International Journal of Applied Engineering Research ISSN 0973-4562 Volume 12, Number 21 (2017) pp. 11700-11706

An Adaptively Optimized Approach on Random Test Case Generation Using Intuitionistic Genetic Algorithm

G.Manivasagam
Ph.D Research Scholar, Department of Computer Science,
Karpagam Academy of Higher Education, Coimbatore- 641021, Tamil nadu, India.
Orcid: 0000-0002-1157-1349

Dr. R.Gunasundari
Associate Professor, Department of Information Technology,
Karpagam Academy of Higher Education, Coimbatore- 641021, Tamil nadu, India.
Orcid: 0000-0003-4157-285X

Abstract

Objective**:** Day by day the IT field rapidly changed and it leads to increase in software complexity the testing of software is highly cost effective and it takes more processing time because in research it accumulate large amount of data.

Findings: This paper introduces an enhanced approach on automatic test case generation by overcoming the existing problem in random testing. The proposed method increases the coverage of the input space by using fuzzy C-means based Centroidal Voronoi Tessellation. The input space is obtained from already available methods like adaptive random testing and quasi random testing. This proposal acts as the add-on for the former random testing techniques. The Test cases at the border of the CVT are well identified by using Intuitionistic Fuzzy C-means which fails in the traditional version. The evolutionary approach is used to pick potential test cases for testing the software. The experiment was conducted in addition to three failure patterns block, point and Strip.

Improvements/Applications*:* The result of the proposed framework reveals the high contribution of the work comparing with the previous random testing techniques. The test cases are automatically produced in a effectual manner. Similar test cases are clustered with Intuitionistic Fuzzy C-means algorithm

Keywords: Random testing, fuzzy c-means, genetic algorithm, Intuitionistic Fuzzy C-means (IFCM), Centroidal Voronoi Tessellation (CVT)

INTRODUCTION

The most essential development part of software is testing. In a discipline of the software engineering, software testing is a one of a primary pace whilst it takes more time and it needs more money in the development of software life cycle [1].

The high impact of a software testing process mainly relies on generation of test case. It is a critical task to produce the effectiveness of testing process in terms of test cases. A test case is defined as successive execution of set of tests which is highly dependent to a test objective which generates set of tests comprising of input, observed and expected output and extra information essential for running the test like setting up environmental variables [2]. With the aid of test cases the entire application can be covered by all the possible combination of testing. But it is not possible to a complete overall testing due to infinite number of test cases.

The optimal quality of test case depends on least cost and coverage of test objectives with high number of features. The existing methodologies are categorized as random path-based, gold-based and few of them relay on machine learning approach in generation of test cases. In this paper an efficient approach based on random testing is introduced

RELATED WORK

In [3] the authors specify it is highly complex and time consuming to develop fault oriented algorithms. In simulation based test case generation Genetic Algorithm as a base structure here, data points are represented in the form of strings in terms of representation of chromosomes and populations. To evaluation of CRIS test generator to find the logic simulator performance comparison on test sequences of candidates is utilized. In addition CRIS coverage of faultiness is very low comparative deterministic test generation.

A modeling architecture was designed by Cavarra et al. [4]. Its structural design consists of two different components. Both components are written in UML. The architecture of system model represents first component. It is collection of class diagram, state diagram and object diagram in a system the entities are represented using diagram of the class. Justification in evolving of entities is represented using diagram of state. The last diagram is that is object which is specifies configuration during initialization of entities.

The second component is test directive which compressive object and state diagram. The criteria of state coverage is represented using object and state is used for identification of test. Building of system model and test directive can be done using any of the tools set like rational rules.

T.Y. Chen et al. [5] presented category based test case portioning using choice relation. Manually the choices constrains should be defined. Using the different relations the constrains are captured in a regular manner. It was a specification based testing method which aids in developing test cases by cleansing the specification of a program in terms of functional to test. The consistency of the proposed work is determined using choice relations atomized deduction.

