iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 6 **Issue:** VII **Month of publication:** July 2018

DOI: http://doi.org/10.22214/ijraset.2018.7139

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 6.887
Volume 6 Issue VII, July 2018- Available at www.ijraset.com

A Particle Swarm Optimization Algorithm for Test Scheduling in Agile Testing

Bhuvneshwari[1], Ruchika Sharma[2]
[1]*Research Scholar,* [2]*Associate Professor*
[12]*Baddi University of Emerging Science and Technology, Solan, India*

Abstract: ***The testing is the technique which is used to test the software. The various types of testing are proposed which will test the software and test number of faults. In this work, agile testing is discussed which contains the multiple testing's. These testing are black box testing, white box testing, mutation testing and model based testing. There are various type of testing's can be applied in the agile testing. To execute agile testing in the efficient manner, the testing's need to be scheduled . When the testing's of the agile testing is not scheduled in the efficient manner, it leads to detection of fault in incorrect manner. In this work, improvement is been proposed in the agile testing which can schedule the testing'. To schedule the testing's technique of PSO technique is applied which will schedule the testing's. The proposed and existing technique is implemented in MATLAB and it is been analyzed that proposed technique performs well in terms of fault detection rate.***
Keywords: Software Engineering, Testing, Agile Testing, Particle Swarm Optimization algorithm, Scheduling

I. INTRODUCTION

Software Engineering is the branch of study and application of engineering to the software which initially design development of the same software by using various technique or updations, maintenance of the software. Software engineering deals with the all kind of software production, design to coding, software accuracy and deals with the complexity of any software system. Software industry is moving very fast in the current scenario [1]. Even big industries spend large some of amount on their software engineer for the software development. Basically software engineering is a study and integral part of engineering which include design, development and maintenance of the system. Software Engineers are applied principles to the software engineering to develop, design, test and maintain any software. There are clear likenesses between the orders of software engineering and instrument design, yet enough contrasts to entice the transfer of effective development techniques from either train to the next [2]. The SEED project has systematically contemplated these similarities and differences in its attempt to transfer innovation and skill from one control to the next. Simultaneously, an organizational framework for systems development methodologies has been built to portray, manage and apply the engineering design process to the development of heterogeneous, composite systems. Software engineering research has created a multitude of specification and design techniques that might be utilized to portray framework requirements and design architectures [3]. These strategies commonly use various diverse representation styles or notations together with solutions of how to produce specifications utilizing these notations. Numerous general issues, for example, fragmentation, irregularity and ambiguity in specification have been experienced and effective methodologies created to attempt to determine them. PC Aided Software Engineering (CASE) tools and integrated programming support environments (IPSEs) have risen to give down to earth, automated support for these techniques. Such PC based tools give a way to authorizing techniques' hidden process models utilizing the notations recommended by these strategies. Various software life-cycle models have been produced in the most recent three decades [4]. These models, albeit accommodating is giving general guidance to software developers, don't uncover horde subtle elements that are basic in any substantial software development venture. A critical stride past instantiating a process model is interpreting the resultant process description, in this way ordering the planned software process. The understanding of a process description can be manual or automated. Manually translated process descriptions help individuals in taking after the suggested process models amid software development. The processes of software testing and fault detection keep on challenging the software groups [5]. Despite the fact that the software testing and fault detection exercises are inaccurate and insufficiently comprehended, they are crucial to the accomplishment of a software project. Software testing is a process which is used to identify error or faults in a system to make it correctness, completeness and to identify the quality of already developed software. Software testing is a process which is used to detect the bugs and uncover it. Software testing is a process and discipline also. White-box testing is also known as Glass testing and structural testing. In white-box testing code is visible. It has knowledge of the internal mechanism of the components. White-box testers are aware about the internal structure and also know how code is looks like [6]. It is used in the validation process.

