Journal of Theoretical and Applied Information Technology
15th October 2019. Vol.97. No 19

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

EFFICIENT REQUIREMENT PRIORITIZATION BASED ON ENHANCED MULTI-VERSE OPTIMIZER

KHALID K. ADHIM1, AMJAD HUDAIB2, BASHAR AL-SHBOUL3

[1] Department of Computer Science, King Abdullah II School for Information Technology, The University of Jordan , Amman, Jordan

[2] Department of Computer Information Systems, King Abdullah II School for Information Technology, The University of Jordan. Amman, Jordan

[3]Department of Business Information Technology, King Abdulla II School for Information Technology, The University of Jordan. Amman, Jordan

E-mail: [1]kkhalis68@yahoo.com , [2]ahudaib@ju.edu.jo , [3]bashar.shboul@gmail.com

ABSTRACT

Nowadays software development has more popular, and there are several methods have been introduced for achieving the software faster to meets the customer requirements. At the same time, the engineering requirements are one of the historic software engineering processes for identifying, analyzing, and validating requirements. The prioritization is most essential step for decision making and software product planning. Requirement prioritization is used for determining the requirements of a software product which should be included in the certain release and it is used in improving software product management. To achieve this Enhanced Multi-Verse Optimizer (EMVO) method is proposed. To achieve this more efficiently, MVO (Multi-Verse Optimizer) algorithm is utilized; it contains cosmology of three concepts such as the White hole, Blackhole, and Wormhole. The aim of this paper is to achieve the requirements prioritization in software with high efficient and high accuracy. The evaluation results proved the accuracy of the proposed method and are compared with various existing techniques.

Keywords: *Enhanced Multi Verse Optimizer, Engineering Process, Requirement Prioritization, Optimization, Metaheuristic Algorithms.*

1. INTRODUCTION

The process of constructing a software project and delivering a need of the customer, the requirement prioritization is the significant activity. The main purpose of software development is to provide customer satisfaction with narrowed resources. Here the time and budget is considered as the most essential factor. There are vast numbers of software requirements need to be prioritized with limited resources. If entire requirements are related to delivery at that time the software engineers may not know to prioritize the urge requirement based on the need of the customer. Hence, for the requirement prioritization in the development of the system, numerous numbers of stakeholders have been participated by means to prioritize the requirements in an optimal path based on their importance. Hence, the requirements have been orderly executed. While there is no possible way to execute all the requirements accordingly. Additionally, the opinion of stakeholders is vary based on the priority of every requirement. For the process of requirement prioritization, the agreement of the stakeholders has to be taken into consideration for prioritizing the requirement.

Using Multi-Verse Optimizer (MVO) based on cosmology concepts, it developed to perform exploration, exploitation, and local search. This algorithm solves the problems of search spaces.

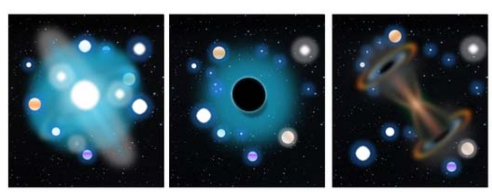


Figure 1: Cosmology basics of a White hole, black hole, and wormhole in MVO

Figure 1 shows the cosmology concept of MVO algorithm. For aggregation as well as selecting appropriate requirements in the process of software engineering, there are the vast number of approaches are in the market. These approaches are employed by various criteria for example importance, cost, time and so on. There are a vast number of features has to be

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

considered for prioritizing the requirements. Here, this feature is a property or project's attribute and its necessity. If the requirement is prioritized with an individual feature, making a decision is easy. If the prioritization is based on more than aspects such as a budget. In some of the situation, the customer may change their opinion thus makes the most wanted requirements to less needed category due to its expensive fact. The features interrelate and alters the priority because of their impacts one among another. At this point, there is a necessity to know which aspect of having an effect on conflicts. Additionally, the auxiliary features have to be considered while prioritizing the requirements along with the vital aspects. But for real-time, considering all the aspect is not possible. For a particular situation, which aspect has to be considered is the furthermost factor. The followings are the most relevant aspects of software projects, those are managers, users, developers. These aspects are generally appraised by the project's stakeholders.