Franck Fleurey et al. [6] developed an object oriented embedded software which completely automates test case derivations. The main objective of this work is to generate an efficient fault detection technique which covers the system strategy in a maximal rate by producing powerful test cases. Monalisa sarma [7] devised a UML diagram based test case generation in this the test generation regarding information are stored in STG. The algorithm travels the STG in two levels. It visits all the possible use cases and produce efficient fault deduction.

Alexander pretschner et al. [8] adopted Eiffel library which contains 27 classes and based on it and automatic test case generation is produced. It is cheap and time consuming because of its testing strategy. The resultant and proof is that the efficiency on predicting defects is high with limited time period. Hassan Reza et al.[9] founded a testing model for application of web pages based on the behavioral analysis derived from chart of state model. This model mainly concentrates on designing and testing websites. It also tested the functionality of frontend of web applications.

Chandran and Prasanna M.[10] proposed a model to generate test cases for the object diagrams. This methodology various steps were evolved. Object diagrams were constructed, stored and object are named. The tree was build using these object names which further applied with crossover operator of GA. These new generation trees are converted into binary tree and traversed using DFS technique and this gave all the valid, invalid and termination sequences for a given application. Mutation analysis was done by infecting faults into the system are then results are compared with original, if there exists some differences then these mutants are considered as killed test[16].

Kulvinder Singh et al. [11] stated that genetic algorithm could be widely used as optimization techniques. Some of the researchers concluded that genetic algorithm has important impact on the optimize test cases generation. The conventional methods did not given fault free software in short time. It took them long period of time. So there was need of automated test case generation using genetic algorithm which could found large vector of faults in short time. This was demonstrated with the help of an example in the paper. Due to the highest impact on genetic algorithm the researchers the optimized output on test case generation. The traditional methods takes long time to produce fault free software it necessitates the need of automated test case generation by adapting genetic algorithm in short time.

RANDOM TESTING (RT)

The input domain is used for selecting test cases randomly in Random Testing (RT) [12]. In the absence of specification of software and source code the random testing generates test cases automatically it is one of its major merits. In addition, it also lessens the factor of computing complexity by reducing the cost with its characteristics of randomness. The chaos of the system operational cost can best reflect the performance of randomness as a result. The deterministic approach fails to determine certain software failures whereas RT can detect and reveal such type of failures.

Random testing also has limits such as wastage of resources increasing in sequence of events that are illegal and not reachable. Whilst the control in selecting sequence of event is not under the control of test designer. All these factors of RT are not producing optimal result in fault deduction. Though testing randomly is a common nature any considerable enhancement in the strategy leads to noticeable effect. The two most importantly used approaches in this work are adoptive and quasi random testing.

OVERVIEW OF GENETIC ALGORITHMS

In terms of optimization the evolutionary algorithm which best suits for searching maximal test cases is done by genetic algorithm [14, 15] which has the feature of representing test cases in the form of binary strings using its evolutionary operation selection mutation and crossover. In this case the set of test cases which covers maximum logical path of a program holds the highest fitness value. The test cases which have optimal fitness value are considered for testing. This is adopted by the fittest of the survival is a goal of genetic algorithm. In each iteration the test cases with highest fitness value is considered for generating new population using cross over among the individual test cases and to overcome pre-mature convergence on off spring based on mutation process.

Proposed Architecture for Intuitionistic Fuzzy C-Means based RBCVT

With the input of points generated and using probabilistic method in this work random boarder centroid voronoi tessellation is computed as follows,

Step 1: find the set of initial test cases represented by T = $\{ti\}_{i=1}^{|T|}$ as generators, ti ∈ T, where i = 1 . . . |T|.

