



Regression Testing in Green Cloud Based Software with the Aid of Hybrid PSO-CS Algorithm

¹Prakash Venkatraman and ²Goplakrishnan Sethumadhavan

¹Assistant Professor, Department of Computer Applications, SASTRA University, Thanjavur, Tamilnadu, India. E-mail:Prakash125@mca.sastra.edu

²Associate Dean, Department of Computer Applications, SASTRA University, Thanjavur, Tamilnadu, India. E-mail: headca@sastra.edu

Abstract

Green Cloud computing has as of late changed the methodology of acquiring computing assets, (for example, PCs, foundations, information stockpiling, and application administrations), alongside the methods for overseeing and conveying computing administrations, advances, and arrangements. Green Cloud computing drives an open door in contribution testing as assistance in clouds. In the interim, it contain a innovative problem, difficulties and needs in programming testing, specific in difficult clouds and cloud-based applications. Regression testing is utilized to test the adjusted variants of the product to verify that the altered qualities execute similarly and that the progressions didn't create sudden flaws, in addition known as disintegration mistakes. Based on the concept of software regression testing we propose an innovative technique in order to test the quality of the cloud based software. For this, we utilized the parameters such as dependability, reliability and maintainability as the quality metrics under evaluation. In our system initially testcases are obtained and are subjected to optimization process in order to refine the exact test suites. In our future framework we have utilized modified cuckoo search used to optimizing test cases that are necessary for the quality measurement. The quality metrics like dependability, reliability and maintainability are estimated.

Keywords: Green Cloud computing, Regression testing, cuckoo search , dependability, reliability and maintainability.

1 Introduction

A constant arrival of value items among improved functionalities is necessary used for production accomplishment in the product business. To discover whatever number shrouded blames as could be expected under the circumstances preceding a discharge, programming testing's have been performed at different formative stages. A framework difficult is one of the pivotal testing exercises since it is executed to ensure the objective framework's effective conduct including their functionalities, and to be the last check before a discharge [5]. The regression test aims to test the improvements made in the targets code and guarantees that the improvement does not bring any signs. One of the habits in which this can be guaranteed by re-running all the significant old experiments in old test suite document, which be that as it might, can be restrictively expensive. A superior methodology can be to distinguish the abused piece of the framework and pick experiments that mapped to altered and unaltered segments of the regulations [1]. By and large, regression testing advanced an isolated into three principle forms - test case minimization, test case decision and test case prioritization. Test case prioritization procedure considers requesting of test cases for location of issues at the soonest while test case minimization is done to clear out the overabundance test cases. Finally, test case assurance process, its point is to have the option to choose the detachment of test cases from the test group that can possibly recognize blunders started from a change [7].

It is unavoidable the usefulness of a product framework may change during programming improvement and upkeep. Each time the product is adjusted, regression testing is important to guarantee the nature of the product framework. That is, new test cases will be intended to guarantee the accuracy of the new capacities and the first test cases ought to likewise be re executed to ensure in majority of the unchanged capacities still work accurately [2]. Administration Composition gives numerous advantages that basic nuclear administrations can't. A case of this is administration arrangements can give the chance to organizations to have the option to create frameworks dependent on their necessities and requirements [6].

A test case is a lot of info information and predictable yield consequences which are intended to practice a particular programming capacity or test prerequisite. During testing, the basic programming framework will be executed to look at the related program to decide the rightness of a product work [3]. Among the launch of green cloud

computing, companies contain a lot of chances to shift a fresh computing and software permitting paradigms. One such representation that is in effect more and more utilized is Software as a Service (or SaaS). In a SaaS model companies give on-line services (applications) for their clients. The memberships can be arranged and modified by the client as their industry requirements modification [4].

