



Optimal Test Case Prioritization in Cloud based Regression Testing with Aid of KFCM

Sunitha Badanahatti^{1*}

Yelisetty Satya Sree Rama Murthy¹

¹Mewar University, Rajasthan, India

* Corresponding author's Email: sunithab0516@gmail.com

Abstract: Regression testing is a kind of software testing that authenticates that software previously developed and established still accomplishes correctly after it was altered or interfaced with other software. The aim of the investigation is to progress the software competence via cloud based regression testing. The projected technique has three chief stages such as, 1) test case generation, 2) clustering and 3) test case prioritization. Primarily the input implementation is send to the cloud for test case generation. For producing the test cases the projected method utilizes coverage metrics. Following the test case generation, the existing test cases are gathered with the help of kernel fuzzy c means clustering algorithm. Subsequently each group is fed to test case prioritization; in that grey wolf optimization algorithm is utilized for test case prioritization. Thus we will get operative prioritized test cases. Our method will be implemented on JAVA with Cloud Sim platform. The presentation will be assessed using execution time and memory use. From the solution the applied method precedes minimum implementation time associated with the available technique.

Keywords: Regression testing, Test case generation, Kernel fuzzy c means clustering, Grey wolf optimization, Test case prioritization.

1. Introduction

Software testing is a complete group of activities accompanied with the determined of verdict faults in software. It is one action in the software development procedure intended at assessing a software item, namely subsystem, system, and features against a provided group of scheme necessities. Also, software testing is the procedure of validating and confirming that a program performances appropriately [1]. Regression testing is a kind of software testing. It is extensively utilized for testing an adapted software implementation to form whether it performs as per stipulations [2]. The chief attitude in the regression testing is to give assurance that the newly reputable vicissitudes will not interrupt the behavior of the available, unaffected portion of the software [3]. Traditional methods to software testing have become expensive in the name of cost, effort, and overheads. Hence forth testing practices have shifted gears to approaches that would help the organizations in the

name of business and profits. Leading corporations namely IBM, Google, Microsoft, and Amazon have a conferred attention in the biz word "CLOUD" [4]. Cloud computing not only takes fresh business chances but also originates to a few main influences on software testing and conservation. A main influence is called as "Cloud Testing" [5].

With cloud-based testing, administrations no longer essential to worry about detecting servers, obtaining licenses for programs and testing tools and connecting them. Service providers provide testers access to scalable and ready-to-use virtual labs with a library of operating schemes, test management and implementation tools, middleware and storage necessary for generating a test atmosphere that thoroughly mirrors the real atmosphere [6]. At the time, Regression testing fresh test cases is not fashioned but formerly created test cases are re-executed. The humblest method is to run all test cases for confirming the adapted program. This is the safest method, but it is applied only if the size of test suite is minor [7]. The adapted portions of the

system are primarily tested. Then the entire scheme desires to be retested with the help of the old test suite to have confidence that the alterations did not familiarize novel errors into the scheme [8]. Two main portion of Regression Testing are Test case assortment and Test case prioritization. Test case assortment resolves the issue of picking the test case that will be valuable to test the adapted portion of the software. Test case arrangement apprehensions ordering the test cases that exploit the attractive possessions of current fault recognition [3].

Regression test case assortment is a method that is proposed to decrease time and resource consumption in regression testing [9]. The issue of regression test case assortment is resolved by prioritizing test cases [10]. As the test cases produce in number because of the development of implementations, it becomes very challenging to perform all the test cases within an identified amount of time. To solve the issue of higher test execution time, strategies have been proposed for prioritizing test cases [11]. Test case prioritization is important in regression testing. It schedules the test cases in a regression test suite with the aspect for exploiting certain objectives, which help decrease the time and cost obligatory to preserve service-oriented business implementations [12]. Test case prioritization methods aim to recover the efficiency of regression testing by organization the test cases so that the most advantageous are implemented primarily [13]. This methods schedule test cases in an order that upsurges their efficiency at meeting some functional goals [14]. And also it schedules test cases for regression testing in an order that efforts to make the most of some objective performance [15]. The overall objective of the suggested technique is to increase the software competence through cloud based regression testing. In order to attain the objective function effective clustering and optimization technique for test cases prioritization is examined in our proposed technique. The main advantage of the recommended technique attains minimum execution time and memory space when compared to the existing technique.

The remainder of the paper is organized as follows: Section 2 provides an overview of related work in the field. The present problems are identified in section 3. In Section 4 narrates the testing model where the various stages of the work are explored. Section 5 brings out the experimental results and discussions.

2. Related Works

Numerous methods were projected using

numerous authors for Regression Testing and an inadequate of them is elucidated below:

E. Proko and I. Ninka [16] have engrossed mostly on investigation presentation metrics in Cloud Computing, on the basis of their features namely scalability, elasticity, pay-per-use and error tolerance, and then they have deliberated why necessary fresh approaches for performance testing in Cloud Computing and creation of fresh benchmarks. From that analysis they have decided that the performance testing and assessment should be achieved with the help of novel mode ls testing that were created rendering to Cloud Computing features and metrics.