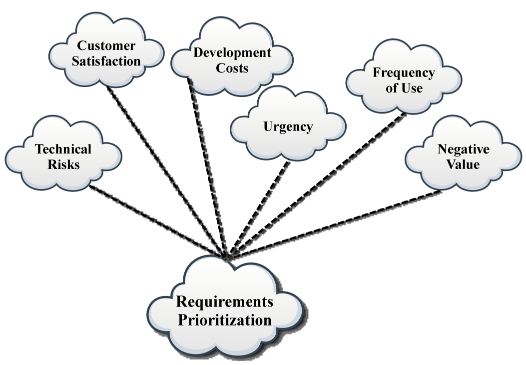


Figure 2: Requirements Prioritization Criteria [1]

Among the entire progress of the requirements prioritization, the most valued requirement sets are identified that provides support,

1. Preparing requirement subdivision and set up production which has to satisfy the need of the client.

2. Managing conflicting needs via determining variances among stakeholders.

3. Assessing the accomplished business assistance via requirement over the corresponding cost.

4. Agreeing on the essential set of requirements by the Stakeholders.

5. Scheduling and choosing a perfect requirement set which has to be accomplished in a consecutive delivery.

6. Executing qualified implication of each requirement thus provide an immense value at minimum cost.

In Requirement Prioritization, the first step is to group the priority based on stakeholder’s core requirement for the system. Then selecting and planning of prioritized order for implementation is performed. Then determining and managing the conflicting constraints among stakeholders. The constraints such our budget, time and quality. Then balance the benefits of requirements over the corresponding cost. Then estimation the group of requirements expected by stakeholders satisfaction. And consider the technical advantages of optimization. Finally, scheduling and choosing the group of requirements with the greatest value at the minimum cost.

1.1 Objectives of the manuscript:

• To prioritize the requirement effectively by proposing a novel improved MVO algorithm.

• To reduce the computational cost by using stochastic universal sampling in the calculation maximization problem

• To reduce the time by proposing RP-NMVO.

1.2 Organization of the manuscript:

Section I provides the information about the domain as well the concept of Requirement prioritization (RP). Section II fives the discussion about the existing methods in RP. Section III provides the explanation about the proposed approach. Section IV gives performance analyses and comparisons of performance measures. Section V concludes the research work

2. RELATED WORK

This portion provides a discussion about the existing works and their merits and demerits.

[2] Presented the process of requirement prioritization which was combining two sources of information. Those two sources are end users and decision-makers. It was provided an exact description of various modules in order to sustain the information flow from the feedback of users'. Next, in the automation prioritization step, the feedback extraction was carried based on the preferred combination of the decision-makers. Further, the feedback properties are characterized in

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

the step of automated prioritization. Hence, there was a need for human abilities and interference was needed additionally for a decision process. [3] have classified and analyzed the approaches for non-functional requirements (NFR). In the process of Information Systems Engineering NFRs plays an important role. But there was no proper classification and presentation. According to the process of NFRs engineering and choice, it was classified as the pattern, Aspect-oriented and Goal oriented. To calculate the capability and NFRs quality, the Goal based methods are used. In the specified document, the identification of defects has been utilized through the approach based on aspect approach. From the same kind of project development, Pattern-based methods have been considered in order to measure NFRs specification depends on the collected knowledge. The main responsibility of this paper was mainly to provide classification in NFR's engineering process and discussing their scopes. [4] observed various Decision-making models based on Multicriteria namely MCDMs in Prioritizing Flood management Alternatives. In this Paper, Gorganrood River in Iran was applied as a real-world structure in order to arrange the management of risks occurred in flood. Flooding results in various hazards such as famine, disease, health impacts, cut down of services. So flood protection is inevitable. Many (MCDM) models had been proposed. The MCDMs were, CP SAW, VIKOR model, ELECTRE III, TOPSIS, ELECTRE-I, M-TOPSIS AHP, MCDMs used aggregation methods, non-parametric stochastic tests, correlation tests lastly analysis of sensitivity to determine the best model. From the methodological remarks, ELECTRE III was the most accurate and robust procedure for flood management mitigation, since it has the most appropriate weighing and ranking values. Out of all other models, they concluded that ELECTRE III is the robust model. And this approach was advised for making flood /water management problems [5] implemented a semi-automated method to prioritize requirement, based on preferences and dependencies, called DRank. In software requirements, there are many dependencies such as contribution dependencies, business dependencies.DRank method took dependencies into consideration. Tree – prioritization evaluation attribute is used to create an easier selection of ranking criteria. RankBoost which was used to calculate prioritization based on stakeholder's preferences. PageRank, it is an algorithm used to investigate the required dependencies. An integrated requirements prioritization method was introduced to develop the procedure further applicable and reasonable. They have conducted an experiment under control and the outcome demonstrated that DRank was consuming less time when compared with prevailing approaches. [6]Surveyed Analytic Hierarchy Process (AHP), hybrid assessment method (HAM) and integrated prioritization approach (IPA) empirically to arrange requirements that are non-functional and functional.