2 Related work

Regression test determination is a procedure to pick a subset of existing test cases, which are trailed by used with a various original test cases for regression testing. Regression testing guarantees that the changes finished to the framework have not affected the present functionality. The ongoing years, there is no sufficient procedure which safeguard a relapse test assurance by contemplating changes in semantics of operations (alongside the other sentence structure and semantics changes) utilizing UML diagrams. Transform in semantics of a procedure alludes to the change in contingent articulations, change in free ways/extraordinary ways, change in charge stream and expansion or cancellation of any substance from the current usefulness. Dahiya et al. [8] have presented an approach which could do this utilizing class, arrangement and activity diagrams. The device looked at old and new types of UML diagrams to order test cases into reusable, retestable, outdated and recently produced classification. Activity diagrams are particularly used to analyze the semantics of tasks. The altered process relating to the action of diagrams were additionally search in class and gathering diagrams for relapse test determination.

Regression tests frequently bring numerous deviations (contrasts between two framework variants), either because of changes or regression faults. For the tester to investigate such deviations proficiently, it is useful to exactly gather them, with the end goal that each gathering contains deviations speaking to one remarkable change or regression fault. Because it is impossible that a general answer for the above issue can be discovered, we center our work around a typical kind of software framework: database applications. Rogstad and Briand [9] investigated the utilization of bunching, based on record manipulations and test condition (starting analysis models), to gather regression test deviations according to the faults or changes cause them. The proposed estimation criterion based on the idea of entropy to measure up to alternative grouping strategies. Among the four examination campaigns assess, deviations were grouped giftedly for two of them, while the other two groups were every homogenous.

Deciding how to pick a subset of test cases with high-fault discovery capacity transforms into a key inquiry in code-level relapse testing. Bunch investigation has been intended to course of action through this theme. It

allotments test cases into bunches dependent on the equal of implementation profiles. In past assessments, affecting profiles were addressed as binary or numeric vectors. The vector model just considers the occasions that a capacity or explanation is executed. In any case, it overlooks successive, the relations and auxiliary information between capacity calls. In this manner vector-based strategies don't generally create satisfying results. Wang et al. [10] have displayed bunch analysis of three distinct sorts of structural profiles, work execution succession, Function Call Grouping (FCS) and capacity call tree. They structured and directed experiential examinations on five medium-sized programs to approve the effects of different profiles on regression test case decline.

Location-Based Services (LBS) are generally sent. Exactly when the execution of a LBS-enabled service has propelled, regression testing can be used to guarantee the as of late settled practices not having been hurtfully influenced. Genuine test case prioritization uncovers tune abnormalities profitably so that fixes can be reserved past to confine the annoyance to service shoppers. A key viewing in areas caught in the wellsprings of information and the ordinary yields of test cases are truly associated by the LBS-enabled service, and these military heuristically utilize expected and unsure areas for their calculations, making these services will when all is said in done treat areas in proximity homogenously. Zhai et al. [11] have proposed a gathering of estimations and instates them to demonstrate input-guided systems and Point of Interest (POI) ready test case prioritization methods, fluctuating whether the situation in succession in the ordinary yields of test cases was used. It reports a case study on a stateful LBS-enabled service. The case learning demonstrated that the POI-aware strategies could be more effective and steadier than the gauge, which reorders test cases randomly, and the information guided techniques.

To guarantee that a changed software framework has not relapsed, one approach is to repeat offered test case. Be that as it may, this is a potentially costly task. To moderate the costs, the testing exertion can be enhanced by executing only a picked division of the experiments that are accepted to have an unrivaled chance of supportive faults. Mirarab et al. [12] have planned a move toward for choosing and requesting a foreordained numeral of experiments from a present test suite. Their approach frames an Integer Linear Programming issue utilizing two diverse coverage-based criteria, and utilizations constraint relaxation to find many near optimal arrangement centers. These centers are then joined to obtain a final arrangement utilizing a democratic mechanism. They picked subset of experiments was at that point organized utilizing an avaricious algorithm that maximizes least coverage in an iterative behavior.

3 Proposed Methodology for Regression Testing in Cloud Based Software

During Software configuration process the regression testing is applied to the personalized version of software so the altered area continues as before according to the necessity without delivering any kind of blunders. The regression testing is also necessary in the cloud based software system. The planned process we have designed a capable technique in order to trial the software in cloud environment. The proposed technique of regression testing is demonstrated in the beneath figure 1.