A method for selecting and ordering a predetermined number of test cases from an available test suite was given by S. Mirarab et al [17]. Their method has formed an Integer Linear Programming issue with the help of two dissimilar coverage-based criteria, and has utilized constraint relaxation to detect numerous close-to-optimal result points. Those points were then joined to attain a final result using a voting mechanism. The designated subset of test cases was then prioritized with the help of a greedy algorithm that exploits minimum coverage in an iterative manner. Their projected method has been assessed and the solutions display important enhancements over existing method for some cases and analogous solutions for the rest. Their method has also delivered more consistency associated with available methods.

P. K. Bhowate and V. B. Gadicha [18] have appraisals for migrating testing to the cloud. Software testing in the cloud could decrease the necessity for hardware and software resources and deliver a supple and effectual substitute to the outdated software testing procedure. A main problem to the extensive usage of testing in the cloud was associated with security problems. Cloud computing clues an chance in proposing testing as a service (TaaS) for clouds. In the meantime, it origins novel problem and trials in software testing that are specific in testing clouds and cloud-based implementations.

The notion of uniting multiple criteria into a hybrid was elucidated by S. Sampath et al [19]. Their aim was to generate a uniform demonstration of such amalgamations so that they could be designated unmistakably and collective among investigators. They have exactly articulated three hybrid amalgamations, Merge, Rank, and Choice, and established their practicality in two conducts. Their conclusions have recommended that hybrid criteria of others could be designated by their Merge

and Rank preparations, and that the hybrid principles they industrialized most frequently outstripped their essential discrete principles. Lastly, they have predicted that the outline gives a step toward serving investigators to generate shared tools and artifacts that usage a uniform illustration.

X. Xu et al [20] have projected a novel type of model, termed the User Representation Model Graph (URMG) that was built on the basis of CBMG. It utilizes an algorithm to improve CBMG and enhances the assessments execution procedure. On the basis this model, an involuntary testing and assessment scheme for web implementations was designed, applied, and organized in their test cloud that was able to perform all of the investigation and testing operations by only web access logs. In our scheme, the error rate instigated by arbitrary access to implementations in the implementation stage was also abridged, and the solutions have revealed that the error rate of the assessment that on the basis of URMG was 50% less than that on the basis of CBMG.

A suite of metrics and primed them to establish input-guided methods and point-of-interest (POI) aware test case prioritizing methods, opposing by whether the location data in the predictable outputs of test cases was utilized K. Zhai et al [21]. It was described a case study on a state full LBS-enabled service. The case study has displayed that the POI-aware methods could be more operative and steadier than the baseline that reorders test cases arbitrarily, and the input-guided methods. They have also established that one among the POI-aware procedures, *cist*, was moreover the most operative or the second most actual method among all the deliberate methods in our assessed aspects, although no method excels in all deliberate SOA fault classes.

A. Shrivastva et al [22] have delivered us with numerous cloud testing methods, challenges, problems and benefits in testing areas. It also particularizes all the important concepts concerning features and necessities in cloud testing. Besides, numerous cloud testing platforms were also deliberated briefly. Cloud testing has been elucidated extensively in that article that would help to comprehend numerous characteristics of cloud testing in a much better manner.

From the literature survey, they mainly focused the test case prioritization using greedy algorithm. The main drawback of this algorithm is hard to find the right approach. A major obstacle to the wider use of testing in the cloud is produced security issues. So that the suggested technique is using the effective clustering and optimal prioritization technique for secure cloud based regression testing.

3. Problem Definition

Regression testing is distinct as the process of test again the alteration portions of the software and confirm that no novel errors have been familiarized into already tested code. The common problems in existing regression testing test case prioritization approaches are provided below.

- ❖ Though emerging software, the issues that most of the developers face is the cost and complication of purchasing, positioning and upholding the fundamental hardware/software layers.
- ❖ Main problems that requirement to be addressed for testing cloud services in regression testing; there is no perfect testing methodology to legalize cloud services in contradiction of safety policy. Deficiency of comprehensive usability testing criterion further intensifies the issue.
- ❖ Software testing becomes more problematic because of the expansion and the difficulty of the scheme. In specific, it is too challenging to test parallel and dispersed schemes adequately although dependable schemes.
- ❖ When a failure happens in parallel and distributed schemes, the reproducibility of the actual scheme is so deprived that the discovery of the defective portion has been sober issue.
- ❖ Testing software needs sufficient resources and budget to comprehensive it effectively. But most among the organizations face the trials to give enough resources to test their software in disseminated atmosphere, with dissimilar loading level. This clue to simple issue when the software arranged into dissimilar client atmosphere and fluctuating user load.

These are the chief disadvantages of numerous available works that inspire us to do this investigation on Regression testing on cloud.