©IJRASET: All Rights are Reserved

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 6.887
Volume 6 Issue VII, July 2018- Available at www.ijraset.com

Black-box testing is also known as functional testing. This testing ignores the internal mechanism of the system. It is a testing which is based upon the output and having no knowledge of internal code. It is a type of white-box testing. It is a testing for low-level design code. It is done within a class and starts from a individual module. It has testing the smallest unit of elements of software which module or component or unit .It is basically done by programmer not tester and requires detail of structure of code. Regression testing is used to retest the component of a system that verifies that after modifications defects are removed from the in effected software. Automation tools are required for these types of testing. The process of verifying the modified software in the maintenance phase is known as Regression testing. Time and budget constraints are its major disadvantage due to complex process [7]. Agile Testing is a testing which follows the rules of the agile software development. Agile is an iterative development methodology, where requirements evolve through collaboration between the customer and self-organizing teams and agile aligns development with customer needs. Agile ordinarily alludes to an expert tester who embraces changes, works together well with the business technical team and comprehends the idea of software document requirement and to infer them. They will find out about the customer so they can recognize what they can do as per the customer needs according to the software requirements.

II. RESEARCH METHODOLOGY

In this case study various modules of the website are considered and their test cases. The agile testing will be applied on each module of the website with their developed modules. To select best value, technique of PSO will be applied which select value on the basis of type of software for which test cases are going to generate. The Particle Swarm Optimization is the algorithm utilized as the optimize solution using which difficult problems are solved. The development of PSO is modeled by the bird flocks and schools of fish. It is considered as the evolutionary computation technique but in some manner it differs from other algorithms such as genetic algorithms. In the initial state, there is initialization of the swarm of particles in the PSO that proceed with a random solution to the problem. In the given velocity around the search space particles moved. A potential solution is represented by the location of a particle at a given point in which the fitness function is used to evaluate fitness. The previous generations has been utilized in order to govern and generate subsequent generations. Therefore, in order to find out the optimal solution for the given problem, the particles moved around the search space to produce new generations. Until the generation limit reached or the optimal particle in the population cannot be improved this process continues. The behavior of the swarm is given by the following equations:

$$v(t+1) = v(t) + \emptyset \,.\, (x - x_p) + \emptyset \,.\, (x - x_n)$$

$$x(t+1) = x(t) + v(t+1)$$

A. *Proposed Algorithm*

1) *Input:* training examples, list of predictor attributes
2) *Output:* Detected faults
3) *I*nitialisePheromones(
4) . ComputeHeuristicInformation(
5) $\leftarrow \emptyset$
6) m $\leftarrow$ 0
7) while m < maximum iterations and not CheckConvergence() d
8) $.tree_{ib} \leftarrow$
9) . for $n \leftarrow 1$ to colony size do
10) $tree_n \leftarrow CreateTree(Examples, Attributes, -)$
11) $Prune(tree_n)$
12) if $Q(tree_{ib}) > Q(tree_{gb})$ then
13) . $tree_{ib} \leftarrow tree_n$
14) end if
15) end for
16) $UpdatePheromone(tree_{ib})$;
17) . if $Q(tree_{ib}) > Q(tree_{gb})$ then
18) . $tree_{gb} \leftarrow tree_{ib}$
19) end if
20) . $m \leftarrow m + 1$;

©IJRASET: All Rights are Reserved

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 6.887
Volume 6 Issue VII, July 2018- Available at www.ijraset.com