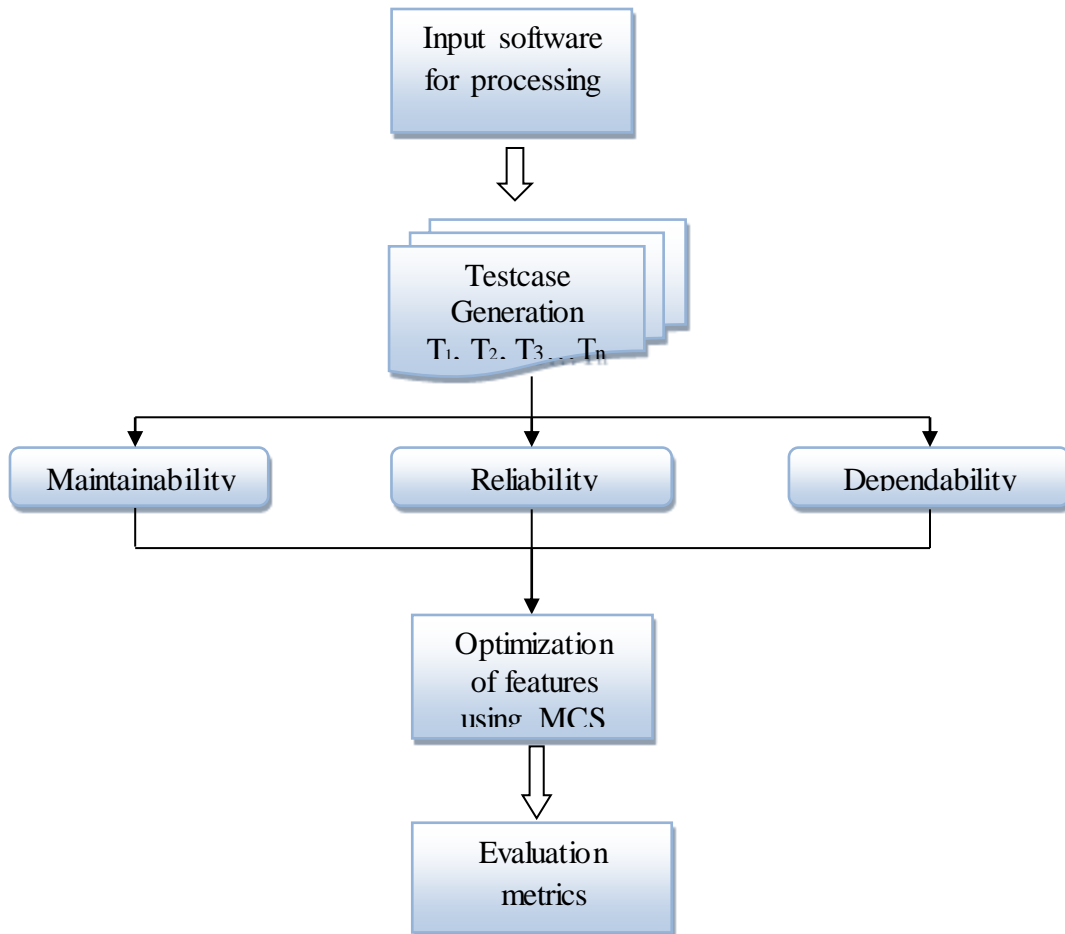


Fig. 1 Flow diagram of proposed system.

3.1 Test Case Generation

Test cases are utilized to examine all possible combinations in the function and as well it offers the client to just repeat the means that were unspecified to uncover a deformity that as distinguished during test. It can be charted legitimately and obtained from use cases. Besides, when the test cases are delivered early, Software Engineers can as often as possible find ambiguities and irregularities in the necessities requirement and configuration records. The generated test cases will be nourished to the difficult neural system for organization based on which the software dependability will be anticipated.

3.2 Software Quality Metrics

Software quality measure is to approximate value of the software. At this time we used to calculate the software quality by reliability.