4. Proposed Method

Regression testing is a kind of software testing that confirms that software formerly developed and verified still achieves appropriately after it was altered or interfaced with other software. Throughout regression testing, novel software bugs or regressions may be exposed. Occasionally a software alteration impact investigation is achieved to regulate what areas could be exaggerated using the projected changes. The aim of the investigation is to increase the software competence through

cloud based regression testing. Originally the input presentation is direct to the cloud for test case generation. The test case generation is one among the significant stages utilized to detect the most significant test cases in the study. In this investigation we have to cluster the pertinent and irrelevant test cases in cloud. So that the projected technique usage kernel fuzzy c means (KFCM) clustering algorithm for grouping the pertinent and immaterial test cases. In order to recover the efficiency of testing, the relevant test cases are provided to the cloud for test case prioritization. The test case prioritization issue has lately complicated in preparation test cases for regression testing in an order that upsurges their efficiency of presentation aim. Since of that, test case prioritization is desirable in our projected work. For test case prioritization we have to enhance the test cases so the projected technique usage Grey wolf optimization (GWO) algorithm for test case prioritization. Thus we will attain operative prioritized test cases.

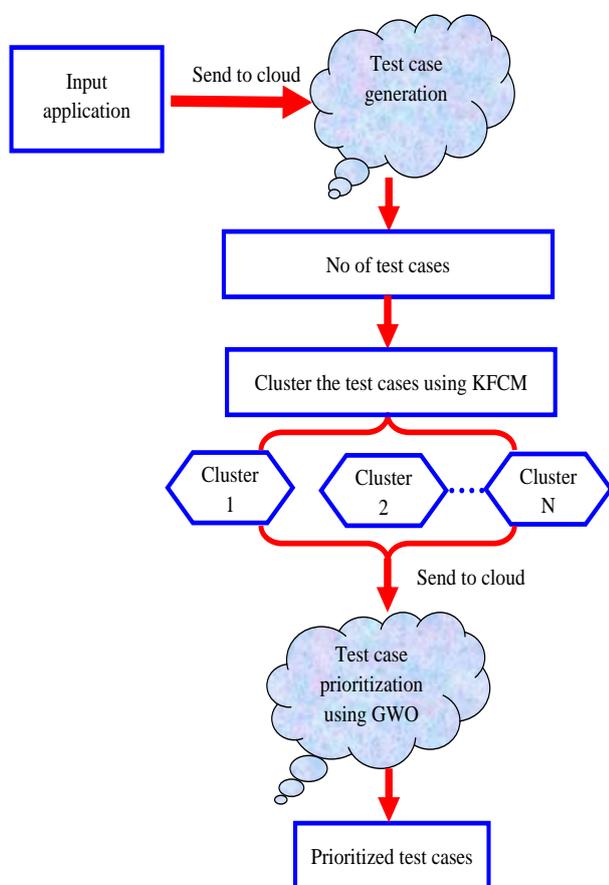


Figure.1 The overall block diagram of the proposed method

The aspect progression of our projected system is demonstrated in further segment and the global block diagram is revealed in fig.1,

Our proposed method has three stages namely,

1. Test case generation
2. Cluster the test cases using KFCM
3. Test case prioritization using GWO

4.1 Test Case Generation

Primarily, the submission that we are examination, receipts as an input for test case generation. Each implementation has the number of performance that is engaged for the generation of test case. The suggested technique mainly focuses on the performances and coverage metrics of the implementation that we are applied for the test case generation. In our projected method we are considering the coverage metrics are loop coverage, statement coverage, line coverage and comment coverage. On the basis of the feature value the applied technique produces the test cases.

Statement Coverage

Statement coverage is a metric that is utilized to estimate and measure the number of statements in the source code that have been implemented.

Line coverage

Line coverage is as well recognized as the statement coverage or segment coverage. Only correct circumstances are enfolded using line coverage. It as well events the excellence of the code and makes sure the flow of dissimilar path in that code.

Loop coverage

These coverage metrics hear says whether each loop body is applied zero times, exactly once and more than once. This metrics reports whether loop body is applied exactly once and more than once for do-while loops. And also, while-loops and for-loops achieve more than once. This information is not accounted using other coverage metrics.

Comment coverage

Comment coverage is a metric that is utilized to compute and measure the number of comments line in the source code that have been implemented.

On the basis of the above metrics we appraise the test cases for input implementation. We are taking the input performance is hospitality, banking, and library management implementation. Banking implementation comprises nearly 50 classes and 92 functions and hospitality management application comprises nearly 30 classes and 120 functions. Library management comprises nearly 22 classes

and 63 functions. Entire number of test case generation in banking, hospitality and library implementation is around 38, 115 and 15 test cases for each.

The sample test cases for the input application is provided below,

- [SC: 27.0, LC: 0.0, LPC: 30.0, CC: 0.0]
- [SC: 2.0, LC: 0.0, LPC: 71.0, CC: 0.0]
- [SC: 9.0, LC: 0.0, LPC: 99.0, CC: 1.0]
- [SC: 0.0, LC: 0.0, LPC: 61.0, CC: 0.0]
- [SC: 0.0, LC: 0.0, LPC: 40.0, CC: 0.0]
- [SC: 6.0, LC: 0.0, LPC: 158.0, CC: 0.0]

Then the existing test cases are grouped by clustering algorithm. In our projected method kernel fuzzy c means clustering algorithm is engaged for grouping the existing test cases on the basis of the resemblance of coverage metrics. The evident explanation of kernel fuzzy c means clustering algorithm displayed in underneath segment,