The research follows below; Requirement prioritization was useful in completing projects on an early schedule. They have conducted two experiments, in order to find the best approach. Firstly, they compared, IPA with the further method, named AHP-based method. Secondly, IPA was compared with the other substitute, named HAM-based approach. For experiment 1, Analyzing the IPA and AHP-based method, by evaluating, consumption of time, result from accuracy and finding answers to questions based on "how fast the approaches/which approach is easier/which approach produce more accurate result?. For experiment 2, analyzing the IPA and HAM same as the previous experiment. In addition, the collecting of twenty real requirements is achieved. From both experiments, they have concluded that the IPA approach presented high efficiency on comparison with AHP, HAM based approaches. [7] came up with as killed scheme for the process of software requirement prioritization named handler. The following are the findings of this paper,

- Selecting and prioritizing software requirements are the major difficult thing in software development. There were no current techniques used to prioritize a large number of software requirements.
- The Priority Handler (handler) based on the analytical hierarchal process and neural network to create the progress as scalable prioritization.
- The value of a requirement was predicted by a back-propagation neural network in order to make Handler efficient.

[8]Proposed a controlled experiment to prioritize Software requirements by linguistic tools as well as constraint solvers. Successful delivery of software system was based on prioritizing a large number of requirements. There are no feasible methods to prioritize software requirements, makes it very challengeable. They have inferred a method named, SNIPER which incorporated the usage of a linguistic tool and constraint solver. The SNIPER prioritized and selected the requirements according

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

to the handling of natural language as well as satisfiability modulo theories solvers. They have experimented on 40 system engineers –selected 20 from the list of 100 requirements. The final result showed that SNIPER ingests less time, improved accuracy of selection was very easy to achieve a weighted sum model.