3.2.1 Maintainability

Viability of the software is a method by which software can be changed and it is regarded to the most important software excellence element. While planning the software correspondence, high steadinesses with improved confidence are the major skin textures that are regarded. Individual's software parcels are regarded to be the enhanced maintainable software's. By discerning the distinctive structure quality measurements of software, the reliance of software can be calculated which in addition calculates approximately the maintainability of that software. The Abstractness, insecurity and efferent combination are the three significant value measurements in the software.

The maintainability of the software is a method it can be adapted and regarded to the most important software assessment in an element. The correspondence and parcels with efferent mixture ($C_e=0$) and insecurity ($I=0$) are assembled under the reliable packages. The trustworthy and the Non-Dependable gatherings in any software are in charge of reducing the maintainability of the software. Utilizing the articulation determined underneath the dynamics for the software is figured,

$$A = N_A / N_C \quad (1)$$

Where,

N_A = entirety numeral theoretical class in the function,

N_C = entire numeral module in the relevance

Similarly the insecurity is processed by methods for the underneath enunciation,

$$Instability(I_n) = \frac{c_e}{(c_e + c_t)} \quad (2)$$

Where, c_e =Efferent Coupling, c_t = Total Coupling

3.2.2 Reliability

In a particular environment, software dependability is described by the possibility of sans failure software operation for a given timeframe. The excellence of software testing is openly associated to reliability growth. During the software designing the mistakes are mainly generated and these mistakes prove the main motive for finding out the reliability of software. The quantity of mistake in the software should be designed something like correctly and must be eliminated. The unwavering quality can be determined by assessing the testing exertion of the particular software. The failure rate with respect to the time of execution can be figured and this presents as the unwavering quality of that particular software at the usage time. The software unwavering quality is assessed dependent on the failure rate that is gotten from the software.

The consistency of the software preserve determined through the appearance specified below,

$$reliability = \frac{failure\ rate(f_r)}{execution\ time} \quad (3)$$

3.2.3 Dependability

Dependability is regularly the majority important framework belongings of a serious framework. It characterizes the degree to which a significant framework is trusted by its clients. Dependability mirrors the degree of the client's certainty that it won't fail in ordinary process. The above parameters are measured and are optimized

3.3 Optimization using Hybrid PSO- CS Algorithm

In the PSO between the possible arrangement, the random velocity of a particle is assigned. Each particle continues to interrupt the problem of the best arrangement of its coordinates. Here the value of health is measured for

further process. This health value is referred to as *pbest*. The location of these arrangements is considered *gbest*. We have used a modified form of PSO in our planning system. In this PSO we include incorporated MCS algorithm in the updation phase of PSO which would additional increasing the plausibility of picking the best particle.

The various advances required for actualizing the PSO is explained underneath,

- i. Initially with the location and rate of the particles (solutions) choose the *n*-variable for the interval of difficulty.
- ii. Each of these unintentionally generates the optimization strength work on particle estimation *n*-variables.
- iii. Equate this conditional value with the *pbest* value at this time. If this existing condition is enhance the *pbest* then choose the recent condition assessment as the *pbest* for the additional dispensation.
- iv. These strength ethics is compared through the general greatest preceding values and this condition the present value is improved after that inform the *gbest* for the recent particles selection indicator and price as the new *gbest*.

Modify the rate and the particle location, and then repeat the steps until you get the magnitude of the force.

In updating phase, we have modified the traditional cuckoo search algorithm. The crane search algorithm speaks to a meta-heuristic algorithm that owes its origin to the reproductive behavior of the crane and is easy to implement. There are plenty of houses in search of the crane. Each egg represents a response and the crane compares an egg to a New Testament. The arrangement of story and practice changes the most unpleasant arrangement at home. We have adjusted the conventional crane search algorithm by combining the Gauss distribution in the updating phase, where the charge plane position is used. The best effects of expansion are included when the Gaussian distribution is contrary to usual practice.