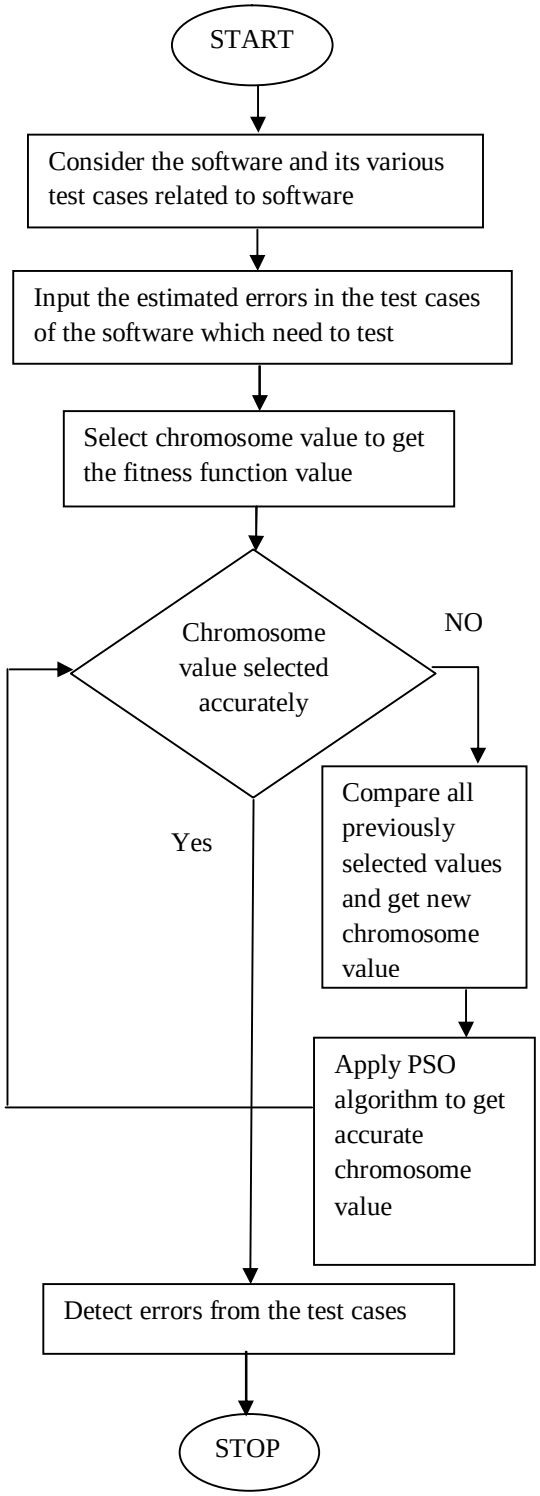


Fig 1: Proposed Flowchart

III. EXPERIMENTAL RESULTS

The proposed algorithm will schedule the testing's of agile testing for the detection of maximum faults from the software. The proposed algorithm is implemented in MATLAB by taking five type of testing's as input

©IJRASET: All Rights are Reserved

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 6.887
Volume 6 Issue VII, July 2018- Available at www.ijraset.com

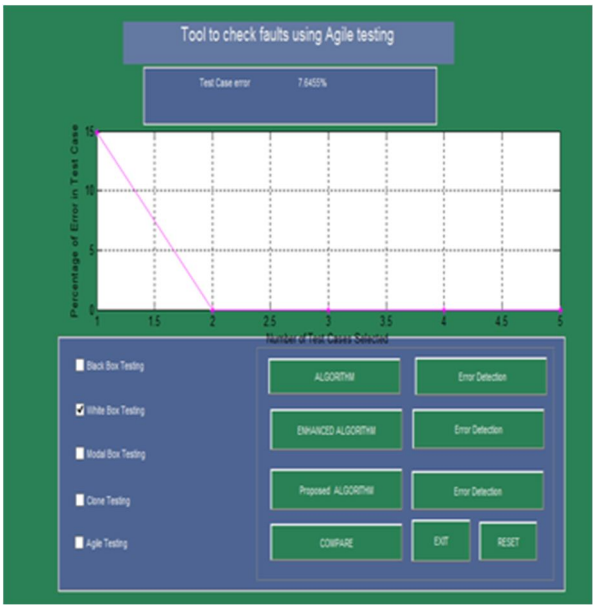


Fig 2: Display value of faults using white box testing with PSO

As shown in figure 2, the algorithm is PSO is applied on white box testing. The white box testing fault value is taken as input and on that value PSO algorithm is applied which will detect maximum number of faults from the project