[9] used grey wolf optimization for prioritizing the requirements mainly for the software projects. This algorithm mimics the grey wolves hunting behavior. Hence, it was a unique algorithm from others because it was having leadings in the control power that comprise four fundamental categories. That was alpha, beta delta, and omega wolves. In this work, the proposed algorithm presented requirement prioritization and executed in sequential order. Additionally, this work compared and investigated with AHP methods (analytical hierarchy process) based on the size of the dataset and average running time. Finally, this work demonstrated that the RP-GWO makes better outcomes than AHP approach about thirty percent. [10] studied the estimation of priority by deploying various approaches such as modified Fibonacci series cards, planning poker. This work also provided a solution for a multi-phase in order to the product backlog. Thus made Return on Investment ROI as maximum. Additionally, provided a handling approach. On technical debt prioritization, this work provided a prioritization of technical debt and the influence of non-functional requirements. There were no such kind of solution was described before to manage the technical debt particularly for legacy projects. So there was a need to manage technical debt effectively. This work found a solution for that by deploying various approaches. [11]Assessed the possibility of combining various models along with the process of testing in order to develop the statistical testing methods especially for Software Product Lines. As Markov chains, models are provided. The behavior of selection was evaluated by using Featured Transition Systems. This analysis was used for determining the features as well as products easily understand the behavior. Through enabling the integration of tool, efforts of modeling have been satisfied. As like, statistical prioritization has been achieved in noteworthy state space reduction. The report was according to the criteria of the feasibility on two various systems, such as Caroline and SferionTM, where, Caroline was a management system based on configurable course and SferionTM was dealt with a function of embedded helicopter landing. [12] proposed an innovative method by combining LFTA along with ANN in order to arrange the requirements. This method was delivered the furthermost client fulfillment along with all the features. On MATLAB software this was executed. The outcomes observed that decision making was improved when compared with existing approaches in case of high priority. By using fuzzy AHP, the real-time assignment was executed that was the best college selection. [13] Proposed a novel approach Adaptive Requirements Prioritization (ARP) which was improved decision making among issued requirements because of their concepts based on the objective as well as multidimensionality. By using Monte Carlo simulation, the effectiveness of the proposed approach was proven for various dimensions and various level of priority. [14] presented on-function's taxonomy requirements hence the analyst of the particular requirements effortlessly recognized various NFRs kinds based on the requirements in the prior requirements engineering stages. proposed a method analytic hierarchy process (AHP) technique mainly for effectively rank the strategy of decisions and devices during the consideration of the connections among the quality of system requirements, tactics of design and basic philosophies. [15]For medical patients, the method was established on the system of a remote monitoring system. The proposed method enables an aim that tactics ranking and making the principles. Finally, it has been removed discrepancies among commercials well as a valuation of technical stakeholder. [16]estimated the hybrid algorithm performance by using 19 artificial problems. The hybrid algorithm was made by merging multi-objective search algorithm contains NSGA-II with Random Search (RS). The performance evaluation was carried out onRALIC dataset main focusing various 19 issues. The outcomes demonstrated that the algorithm NSGA-II solved the requirements prioritization issues along with improved performance when compared with RS.

3. PROPOSED WORK

This section deliberates the proposed approach with the flow of RP-EMVO and explains about enhanced MVO algorithm in Requirement Prioritization (RP).