The usual methodology of the bunching system is appeared as pursues:

Stage 1: Initialization Phase

The populace (m_i , where $i=1, 2, n$) of crowd home is started self-assertively.

Stage 2: Create New Cuckoo Phase

A crane is chosen arbitrarily with the help of customs, which creates new arrangements. In this way, the induced cuckoo is assessed by utilizing the target work for determining the greatness of the arrangements.

Stage 3: Fitness Estimate Phase

The wellness capacity is assessed as per Equations 4 and 5 demonstrated hereunder, trailed by the choice of the best one.

$$P_{\max} = \frac{P_S}{P_T} \quad (4)$$

$$\text{fitness} = \text{maximum popularity} = P_{\max} \quad (5)$$

Where,

P_S - connotes the chose populace

P_T - speaks to the all out populace

Stage 4: Updation Phase

Initially, this provision is enhanced by duty aircraft by using the cosine change. The nature of the novel arrangement is evaluated and a house is chosen from them. In the possibility that the past arrangement in the chosen home is better than the past arrangement, it is replaced by the novel arrangement (crane). If anything else, the past arrangement is considered the best arrangement. The duty planes used for the Common Crane Search algorithm are connected by Equation 6, which proves the following:

$$m_i^* = m_i^{(t+1)} = m_i^{(t)} + \alpha \oplus \text{Levy}(n) \quad (6)$$

By appropriately adjusting Equation 6, we inform the arrangement and the additional handling is conveyed according to the above advances.

Stage 5: Reject Worst Nest Phase

In this segment, the most exceedingly awful homes are disregarded, as per their plausibility esteems and novel ones are developed. In this manner, contingent on their wellness work the best arrangements are positioned. From that point, the best arrangements are distinguished and set apart as ideal arrangements.

Stage 6: Stopping Principle Phase

Till the accomplishment of the greatest emphasis, the method is preceded.

By precisely utilizing the previously mentioned characterization method, we have had the option to accomplish excellent order of hubs through which the issue free communication of information can be performed.

4 Result and discussion

In our planned technique an innovative technique in order to test the quality of the cloud based software is employed. Our planned strategy was executed by utilizing JAVA. Following test case creation, the measurements principles are designed. Test cases are organized using a hybrid PSO-CS algorithm based on the principles of measurement. The outcomes obtained from the project strategy are appeared in the underneath. The parameters like code coverage, branch coverage and time are estimated for different iterations. The obtained results are given in the table 1 under,

Table 1 Coverage values for testcases

Test Cases	Code Coverage	Branch Coverage	Time
1	0.263	0.321	0.01
2	0.168	0.221	0.02
3	0.325	0.032	0.03
4	0.158	0.136	0.01
5	0.147	0.214	0.04

Our planned Hybrid PSO-CS method is compared through existing algorithms like CS and PSO. For different iterations the respective convergence time are noted. The project strategy meets with less time compared to the concession method, i.e., the planned process receives an earlier arrangement similar to the existing structure. The obtained results are given in table 2 below,

Table 2 Convergence time for proposed Hybrid PSO-CS algorithm and existing CS and PSO algorithm

Iterations	Convergence Time		
	Proposed PSO- CS Algorithm	CS	PSO Algorithm
10	10	10	10
20	18	20	20
30	18	30	30
40	18	35	39
50	18	35	39

The fig 2 given beneath demonstrates the graphical demonstrate for comparison of convergence time with respect to different iterations. The convergence value for proposed PSO-CS and existing algorithms are graphically compared and since the diagram is evident to facilitate our future scheme delivers superior convergence rate than existing algorithms.

