug.1996

[13] S. Yoo and M. Harman, “Regression testing minimization selection and prioritization:A survey,” Software Testing Verification Reliability, vol. 22, no. 2, pp.

67-120, Mar.2012.

[14] A. Panichella, R. Oliveto, M. D. Penta and A. De Lucia, "Improving Multi-ObjectiveTest Case Selection by Injecting Diversity in Genetic Algorithms," in IEEE Transac-tions onSoftwareEngineering,vol.41,no.4,pp.358-383,1April2015

[15] A. Auger, J. Bader, D. Brockhoff, and E. Zitzler, "Theory of the hypervolume indica-tor: Optimal m-distributions and the choice of the reference point," in Proc. SIGEVOWorkshopFound.GeneticAlgorithms,2009,pp.87–102.