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

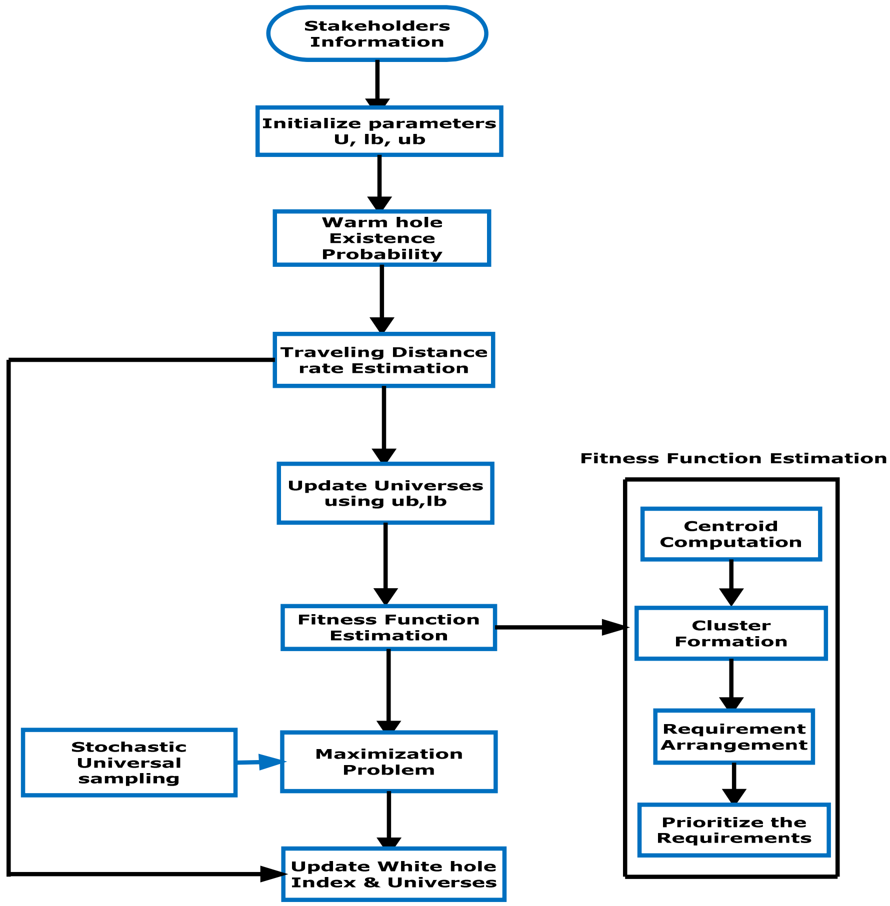


Figure 3: Overall flow of proposed RP-EMVO Approach

The MVO (Multi-Verse Optimizer) algorithm is one of the benchmarked algorithms, and it utilizes the concepts of a white hole, black hole, and wormhole. It contains various rules are applied in MVO during optimization. It involves the process of changing the objects from a high inflation rate to low inflation rate. At each iteration, the objects are moved with maximum inflation rates tend to move to the universes with minimum inflation rates via white/ black holes. In MVO the wormhole helps to exploit search spaces. The solution of each universe, a variable in solution corresponds to object in-universe. The main advantage of this algorithm is that can exchange the information between candidate solutions, and the disadvantage of this algorithm is minimum simplicity, requires the maximum number of functions.

$$U = \begin{bmatrix} x_1^1 & x_1^2 & \cdots & x_1^d \\ x_2^1 & x_2^2 & \cdots & x_2^d \\ \vdots & \vdots & \vdots & \vdots \\ x_n^1 & x_n^2 & \cdots & x_n^d \end{bmatrix}$$

d ---→ number of variables
n --→ number of universe

MVO pseudocode

```
SU=Sorted universes
NI=Normalize inflation rate (fitness) of the universes
for each universe indexed by i
Black_hole_index=i;
for each object indexed by j
r1=random([0,1]);
if r1<NI(Ui)
White_hole_index= RouletteWheelSelection(-NI);
U(Black_hole_index,j)= SU(White_hole_index,j);
end if
```

```
end for
end for
```

MVO performs the optimization randomly and creating set of random solutions. The ability of MVO algorithm is to solves the population-based problems, and it easily compared with other optimization algorithm such us Grey Wolf Optimizer, Particle Swarm Optimization. In this algorithm every universe has an inflation rate and it has the expansion.

Figure 2 explains, the efficient requirement prioritization, the process is initiated by initializing the parameters such as U, lb, and up. The step by step procedure of RP- EMVO is given in algorithm 1. In EMVO, initially warm hole probability is calculated. The rate of traveling distance is calculated subsequently, the updating of universes is carried by using up and lb. Then the fitness function is calculated. For the fitness function estimation, there are four essential steps are followed those are

- Centroid computation
- Cluster Formation
- Requirement Arrangement
- Prioritize the Requirements

After calculating the fitness function the process of maximization problem is done and finally updates the white hole index and universes.

3.1 Fitness function estimation:

The requirements of ordering define the fitness function, the proposed algorithm is the input to the fitness function and it contains clustering formation. Fitness function doesn't depend on the available information. An individual is, all the requirements of a sequence are to be prioritized. To derive this individual's population, the first fitness function is estimated.

The main aim of this fitness function is the measurement of the distance among every universe and every centroid to coordinate with the nearest one universes and it becomes the cluster member.

3.1.1 Clustering:

To achieve a group of clusters, the k-means clustering algorithm is used. Through this k-means clustering, the centroid estimation and cluster formation were done. It defines the number of clusters ask, the clusters are identified by distance, connectivity. Choose the random data point as the centroids due to iteration then Calculate the distance between pixel intensities and cluster centroids. And require to grouping the pixels which are the closer distance of centroids.

By using equation (1) the number of clusters is found. Then by using Equation (2), the number of the centroids is found. This means for the value of the number of the centroids is the same as with the number of clusters. Thus the centroids are chosen randomly. Every universe is linked with the neighborhood centroid by using equation (3).