Step 2: choose a set of boarder point randomly using R = $\{rn\}_{n=1}^{|R|}$, n=1 in which rn ∈ R, where n = 1 . . . |R|. In addition, the grouping of T and R is termed as TR = T ∪ R = $\{t \cup m\}_{m=1}^{|TR|}$

RANDOM POINT SELECTION USING INTUITIONISTIC FUZZY C-MEANS CLUSTERING

Intuitionistic Fuzzy C Means (IFCM)

Based on the theory of intuitionistic set the fuzzy C means [13] was framed. The degree of membership function in fuzzy set is not suitable for finding uncertain presences of test points lying on the boarder so, the degree of non-membership function is also included with the aid of intuitionistic fuzzy.

Intuitionistic fuzzy represents memberships function as μ(x), and non-membership as ν(x). An intuitionistic fuzzy set [14] A in X, is written as:

$$A=\{x, \mu_A(x), \nu_A(x) \mid x \in X\}$$

Where $\mu_A(x)=[0,1]$, and non-membership $\nu_A(x)=[0,1]$ are the membership and non-membership degrees of an element in the set A with the condition:

$$0 \le \mu_A(x)+ \nu_A(x) \le 1$$

When $\nu_A(x) = 1- \mu_A(x)$ for each x in the set A, becomes a fuzzy set. And introduction of hesitation degree $\pi_A(x)$ is converted in to Intuitionistic fuzzy . now able to handle imprecise knowledge of selecting test points

$$\pi_A(x) = 1- \mu_A(x) - \nu_A(x) ; \ 0 \le \pi_A(x) \le 1$$

Due to hesitation degree, the membership values lie in the interval

$[\le \mu_A(x), \mu_A(x) + \pi_A(x)]$

Intuitionistic fuzzy c-means [13] contains it objective function in two different terms: (i) modified objective function and (ii) intuitionistic fuzzy entropy (IFE). which is represented as,

$$J_{IFCM}= \sum_{i=1}^{c}\sum_{k=1}^{n} \mu_{ik}^{*m} d_{ik}^2$$

$$\mu_{ik}^* = \mu_{ik} + \pi_{ik}$$

where μ_{ik}^* denotes the intuitionistic fuzzy membership matrix and μ_{ik} denotes the conventional fuzzy membership of the kth data in ith class.

π is hesitation degree, which is defined as:

$$\pi_{ik} = 1- \mu_{ik} - (1- \mu_{ik}{}^{\alpha})^{1/\alpha}, \alpha > 0$$

Resilient Intuitionistic Fuzzy Cmeans Clustering

1. Find the initial centroid using intuitionistic fuzzy objects randomly. And it is represented as C.
2. Calculate the membership value and hesitation value for IFCMM using the following equations,

$$\mu_{ik} = \frac{1}{\sum_{j=1}^{c}\left(\frac{d_{ik}}{\bar{d}_{jk}}\right)^{\frac{2}{m-1}}} \forall k, i$$

$$\pi_{ik} = 1- \mu_{ik} - (1- \mu_{ik}{}^{\alpha})^{1/\alpha}, \alpha > 0$$

$$\mu_{ik}^* = \mu_{ik} + \pi_{ik}$$

$$S_1(A,V) = \frac{S'(\mu_A(x_i), \mu_V(x_i) + S'(\nu_A(x_i), \nu_V(x_i))}{2}$$

3. Modified cluster centers are:

$$\nu_i^* = \frac{\sum_{k=1}^{n}(\mu_{ik}^{*m} x_k)}{\sum_{k=1}^{n}(\mu_{ik}^{*m})} \forall i$$

4. Update degree of membership and degree of membership and non–membership.
5. Repeat step 2 to step 4 until converges.

Pseudo code for fuzzy genetic Algorithm

6. $t := 0$;
7. Initialization: $P(0) := \{\vec{a}_1(0),...,\vec{a}_\mu(0)\} \in I^\mu$;
8. *Evaluation*: $P(0) : \{\Phi(\vec{a}_1(0)),...,\Phi(\vec{a}_\mu(0))\}$;
9. while $\iota(P(t)) \neq$ **true** do //
10. Stages for Reproduction