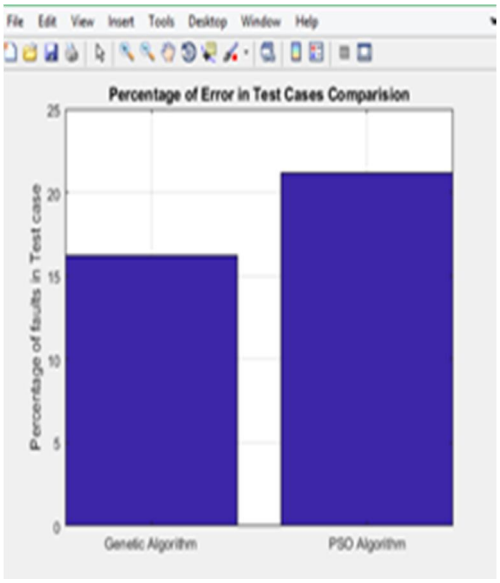


Fig 3: Comparison of results

As shown in figure 3, the results are compared of two algorithms, the existing algorithm is the genetic algorithm, proposed algorithm is PSO algorithm for the fault detection using agile testing. It is analyzed that proposed algorithm detect maximum faults as compared to existing technique

IV. CONCLUSION

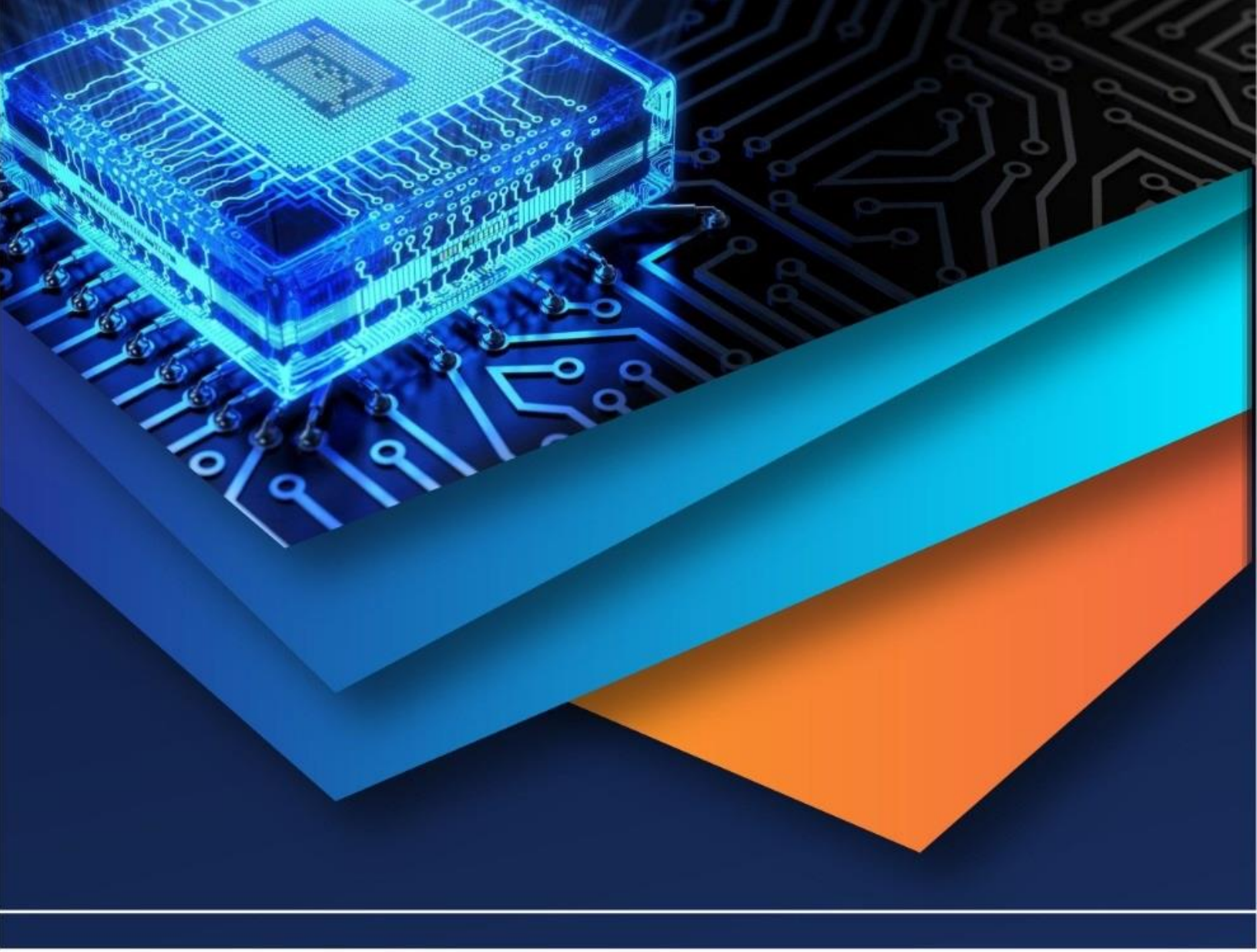
The agile testing is the type of testing in which various type of testing can be applied at different stages to test the software. These type of testing are white box, black box, model based etc. To detect maximum number of faults from the software, these testing needed to schedule in the proper manner. In work, technique of PSO technique is applied which will schedule the testing's of the agile testing. The proposed work is implemented in MATLAB and it is been analyzed that fault detection rate is increased after scheduling of testing.