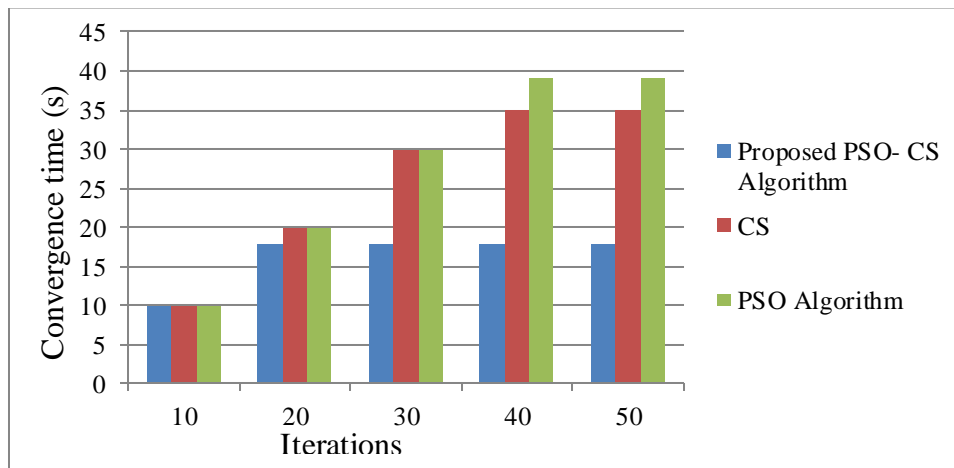


Fig 2 Graphical representation for convergence time using proposed and existing algorithms.

The table 3 specified under shows the different quality parameter principles obtained with and without optimization for different iteration. The maintainability, dependability and reliability are the quality parameters that we measured in our proposed scheme. The outcome shows to facilitate our planned system has improved quality values when compared with those where optimization is not used.

Table 3 Quality parameters with respect to iterations

Iteration	Maintainability		Dependability		Reliability	
	With optimization	Without Optimization	With optimization	Without Optimization	With optimization	Without Optimization
10	0.854	0.811	0.654	0.598	0.521	0.512
20	0.854	0.811	0.654	0.598	0.521	0.514
30	0.854	0.811	0.654	0.598	0.521	0.52
40	0.796	0.712	0.621	0.594	0.521	0.52
50	0.725	0.712	0.608	0.594	0.501	0.522

The graphical representation for the quality measure like maintainability, dependability and reliability are shown in fig 3, fig 4 and fig 5 respectively. For different iterations the respective quality parameter is calculated with and without optimization. Since the figures it is comprehensible that our future structure of regression testing in cloud software has better quality measures when compared with that of system where the optimization techniques are not utilized.