4.2 Kernel Fuzzy C-Means Clustering Algorithm

Fuzzy c-means (FCM) is a process of clustering that permits data focuses to group in the aspect of closeness and basically used in pattern acknowledgement. The change of objective function by minimizing as takes after

$$OF = \sum_{i=1}^n \sum_{c=1}^C M_{ic}^m \|T_i - C_c\|^2 \tag{1}$$

Where m any is real number sophisticated than 1, M_{ic}^m is the degree of membership of T_i in the cluster, c is the d-dimensional measured data, C_c is the d - dimension centre of the cluster. Fuzzy classification is completed with the help of objective performance iteration as seemed above, with the update of membership M_{ic}^m and the cluster centres C_c . Iteration of this procedure will be ended when there is end condition somewhere around 0 and 1. This KFCM procedure deliberates the input as test cases and output as clustered document on the basis of the test cases resemblance of coverage metrics.

Step by Step procedure of KFCM

Step 1: Kernel version of the FCM algorithm and its objective function are given bellow:

$$OF = \sum_{i=1}^n \sum_{c=1}^C M_{ic}^m (1 - K(T_i - C_c)) \tag{2}$$

Step 2: Compute the fuzzy centers C_j

$$C_c = \frac{\sum_{i=1}^n M_{ic}^m K(T_i, C_c) T_c}{\sum_{i=1}^n M_{ic}^m K(T_i, C_c)} \tag{3}$$

Step 3: Calculate the fuzzy membership function M_{ic} using

$$M_{ic} = \frac{(1 - K(T_i, C_c))^{-1/(m-1)}}{\sum_{c=1}^C (1 - K(T_i, C_c))^{-1/(m-1)}} \tag{4}$$

Step 4: Repeat step 2 and 3 until the maximum of OB_m value is achieved

Recognize the vital conditions for decreasing OF is revise equation (3) and (4) only if the kernel function K is designated to be the linear function,

$$K(T_i, C_c) = T_i^T C_c + S \tag{5}$$

Where, S is the constant value. Dissimilar kernels can be designated by substituting the Euclidean distance for diverse environments. Nonetheless, a linear kernel is appropriate for clustering in that it can fundamentally make the indispensable conditions. Lastly all the test cases are gathered with their resemblance of coverage metrics. Finally, the test cases form each cluster is provided to the cloud for prioritization.

4.3 Test Case Prioritization

In recommended technique grey wolf optimization method is utilized for prioritization. The stage by stage process of grey wolf optimization method is designated in additional segment,

Grey Wolf Optimization

The grey wolves efficiently encircle a Canidae's section ancestors and are appreciated as the head predators inspecting their arrangement at the sustenance's food series. They generally exemplify a bias to express appropriate as an assembly. The leader signify a male and a female, noticeable as alpha that are for the popular division in allegation of charming proper variety screening diverse features, for exemplar, sleeping location, the hunting, time to wake, and so forth. The assortments equipped using the alpha is acknowledged on to the assembly. The Beta addresses to the second grade in the striking array of the grey wolves. They are, basically, secondary wolves that effectively suggest a few backing to the alpha in the variety constructing or corresponding assembly utility. The omega is the least division of the grey wolf pack and

huge mission as a replacement present into the additional foremost wolves very almost on every occasion and is permitted to include just the diminutive leftovers charming subsequent a grand feast using the leader wolves. A wolf is designated as secondary or as delta each so regularly in the juncture that it doesn't fit in amongst the audience of an alpha, beta, or omega. Because of the truth that these delta wolves need to admiration the alphas and betas, they have a prosperous high proportion over the omegas. In our technique, the alpha (α) is appreciated as the most suitable collection using an outlook to duplicating sensibly the community pecking classify of wolves whereas visualizing the GWO. Consequently, the second and the third most outstanding provision are beta (β) and delta (δ) self-sufficiently. The left over confident provision are experiential to be the omega (ω). In the GWO method the hunting (optimization) is showed using the α , β , δ and ω . The overall flowchart for grey wolf optimization technique is intended in fig.2,