Selection process :

$$P'(t) := s_{\Theta_s}(P(t));$$

Recombining process:

$$P''(t) := \otimes_{\Theta_c}(P'(t));$$

Process of mutation:

$$P'''(t) := m_{\Theta_m}\left(P''(t)\right);$$

Process of evaluation:

$$P'''(t) : \left\{\Phi\left(\vec{a}_1'''(0)\right), \ldots, \Phi\left(\vec{a}_\lambda'''(0)\right)\right\};$$

Process of replacement :

$$P(t+1) := r_{\Theta_r}\left(P'''(t) \cup Q\right);$$

$t := t + 1$;

end while

Architecture of Proposed Framework

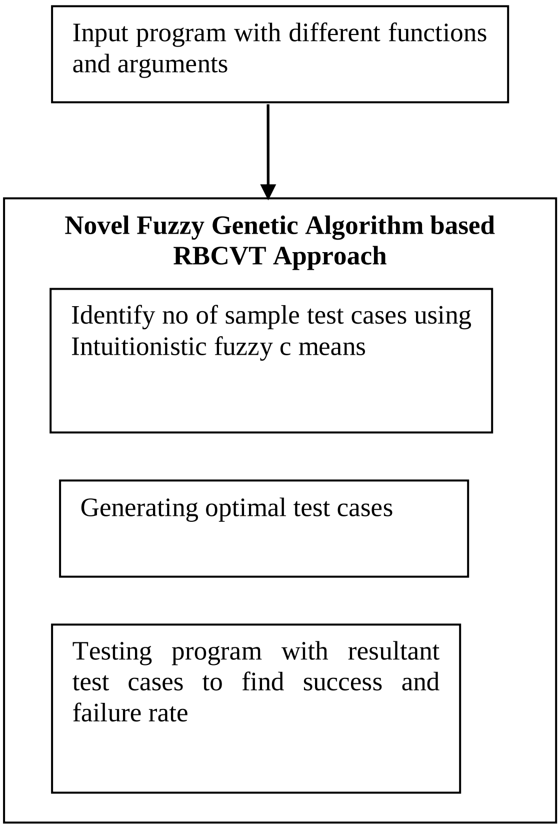

Figure 1: Block diagram of proposed system.

As shown in the figure1 the architecture of the proposed work shows clearly that a toughest part in testing of the software is test cases generation. The main motivation of test case is that covering more features of test which represents the quality of test cases. The selection of minimal test cases which covers adequate criteria to be acceptable in case of testing leads to reduction in testing cost and efficient time consumption.

In this proposed work each test points are clustered based on the degree of membership and non-membership values. The outcome of this techniques revels that each test points does not completely belongs to a single cluster. It holds the membership and non-membership to each cluster. To select minimal number of test cases for a particular constrains of a test the fuzzy based genetic algorithm is used for selecting potential test cases which in each test cases.

The advantage of selecting fuzzy logic is to improve its behavior and designing component. It also extends the limit of GA in fuzzy based searching problem. The process involved in fuzzy genetic algorithm is setting the input and output criteria, deploying fitness finder in terms of fuzzy [16, 17], converting the input value to the fuzzied value, inferring the knowledge using fuzzy inference search engine and at last output defuzzification. In test case generation fuzzy genetic tree was proposed and its distanced are based on fuzzy fitness function. This proposed work proves the successful outcome of producing test cases which covers full branch.

Parameters for generating Test Case in this proposed work

This proposed method uses adoptive random testing for implementing this experiment. The population size K is set to 20 and the ratio for coverage is 2.5. crossover probability is set to 0.5 , the mutation probability is set to 0.1, the mutation size is set to .01 and 200 iterations are set as stopping criteria. The clustering frequency is set to 10.