©IJRASET: All Rights are Reserved

REFERENCES

[1] Finkelstein, B. Nuseibeh, L. Finkelstein and J. Huang," Technology transfer: software engineering and engineering design", 1992, COMPUTING & CONTROL ENGINEERING JOURNA
[2] Nazim H. Madhavji," The process cycle", 1991, Software Engineering Journa
[3] H. D. Mills," The management of software engineering Part I: Principles of software engineering", 1980, IBM SYST J, Vol 19, No
[4] Mary Shaw," Prospects for an Engineering Discipline of Software", 1990, IEEE, 0740-7459/90/1100/0015
[5] William Everett, Samuel Keene, Allen Nikora," Applying Software Reliability Engineering in the 1990s", 1998, IEEE TRANSACTIONS ON RELIABILITY, Vol. 47, No. 3-S
[6] VICTOR R. BASILI, RICHARD W. SELBY," Comparnng the Effectiveness of Software Testing Strategies", 1987, IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, VOL. SE-13, NO. 1
[7] Junpeng Lv, Hai Hu, Kai-Yuan Cai, and Tsong Yueh Chen," Adaptive and Random Partition Software Testing", 2014, IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS: SYSTEMS, VOL. 44, NO. 1
[8] Lipika Bose and Sanjeev Thakur, "GRAFT: Generic & Reusable Automation Framework for Agile Testing" , IEEE, 201
[9] Srinivas Nidhra and Jagruthi Dondeti, " Blackbox and Whitebox Testing techniques", International Journal of Embedded Systems and Applications (IJESA) Vol.2, No.2, June 201
[10] Chandana Bharati1, Shradha Verma, "Analysis of Different Regression Testing Approaches", International Journal of Advanced Research in Computer and Communication Engineering Vol. 2, Issue 5, May 201
[11] Xiaobo Han, Hongwei Zeng and Honghao gao, "A Heuristic Model Based Test Prioritization Method For Regression Testing" International Symposium on Computer,Consumer and Contro, 2012
[12] Junpeng Lv, Hai Hu, Kai-Yuan Cai, and Tsong Yueh Chen," Adaptive and Random Partition Software Testing", 2014, IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS: SYSTEMS, Vol. 44, N0. 1
[13] S´ebastien Bardin, Omar Chebaro, Micka¨el Delahaye, and Nikolai Kosmatov," An All-in-One Toolkit for Automated White-Box Testing", 2014, LNCS 8570, pp. 53–60

©IJRASET: All Rights are Reserved



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)