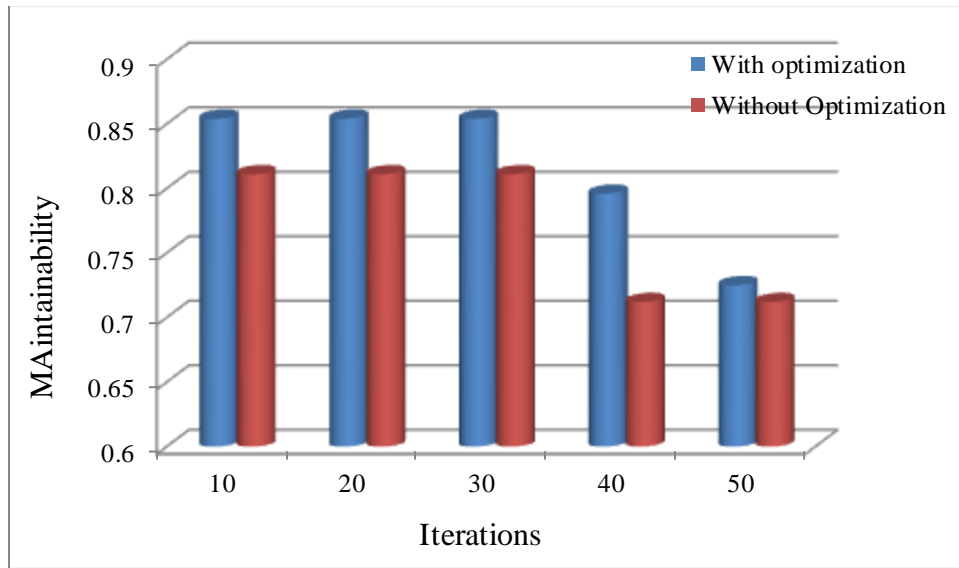


Fig 3 Maintainability for various iterations with and without optimization

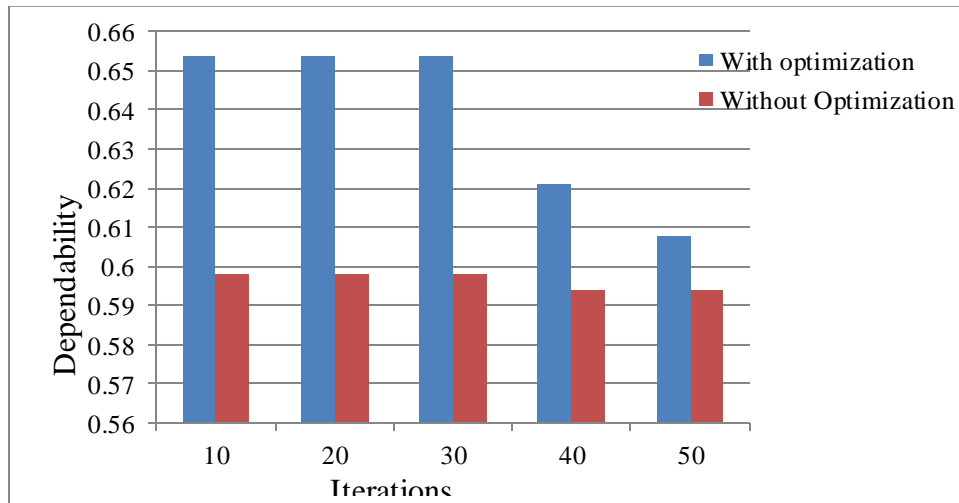


Fig 4 Dependability for various iterations with and without optimization

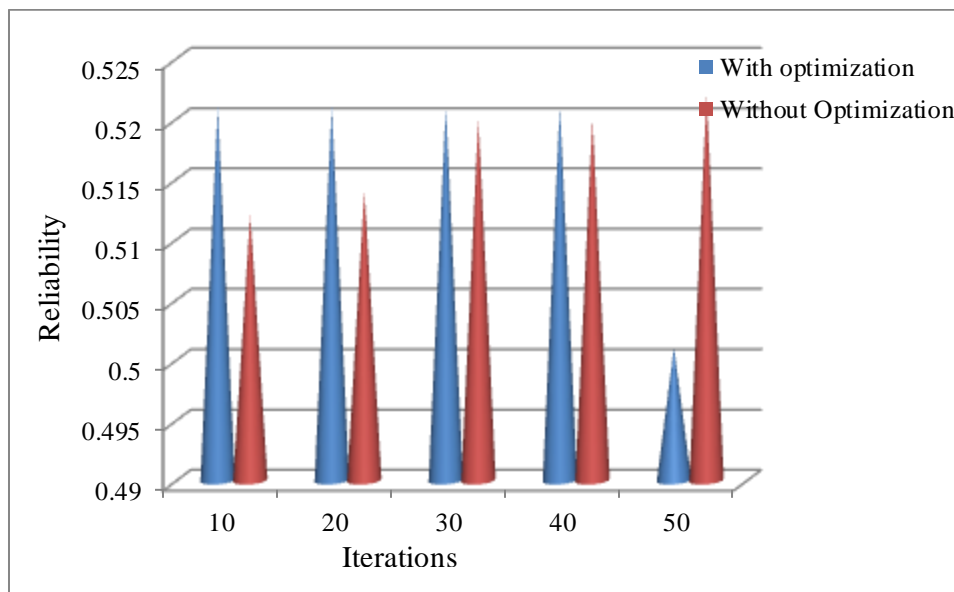


Fig 5 Reliability for various iterations with and without optimization

5 Conclusion

Regression testing in green cloud based software has emerged to motivate investigate theme in current years due to the wide spread usage of cloud system. The proposed system of regression testing in cloud software uses soft computing techniques which improves the quality of the software under testing. In our proposed scheme we have used hybrid particle swarm optimization algorithm where in we have incorporated cuckoo search algorithm for enhancing the optimization process. The results we obtained show to facilitate our future technique has obtained better outcome of convergence time. Also the quality parameters like maintainability, dependability and reliability has obtained better value when compared with optimization and without optimization techniques.