Simulation Framework

This proposed work was simulated using the net beans and the failure patterns are used and the rate of failure associated related to patterns is considered, in addition the test cases used in each test and the number of testing sets are also produced in this work.

Failure Patterns and Failure Rates

To find the ability of resultant test cases generated from the proposed intuitionistic fuzzy method some of the domain of inputs is assumed to be area of failure. While testing is done on this particular the test case should shows the failure of the output. This is to find whether it is capable of finding failure area. Although there are several existing works are under investigation in order to analyze which type of pattern is to be followed for designing input domain this work adapts the block pattern which will choose the test points randomly by constructing a pattern of square surrounding the test points according to the rate of failure. During the selection of test points which lie on the boundaries the pattern of block may does not fit properly within that area so the block is reconstructed each time when it fails to cover the boundary test points. The pattern of strip is produced by choosing the point randomly in the search space of test cases and a angle at random rate is chosen with a line passing across the chosen point. In accordance with the rate of failure the width of the pattern of strip is computed.

Performance Comparison of the proposed method with the existing approaches

The table 1 and figure 2, 3 represents the performance comparison of the existing methods before and after intuitionistic fuzzy based genetic algorithm based RBCVT based on Mean and Standard Deviation is shown clearly.

Table 1. Comparison of Existing methods before applying proposed RBCVT and after applying proposed Intuitionsitic fuzzy based genetic algorithm RBCVT

Method	Before the Proposed IFGARBCVT Process		After the Proposed IFGARBCVT Process	
	Mean	**Std**	**Mean**	**Std**
RT	0.016	0.0029	0.0310	0.0038
Sobol	0.0091	0.0038	0.0303	0.0054
FCFS	0.011	0.0021	0.0302	0.0038
ART	0.0123	0.0024	0.0311	0.0029
QRT	0.0412	0.0039	0.0702	0.0048

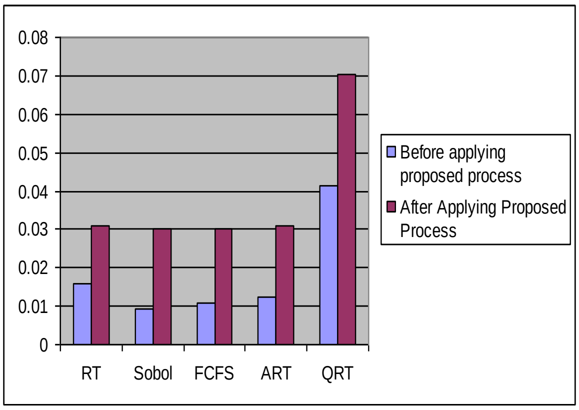

Figure 2. Comparison result of mean value of various testing methods with before and after applying proposed process

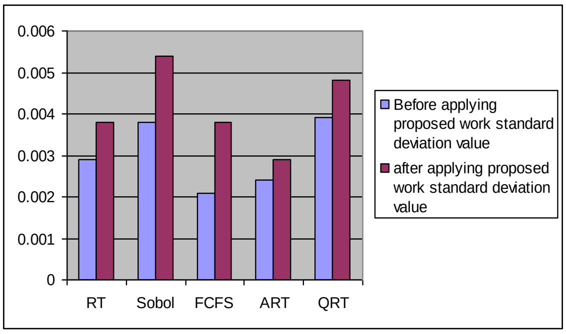

Figure 3: Comparison of Standard deviation value with different testing techniques before and after applying proposed process

Performance comparison based on various rate of failure

In comparison with the existing techniques used in this work the effectiveness in terms of size in obtained by the proposed IFGARBCVT as shown in Table 2, and at the same time no other individual produce effective increasing size.