Number of clusters = ceiling

$$Number\ of\ clusters = ceiling\left(\frac{Number\ of\ universes}{Number\ of\ requirements}\right) \quad \ldots\ldots\ldots(1)$$

$$Number\ of\ centroid = Number\ of\ clusters \quad \ldots\ldots\ldots\ldots\ldots(2)$$

$$Distance = \left\| X_i - centroid_j \right\| \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(3)$$

The double notation of above equation 3 denotes the Euclidean distance function. Where X_i denotes the universes (i), as well as centroid j, denotes to centroid which is presented in cluster j.

3.1.2 Requirements arrangement and prioritization

The entire search agent specifies a necessary from both aspects of importance in weight and definite factors. It is also explained as a point to the important requirements in the progress operation for the defined project. This work explains that the required weight represents the ratio of the value of cost for a provided defined requirement. Likewise in the AHP technique, this work utilizes the based cost value methodology. Further, the efficiency of the proposed algorithm was compared with the AHP method which is very famous technique based on cost value. Apart from that this appliance on the basis of pairwise comparison and the arranged pair of provided requirements increase relays on the manipulated ratio with the use of

$$Ratio = Point / value \quad (4)$$

Where point denotes Stakeholder ratings, Value shows the stakeholder requirements.

While carrying out iterations, the function of requirement prioritization processed till entire clusters become empty. The requirement should have minimum ratio while relating other ratios that need to be prioritized as very important. From its cluster, as well as the search space, it should be dropped out. The subsequent mathematical formula has been proposed.

$$Global\ Min\ Ratio = MIN\ \{Local\ Min\ Ratio\ (1) \ldots Local\ Min\ Ratio\ (C)\}\ (5)$$

Where Signifies the number of clusters.

Algorithm 2: Fitness based RP-EMVO
1. While each cluster is non-empty (1, 2, 3, N)
2. for each cluster C
3. Arrange the requirements by the eq (4)
4. Select the requirement which has a minimum ratio
5. End for
6. Select a minimum of minimums ratio by the eq (5)
7. Remove the selected one from search space

3.2 Stochastic universal sampling:

Some cases generate more value per unit *of work* compared than other use cases when we consider both value (v) and cost(c). The Stochastic universal sampling is a single-phase sampling algorithm. It contains zero bias that means minimum spread. The SUS used for sampling all the solutions by choosing evenly spaced intervals. The selection is a probabilistic process, it's based on the individual's fitness.

SUS (Population, N)
Where F is the population's total fitness
N is the offspring number for manage
P is the distance among two pointers
Start: Among 0 and P the random number has to be lies
Pointers: $[start + i * P \mid i\ in\ [0 \ldots, (N-1)]$
Keep =[]
For P in points
i=0
While fitness sum of the population[o, i]<p
i++
Adding the population [i] in order to keep
Return keep

After sampling at the end of the process, the Multi-class Learning method is used for testing and training, it is based on the linear model. It contains various classes to train the overall training data and find the test samples.

3.3 Enhanced Multi-verse Optimizer:

The algorithm I: Enhanced Multi-Verse Optimizer
Inputs: Stakeholders Recommendations S_{Recom}, Stakeholder ratings S_{Rate}, Stakeholder ratings on requirements $S_{Rate-Req}$
Outputs: Requirement prioritized RP
Procedure:
Steps 1: let take stakeholders information as universes.
$U = \{S_{Recom}, S_{Rate}, S_{Rate-Req}\}$
Initialize,
n – number of universes,
$I_{rBU} = Inf$ // best universe inflation rate
$WEP_{mx} = 1$ // maximum of wormhole existence probability
$WEP_{mn} = 0.2$ // maximum of wormhole existence probability
$Iter = 100$ // Maximum iteration
$BU = 0$ // initialize best universes\ as 0
While t> $Iter$ // perform iteration process
$ub = 100$ // upper bound
$lb = -100$ // lower bound
$WEP = WEP_{mn} + t * \frac{(WEP_{mx} - WEP_{mn})}{Iter}$ // wormhole existence probability estimation
$TDR = 1 - \frac{\sqrt[6]{t}}{\sqrt[6]{Iter}}$ // traveling distance rate estimation
$Flag_{ub} = size(U) > ub$ // calculate label of the universes by the upper bound
$Flag_{lb} = size(U) > lb$ // calculate label of the universes by the lower bound
Step 2: update universes using upper bound and lower bound,
$U = U + (ub * Flag_{ub}) + (lb * Flag_{lb})$
Step 3: Estimate the fitness function for the updated universes which is described in Algorithm II