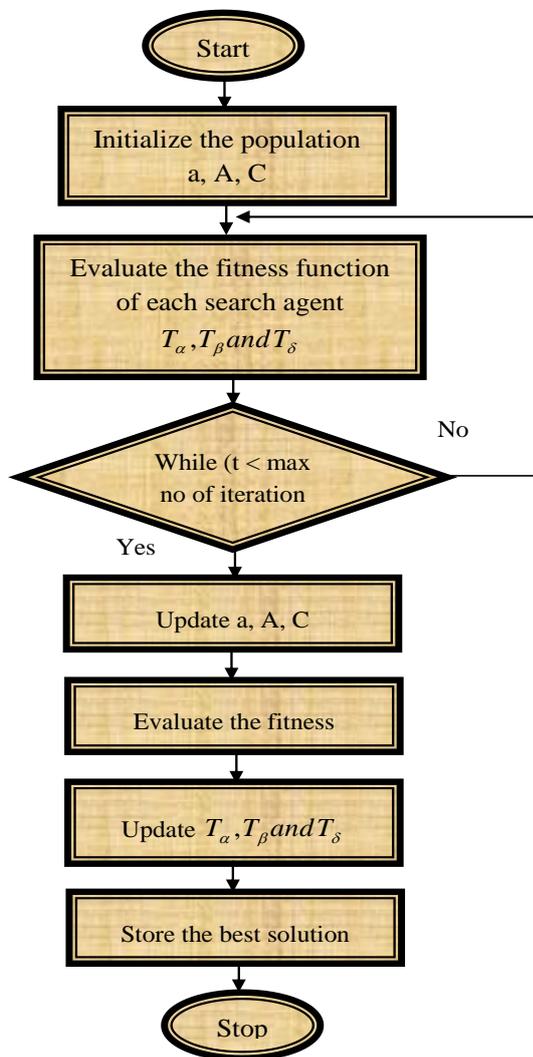


Figure.2 Flowchart for Grey Wolf Optimization technique

The stage by stage process of grey wolf optimization algorithm is provided below,

- **Initialization process**

Here, we start the existing test cases for each group and also a , A and C as coefficient vectors.

- **Fitness evaluation**

Assess the fitness utility on the basis of the equation (6) and further more select the better solution.

$$Fit_i = \min \sum_{i=1}^n \text{coverage metrics} \quad (6)$$

- **Separate the solution based on the fitness**

Presently, we get the split solution on the basis of the fitness value. Deliberate the initial finest fitness solutions T_α , the second finest fitness solutions T_β and the third finest fitness solutions T_δ .

- **Encircling prey**

The tracking is absorbed using α , β , δ and ω tag together with these three contenders. For the assortment to trail a victim is foremost adjoining it.

$$T(t+1) = T(t) + \vec{A} \cdot \vec{K} \quad (7)$$

$$\vec{K} = |\vec{C} \cdot T(t+1) - T(t)| \quad (8)$$

$$\vec{A} = 2\vec{a}r_1 - \vec{a} \text{ and } \vec{C} = 2r_2 \quad (9)$$

Where, t represents the iteration number.

$T(t)$ represents the prey position.

A and C represents the coefficient vector.

\vec{a} is linearly decreased from 2 to 0.

r_1 and r_2 represents the random vector [0, 1].

- **Hunting**

We are supposing that the alpha (best candidate solution), beta and delta comprise the improved data about the probable position of the victim in order to reproduce precisely the tracking activities of the grey wolves. As a result, we accrue the initial three finest solution proficient up to now and require the further discover mediator (including the omegas) to modify their pose on the basis of the pre arrangement of the finest explore mediator. For repetition, the new solution $T(t+1)$ is expectable using the formulae stated underneath.

$$\vec{K}^\alpha = |\vec{C}_1 T_\alpha - T|, \quad \vec{K}^\beta = |\vec{C}_2 T_\beta - T|, \quad \vec{K}^\delta = |\vec{C}_3 T_\delta - T| \quad (9)$$

$$T_1 = T_\alpha - \vec{A}_1 \cdot (\vec{K}^\alpha), T_2 = T_\beta - \vec{A}_2 \cdot (\vec{K}^\beta), T_3 = T_\delta - \vec{A}_3 \cdot (\vec{K}^\delta) \tag{10}$$

$$T_{(t+1)} = \frac{T_1 + T_2 + T_3}{3} \tag{11}$$

Where T is the location of current result and $\vec{C}_1, \vec{C}_2, \vec{C}_3$ and $\vec{A}_1, \vec{A}_2, \vec{A}_3$ are the arbitrary vectors; it can be investigational that the finishing location would be in a casual position enclosed using a circle that is dissimilar through the location of alpha, beta, and delta in the discover gap. By means alpha, beta, and delta predict the situation of the victim, and further wolves' modernize their location unpredictably in the area of the victim.

• **Attacking Prey (exploitation) and Search for prey (exploration)**

Examination and operation are assured using the adaptive values of a and A . The adaptive values of restraint a and A permit GWO to professionally conversion amongst examination and utilization. By deteriorating A , half of the iterations are continuous to examination ($|A| \geq 1$) and the further half are dedicated to utilization ($|A| < 1$). The GWO comprises only two leading limits to be accustomed (a and C). However, we include reserved the GWO algorithm as uncomplicated as attainable via the minimum number of operative to be familiarized. The process will be determined anticipating the highest precision is developed. Lastly, the finest characteristic is designated and supply to the extra process. The pseudo code for the GWO technique is shown in below table 1,

Table 1. The pseudo code for GWO

```

Begin
Initialize the solution  $T_i(i=1, \dots, n)$ 
Initialize  $a, A$  and  $C$  are the coefficient vector
Find the fitness of the initial solution
Separate the solution based on the fitness
 $T_\alpha$  = the first best search solution
 $T_\beta$  = the second best search solution
 $T_\delta$  = the third best search solution
While (t < max num of iteration)
For each search solution
Update the position of the current search solution using equation (11)
End for
Update  $A$  and  $C$  by equation (8)
Calculate the fitness of the new search solution
Update  $T_\alpha, T_\beta$  and  $T_\delta$ 
 $t = t + 1$ 
End while
Stop
    
```

On the basis of the above process we order the test cases. The function of the projected method is assessed by execution time and memory value and it evidently labeled in segment.5.