Table 2: Improvement of test case generation methods with respect to IFGARBCVT process at different failure rates regarding the block failure pattern

Method	"10^{-2}"	"10^{-3}"	"10^{-4}"	"10^{-5}"
RT	0.44	0.34	0.26	0.25
Sobol	0.32	0.21	0.2	0.12
FCFS	0.45	0.18	0.13	0.09
ART	0.56	0.31	0.06	0.06
QRT	0.68	0.2	0.04	0.03

The table 2 and figure 4 shows outcome of each approach namely RT, Sobol, FCFS, ART and QRT on applying proposed IFGARBCVT process with varying failure rate depending on the pattern of block chosen. It is shown clearly that while there is a decreasing failure rate depending on the selection of test cases the level of changes observed.

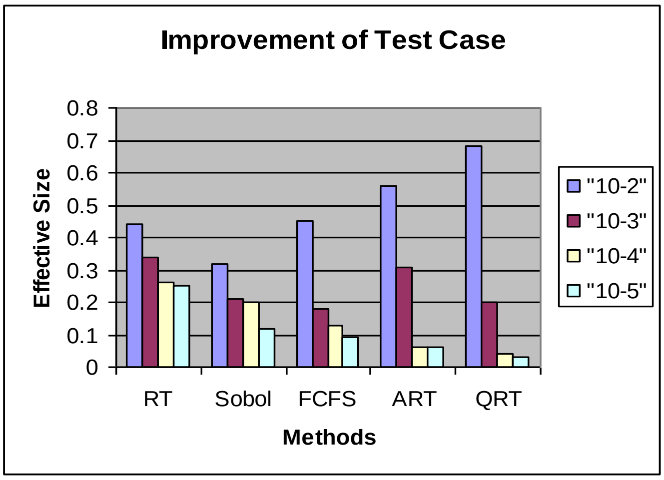

Figure 4 Comparison of test case generation based on the different Effective Size

Performance comparison of Existing approaches with the Proposed IFGARBCVT adapter approaches based on the p-measure testing

Table 4. Comparison of effective testing using P-measure.

Method	10	20	50	100
Sobol	2	3	3	2
RT	4	4	5	4
FCFS	5	6	8	7
ART	8	7	9	8
QRT	12	13	11	9
RT-IFGARBCVT	15	16	13	10
FCFS-IFGARBCVT	17	18	15	11
ART-IFGARBCVT	19	18	16	13
QRT-IFGARBCVT	22	20	18	15

With the proof of simulation result the table 3 and the Figure 5 depicts. The testing measure of P-test the effective size of each method is considerably increased after applying **IFGARBCVT.** According to the table the performance of QRT- IFGARBCVT produces more efficiency with highest rate of test case size selection and the worst result is obtained from sobol technique. On the hole the performance of each technique is enhanced by adopting IFGARBCVT.

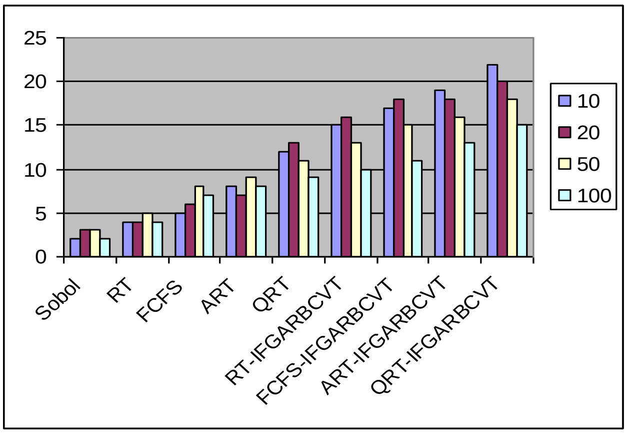

Figure 5: Comparison of effective testing uses a P-measure.