$Ir = fitness(U)$ // Ir-inflation rates
Step 4: Update best universe inflation rate and the best universe at the iteration by comparing with estimated inflation rates
$I_{rBU} = \begin{cases} Ir & if\ Ir < I_{rBU} \\ I_{rBU} & else \end{cases}$
$BU = \begin{cases} U & if\ Ir < I_{rBU} \\ BU & else \end{cases}$
Step 5: Perform maximization problem using stochastic universal sampling which is explained in the algorithm III,
$SU = sort(BU)$ // Sort universes
for each universe indexed by i
$BHI = i$ // Blackhole index
for every inflation rate indexed by j
$ran_1 = random([0,1])$
if $ran_1 < I_{rBU}(U_i)$
$WHI = sus(I_{rBU})$ //WHI – white hole index, sus – stochastic universal sampling
$U(BHI, j) = SU(WHI, j)$ //Update universes
endif
end for
end for
Step 6: update universes based on the upper bound, lower bound and estimated traveling distance rate,
for every universe indexed by i
for every inflation rate indexed by j
$ran_2 = random([0,1])$
if $ran_2 < WEP$
$ran_3 = WEP + ran_3$
$U(i,j) = BU + \left(TDR * \left(\left((ub - lb) * ub\right)\right)\right)$ if $ran_3 < 0.5$
$U(i,j) = BU - \left(TDR * \left(\left((ub - lb) * lb\right)\right)\right)$ if ran
endif
end for
end for

The enhanced Multi-verse Optimizer exploits with cosmology concepts. It performs with effective optimizing real problems. The proposed algorithm outperforms all of the other algorithms for the solving of constrained problems.
In enhanced Multi-verse Optimizer algorithm the input takes as Stakeholders Recommendations, Stakeholder ratings, Stakeholder ratings on requirements and the output gives Requirement prioritize(RP).First considers the stakeholders information as universes then initialize best universe inflation rate and maximum of wormhole existence probability. In number of iterations the boundary function is calculated. Then update the universes using these boundaries value. After that Estimate the fitness function for the updated universes and update the best universes based on comparing with estimated inflation rates. Finally, update the universes based on boundaries and estimated traveling distance rate.

4. PERFORMANCE ANALYSIS

4.1 Dataset description:

The performance of the proposed framework was developed and implemented by using RALIC Dataset[17]. The abbreviation of RALIC is Replacement Access, Library and ID Card project. The dataset had more than 1,000 ratings from stakeholders. We have used the reliable dataset for repeating random sampling, for each experimentation we have achieved many iterations.
The requirement prioritization is embedding with non-functional requirements in RALIC dataset. It improves the prevailing access system of control that contains approximately thirty thousand users and sixty shareholder groups in software development. This dataset consists of shareholder information and their necessaries that comprise original information for the description of the texts,

Journal of Theoretical and Applied Information Technology
15th October 2019. Vol.97. No 19

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

references and significant values from shareholders on the basis of their requirements. No observance of insecure data has been found in the RALIC dataset.

The proposed work was compared with existing requirement prioritization methods AHP (Analytical Hierarchy Process), RP-GWO [9] for time. The grey wolf optimization (GWO) algorithm is mainly applied for prioritizing the requirements of a software development. It is one of the heuristic mechanisms and main goal of RP-GWO algorithm is to set the optimal solution for given population issues. The Analytical Hierarchy Process (AHP) explains the prioritization of possible pairs of requirements. It mainly contains pairwise comparison strategy. The accuracy is compared with the existing stake QP[18] and Lim et. al.[19]. These existing, RP-GWO works contain functional requirements.