5. Results and Discussion

The effective test case prioritization in cloud depended on regression testing by Kernel fuzzy c means clustering algorithm is achieved in working platform of JAVA with Cloud Sim and machine conformation as trails,

- Processor: Intel core i5
- OS: Windows 7
- CPU speed: 3.20 GHz
- RAM: 4GB

In our projected method we are seeing three data center and nine virtual machines are utilized for optimal test case prioritization. The input implementations of the projected methods are banking, hospitality and library management implementation. The total application size is 163002 for banking, 74702 for hospitality and 7156824 for library organization. Dissimilar performance assessment measures are measured to assess the performance of the projected method in test case prioritization in cloud depended on regression testing procedure.

5.1 Performance Evaluation Measures

The performance of our projected technique is assessed on the basis of execution time and memory engaged for implementation procedure by variable the number of iteration in projected detection

procedure.

➤ **Execution Time**

The execution time in the Java with CloudSim package is directly rooted on the basis of the memory values. The time impediment is customarily characterized in a similar manner as to disregard the coefficients and lower order footings and in most cases, the time duration is calculated in millisecond (ms).

➤ **Memory Usage**

The Java with Cloud Sim program efficiently systematizes the memory for usage. Novel objects are produced and situated in the stack. The memory use is calculated in bits.

5.2 Analysis Process

Table 2. Fitness value using GWO for each cluster

Iteration	Fitness value using GWO for each cluster		
	Cluster 1	Cluster 2	Cluster 3
5	0.1077	0.478	0.06046
10	0.079	0.057	0.1923
15	0.088	0.2825	0.048
20	0.0199	0.6466	0.0479
25	0.0099	0.002	0.011

Originally test cases are produced from the input implementation with the help of coverage metrics. Then the existing test cases are congregated by KFCM algorithm. After the clustering procedure each cluster is provided to the GWO for test case arrangement. In GWO, the fitness value for each cluster is tabularized in table.2. The applied method deliberate three cluster and fitness performance for three cluster value is displayed in above segment.

From table.2, it illustrates the fitness value for each cluster. Fitness performance of grey wolf optimization method is minimum sum of coverage metrics value for each test cases, in that the cluster one has the smallest fitness value is reach in 25th iteration. It reaches 0.0099 values in the 25th iteration. For cluster two and cluster three also grasps the minimum value in the 25th iteration. From the solution we evidently understand that when the number iteration influences the maximum the projected method achieves the objective performance.

The projected execution time and memory use is tabularized in table.3. The time duration is calculated to accomplished the projected work is

exposed below,

Table.3 characterizes the execution time by varying the number of iteration of the projected optimal test case prioritization. To comprehensive the 5th iteration the projected method takes 327ms. For implementation the 10th and 15th iteration the advanced optimal test case prioritization takes 393,s and 424ms.The applied method takes 578ms and 619ms to gloss the 20th and 25th iteration. The graphical illustration of execution time by variable the number of iteration is strategized in fig.3,

Table 3. The overall execution time and memory usage of proposed technique

Iteration	Time (ms)	Memory (bits)
5	327	5779872
10	393	5658728
15	424	6458456
20	578	8105880
25	619	4874008

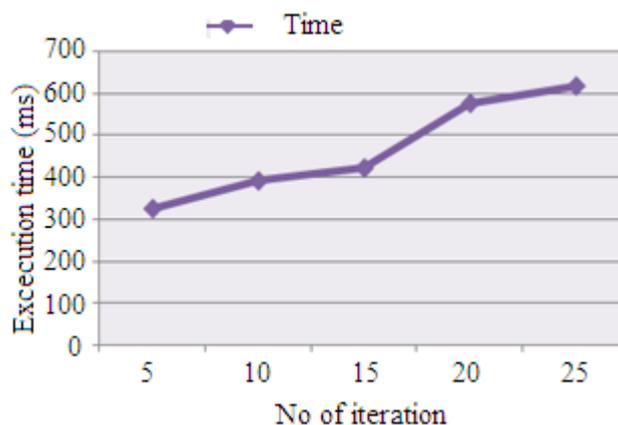


Figure.3 Execution time by varying the number of iteration

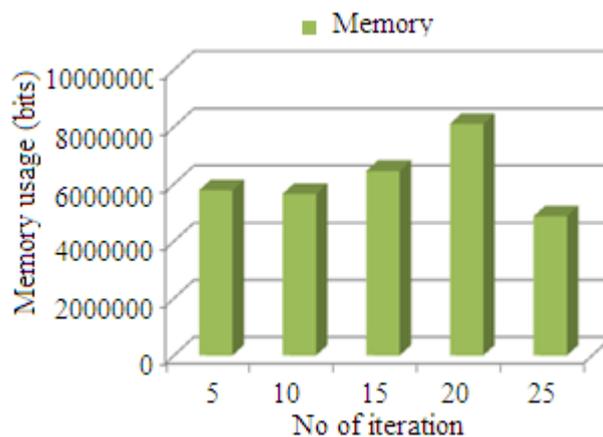


Figure.4 Memory usage by varying the number of iteration

The memory use for concluding the iteration 5th and 10th is 5779872 bits and 5658728 bits. For finishing 15th, 20th and 25th iteration the memory use of our projected optimal task prioritization receipts 6458456 bits, 8105880 bits and 4874008 bits. The graphical depiction of memory use by varying the number of iteration is displayed in fig.4,