CONCLUSION

In this paper, a novel intuitionistic fuzzy modeled genetic approach is developed to tackle the \implicit problem of random testing in selection of test case generated is proposed. The main goal of this proposal is handling hesitated test points in the boundaries of test regions using innovative use of intuitionistic fuzzy c means to cluster the generated test cases to different clusters by assign membership and non-membership value for representing values of nodal by utilizing curves under smoothen area. The proposed methodology holds its own advantage by adaption degree of hesitation in handling uncertainty in test case and testing the performance of software. The influence of different parameters in applying genetic algorithm, concerning to initialization process, procedure for selection, ratio dependent ranking, and the production of nodal values, the number of steps, maximum count of residual node, and size of population is also analyzed. This proposed work is applied on various testing techniques and the result shows the promising improvement on selection of optimal test cases.

REFERENCES

[1] F.-C. Kuo, “An Indepth Study of Mirror Adaptive Random Testing,” Proc. Ninth Int’l Conf. Quality Software, pp. 51-58, 2009.

[2] T. Chen, D.H. Huang, T. Tse, and Z. Yang, “An Innovative Approach to Tackling the Boundary Effect in Adaptive Random Testing,” Proc. 40th Ann. Hawaii Int’l Conf. System Sciences, p. 262a, 2007

[3] Rudnick Elizabeth M., Greenstein Gary S., “A Genetic Algorithm Framework for Test Generation” IEEE Transactions on Computer Aided design of Integrated Circuits and Systems, VOL. 16, NO. 9, 1997.

[4] Cavarra, A., C. Crichton, J. Davies, A. Hartman, T. Jeron and L. Mounier. Using UML for automatic test generation. Oxford University Computing Laboratory, Tools and Algorithms for the Construction and Analysis of Systems, TACAS 2000.

[5] T.Y. Chen, Pak-Lok Poon and T.H Tse," A choice Relation Framework for supporting category-partition Test case Generation", IEEE Transactions on software engineering Vol. 29, No.7, 2003.

[6] Clementine Nebut, Franck Fleurey, Yves Le Traon and Jean-Marc Jeze quel, "Automatic Test Generation: A Use Case Driven Approach", IEEE Transactions on software engineering Vol.32, No.3, 2006.

[7] Sarma Monalisa and Mall Rajib, “Automatic Test Case Generation from UML Models,” 10th International Conference on Information Technology, pp. 196-201, 2007.

[8] Ilinca Ciupa, Alexander Pretschner, Andreas Leitner, Manuel Oriol, Bertrand Meye: "On the Predictability of Random Tests for Object-Oriented Software" International Conference on Software Testing, Verification, and Validation, 2008.

[9] Reza, H., K. Ogaard and A. Malge," A model based testing technique to test web applications using statecharts", Proceedings of 5th International Conference on Information Technology: New Generations, pp. 183-188, 2008.

[10] Prasanna M., Chandran K.R., "Automatic Test Case Generation for UML Object diagrams using Genetic Algorithm",Vol.1,No.1,2009.

[11] Singh Kulvinder and Kumar Rakesh, "Optimization of Functional Testing using Genetic Algorithms" International Journal of Innovation, Management and Technology, Vol. 1,No. 1, 2010.

[12] Utting, Mark, and Bruno Legeard. Practical model-based testing: a tools approach. Morgan Kaufmann, 2010.

[13] Atanassov, K. Two Variants of Intuitonistc Fuzzy Propositional Calculus. Preprint IM-MFAIS-5-88, Sofia, 1988

[14] Atanassov, K. Intuitionistic Fuzzy Sets. Heidelberg, Springer Physica-Verlag, 1999

[15] Tamalika Chaira. Author links open the author workspace.
A novel intuitionistic fuzzy C means clustering algorithm and its application to medical images Applied Soft Computing, Volume 11, Issue 2, March 2011, Pages 1711-1717

[16] Klir, G. J., Yaun, B. 1995. Fuzzy sets and Fuzzy logic theory and applications. Appendix B- Genetic Algorithms- An overview. Pearson Education Inc

[17] Konsaard, P., Ramingwong, L. 2015. Total coverage based regression test case prioritization using genetic algorithm. 12th International conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology. IEEE.