Reference

- [1] Animesh Chaturvedi , Atul Gupta, "A Tool Supported Approach to Perform Efficient Regression Testing of Web Services", IEEE 7th International Symposium on the Maintenance and Evolution of Service-Oriented and Cloud-Based Systems, 2013.

- [2] Chu-Ti Lin, Cheng-Ding Chen, Chang-Shi Tsai ,Gregory M. Kapfhammer, "History-based Test Case Prioritization with Software Version Awareness", 18th International Conference on Engineering of Complex Computer Systems,2013.
- [3] Chu-Ti Lin, Kai-Wei Tang, Cheng-Ding Chen, Gregory M. Kapfhammer, "Reducing the Cost of Regression Testing by Identifying Irreplaceable Test Cases", Sixth International Conference on Genetic and Evolutionary Computing, 2012.
- [4] Hema Srikanth , Myra B. Cohen," Regression Testing in software as a service: An industrial case study", 27th IEEE International Conference on Software Maintenance (ICSM) ,2011.
- [5] Hirohisa Aman, Yuta Tanaka, Takashi Nakano, Hideto Ogasawara , Minoru Kawahara, "Application of Mahalanobis-Taguchi Method and 0-1 Programming Method to Cost-Effective Regression Testing", 42th Euromicro Conference on Software Engineering and Advanced Applications (SEAA),2016.
- [6] Paul Buck, Qi Shi , Aine MacDermott, "A Selective Regression Testing Approach for Composite Web Services", International Conference on Developments of E-Systems Engineering (DeSE), 2015.
- [7] Benjawan Srisura , Adtha Lawanna, "False Test Case Selection : Improvement of Regression Testing Approach", 13th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON) , 2016.
- [8] Sumit Dahiya, Rajesh K. Bhatia , Dhavleesh Rattan, "Regression test selection using class, sequence and activity diagrams", IET Software , Vol.10, No. 3, pp.72-80, 2016.
- [9] Erik Rogstad , Lionel C. Briand, "Clustering Deviations for Black Box Regression Testing of Database Applications", IEEE Transactions on Reliability, Vol.65,No.1,pp.4-18,2016.
- [10]Rongcun Wang, Binbin Qu , Yansheng Lu, "Empirical study of the effects of different profiles on regression test case reduction", IET Software,Vol.9,No.2,pp.29-38,2015.
- [11]Ke Zhai, Bo Jiang, W.K. Chan ,"Prioritizing Test Cases for Regression Testing of Location-Based Services: Metrics, Techniques, and Case Study", IEEE Transactions On Services Computing, Vol. 7, No. 1,pp.54-67,2014.
- [12]Siavash Mirarab, Soroush Akhlaghi, Ladan Tahvildari, "Size-Constrained Regression Test Case Selection Using Multicriteria Optimization", IEEE Transactions On Software Engineering, Vol. 38, No. 4,pp.936-956, 2012.

Biographies



Prakash Venkatraman obtained his Bachelor's degree in Computer Science from Bharathidasan University of Tamilnadu. Then he obtained his Master's degree in Computer Applications from the same University. He has also obtained Master of Technology in computer Science from SASTRA University, Tamilnadu. Currently, he is an Assistant Professor in the Department of Computer Applications SASTRA University His specializations include Software Engineering, Object oriented Analysis and Design His current research interests are Software Testing, Cloud Computing.



Dr. Gopalkrishnan Sethumadhavan received the B.Sc. degree from the Bharathidasan University of Tamilnadu., he had received his M..Sc. degree from the same university. He has received his doctorate from SASTRA University. He is currently an Associate Dean in the Department of Computer Applications at SASTRA University. His research interests are in Digital Image processing and Software Engineering. He has got more than 30 years of Teaching and Research experience.