5.3 Comparative Analysis

In comparative investigation, the available works are associated with our projected work in order to demonstrate the anticipated work is better one. For the available technique, here we are bearing in mind k means clustering algorithm along with grey wolf optimization algorithm. The graphical illustration of comparative investigation is displayed in fig.5. At this point the execution time to implementation the entire procedure of projected technique is associated with the available methods.

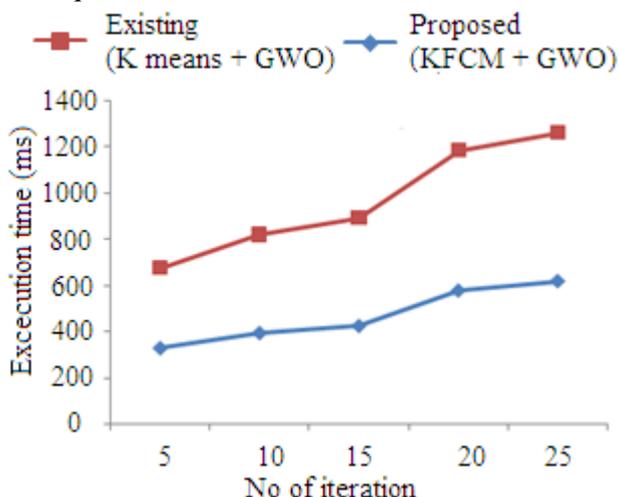


Figure.5 Execution time comparison for proposed Vs existing methods

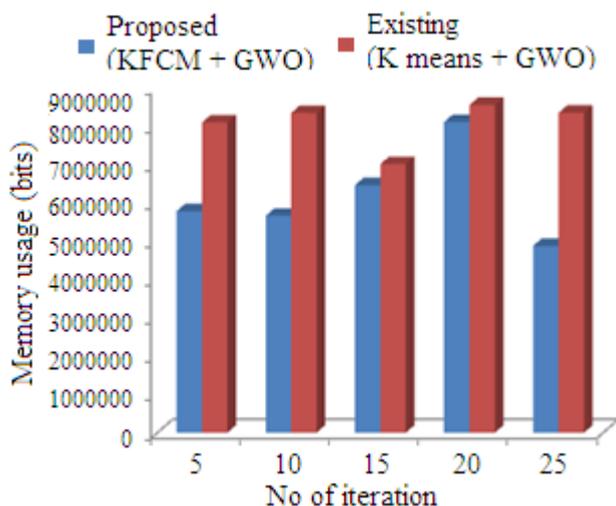


Figure.6 Memory usage comparison for proposed Vs existing methods

When investigating the fig.5, the projected KFCM and GWO algorithm takes minimum execution time to comprehensive the prioritization procedure related to the available K means and GWO. The recommended method also associates the memory use with the available technique. The graphical illustration of memory use assessment of projected and available technique is displayed in fig.6,

From fig.6 the projected system takes least amount of memory use if associated with the available method. Unlike k-means wherever data point should completely belong to at least one cluster center however the planned KFCM the data point is allotted membership to every cluster center as a result of that data point might belong to over one cluster center. And also for overlapped data, the suggested kernel fuzzy c means gives better result and also achieves minimum execution time and memory value when compared to the k means clustering algorithm. From the solution our projected optimal test case prioritization method takes minimum execution time and less memory use for the test case prioritization is cloud depended on regression testing.

6. Conclusion

Effective test case prioritization in cloud depended on regression testing is anticipated in this article. The test cases are produced from the input implementation and then the existing test cases are grouped by Kernel fuzzy c means clustering algorithm. Subsequent each cluster groups are served to the test case prioritization, in that grey wolf optimization algorithm is utilized. The performance will be assessed using execution time and memory use. The applied method takes 468.2ms time to comprehensive the test case prioritization but the available technique takes to whole the prioritization procedure is 500.8ms that is maximum if associated with the projected method. The memory distribution for the prioritization procedure of our planned method usage 6175388.8 bits but the available technique use memory value as 8069208 bits. From the solution the applied method takes minimum execution time and a smaller amount memory use if associated with the available technique. In future work, the test cases are effectively prioritized by various clustering and optimization technique with more security in cloud.

Reference

- [1] S. Roongruangsuwan and J. Daengdej, "Test Case Prioritization Techniques", *In proceeding of the Journal of Theoretical and Applied Information Technology*, pp. 45-60, 2010.
- [2] D. Garg and A. Datta, "Test Case Prioritization due to Database Changes in Web Applications", *In proceeding of IEEE International Conference on Software Testing, Verification and Validation*, pp. 726-730, April 2012.
- [3] A. Pravin and S. Srinivasan, "Effective Test Case Selection and Prioritization in Regression Testing", *In proceeding of the Journal of Computer Science*, Vol. 9, No. 5, pp. 654-659, 2013.
- [4] S. Nachiyappan and S. Justus, "Cloud Testing Tools and Its Challenges: A Comparative Study", *International symposium on big data and cloud computing*, Vol.50, pp. 482-489, 2015.
- [5] S. D. Mungekar and D. K. Toradmalle, "Cloud Testing- An Advancement in Software Testing", *International Journal of Application or Innovation in Engineering and Management*, Vol.4, Issue 3, pp. 61-63, March 2015.
- [6] A.V. Katherine and K. Alagarsamy, "Conventional Software Testing Vs. Cloud Testing", *International Journal Of Scientific & Engineering Research*, Vol.3, Issue.9, pp. 1-5, Sep 2012.
- [7] R. Malhotra, A. Kaur and Y. Singh, "A Regression Test Selection and Prioritization Technique", *In proceeding of the Journal of Information Processing Systems*, Vol. 6, No. 2, pp. 235-252, June 2010.
- [8] S. Mohanty, A. A. Acharya and D. P. Mohapatra, "A Survey on Model Based Test Case Prioritization", *In proceeding of IEEE International Journal of Computer Science and Information Technologies*, Vol. 2, No. 3, pp. 1042-1047, 2011.
- [9] A. Laeprattanakul and T. Suwannasart, "An Approach for Regression Test Case Selection using Object Dependency Graph", *In proceeding of IEEE International Conference on Intelligent Networking and Collaborative Systems*, pp. 617-621, 2013.
- [10] T. Muthusamy and Dr. K. Seetharaman, "A Test Case Prioritization Method with Weight Factors in Regression Testing Based on Measurement Metrics", *In proceeding of IEEE International Journal of Advanced Research in Computer Science and Software Engineering*, Vol. 3, No. 12, pp. 390-396, Dec 2013.
- [11] D. Garg, A. Datta and T. French, "A Two-Level Prioritization Approach for Regression Testing of Web Applications", *In proceeding of IEEE International Conference on Asia Pacific Software Engineering*, Vol. 2, pp. 150-153, Dec 2012.
- [12] L. Mei, Z. Zhang, W. K. Chan and T. H. Tse, "Test Case Prioritization for Regression Testing of Service-Oriented Business Applications", *In proceeding of the Journal of Service Oriented Development*, pp. 901-910, 2009.
- [13] Z. Li, M. Harman and R. M. Hierons, "Search Algorithms for Regression Test Case Prioritization", *IEEE Transactions on Software Engineering*, Vol. 33, No. 4, pp. 225-237, April 2007.
- [14] L. chen, Z. Wang, L. Xu, H. Lu and B. Xu, "Test Case Prioritization for Web Service Regression Testing", *In proceeding of IEEE International Symposium on Service Oriented System Engineering*, pp. 173-178, Jun 2010.
- [15] A. K. Upadhyay and A. K. Misra, "Prioritizing Test Suites Using Clustering Approach in Software Testing", *In proceeding of IEEE International Journal of Soft Computing and Engineering*, Vol. 2, No. 4, pp. 222-226, Sep 2012.
- [16] E. Proko and I. Ninka, "Analysis and strategy for the performance testing in cloud computing", *Global journal of computer science and technology cloud and distributed*, Vol.12, Issue.10, pp. 1-5, July 2012.
- [17] S. Mirarab, S. Akhlaghi and L. Tahvildari, "Size-Constrained Regression Test Case Selection Using Multicriteria Optimization", *IEEE Transaction on Software Engineering*, Vol. 38, No. 4, pp. 936-956, Jul 2012.
- [18] P. K. Bhowate and Prof. V. B. Gadicha, "A Review Of Runtime Software Testing Of A Systems Migrate To The Cloud With A Taas Environment", *International journal of engineering research and technology*, Vol.2, Issue.2, pp.1-7, Feb 2013.
- [19] S. Sampath, R. Bryce and A. M. Memon, "A Uniform Representation of Hybrid Criteria for Regression Testing", *IEEE Transaction on Software Engineering*, Vol. 39, No. 10, pp. 1326-1344, Oct 2013.
- [20] X. Xu, H. Jin, S. Wu, L. Tang and Y. Wang, "URMG: Enhanced CBMG-Based Method for Automatically Testing Web Applications in the Cloud", *TSINGHUA science and technology*, Vol.19, No.1, pp. 65-74, Feb 2014.

- [21] K. Zhai, B. Jiang and W. K. Chan, "Prioritizing Test Cases for Regression Testing of Location-Based Services: Metrics, Techniques, and Case Study", *Transaction on Service Computing*, Vol. 7, No. 1, pp. 54-67, Jan 2014.
- [22] A. Shrivastva, S. Gupta and R. Tiwari, "Cloud based Testing Techniques (CTT)", *International journal of computer applications*, Vol. 104, No. 5, pp. 24-29, Oct 2014.