4.2 Time comparison

Table 2: Comparison of Time with Existing

Data set	AHP (Sec)	RP-GWO (Sec)	RP-EMVO (Sec)
100	0.09064	0.0719	0.0524
200	0.13126	0.09688	0.07459
300	0.17814	0.12814	0.0957
400	0.2875	0.18126	0.7574
500	0.3844	0.2344	0.1512
600	0.49376	0.31564	0.2953
700	0.65002	0.42816	0.3136
800	0.80314	0.58438	0.4421
900	1.0625	0.73752	0.6503
1000	1.24376	0.91876	0.8548
Average	0.532512	0.369704	0.36787

Table 2 shows the comparison of time measures with existing AHP, RP-GWO methods. The requirements are prioritized based on time, and proposed method had the requirement prioritization with minimum amount of time compared than other techniques. So the proposed method achieved the minimum time complexity.

4.2.1 For dataset 100:

The following figure shows, the comparison of dataset 100. It was compared with the existing approach AHP, RP-GWO [9].

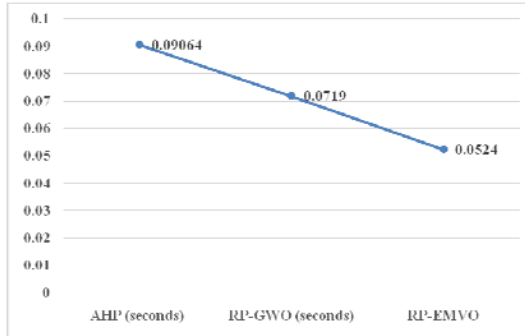


Figure 4: Comparison For dataset 100

Figure 4 shows the time comparison of 100 dataset with existing techniques

4.2.2 For dataset 200:

The following figure shows, the comparison of dataset 200. This was compared with the existing approach AHP, RP-GWO [9].

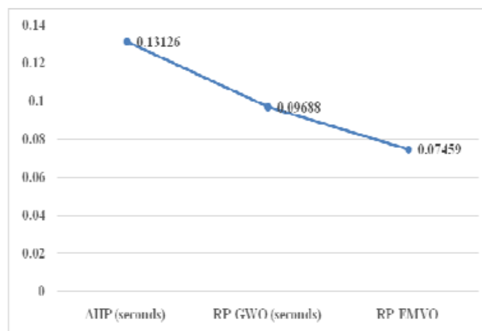
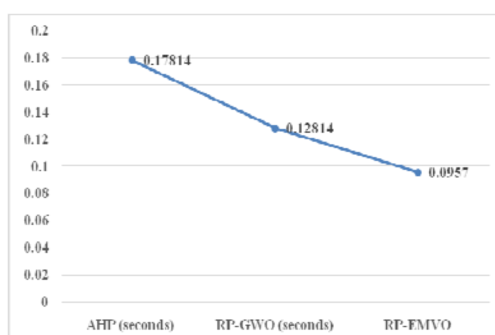


Figure 5: Comparison For dataset 200

Figure 5 shows the time comparison of 200 dataset with existing techniques

4.2.3 For dataset 300:

The following figure shows, the comparison of dataset 300. This was compared with existing approach AHP, RP-GWO [9]



ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Figure 6 Comparison For dataset 300

Figure 6 shows the time comparison of 300 dataset with existing techniques

4.2.4 For dataset 400:

The following figure shows, the comparison of dataset 400. This was compared with existing approach AHP, RP-GWO [9]

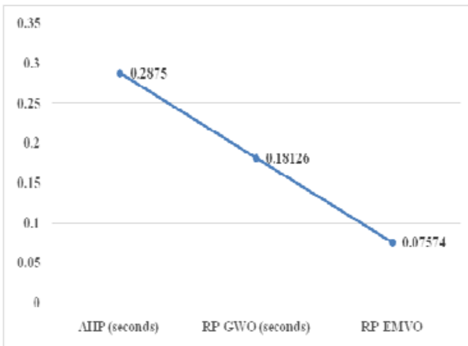


Figure 7: Comparison For dataset 400

Figure 7 shows the time comparison of 400 dataset with existing techniques

4.2.5 For dataset 500:

The following figure shows, the comparison of dataset 500. This was compared with existing approach AHP, RP-GWO [9]

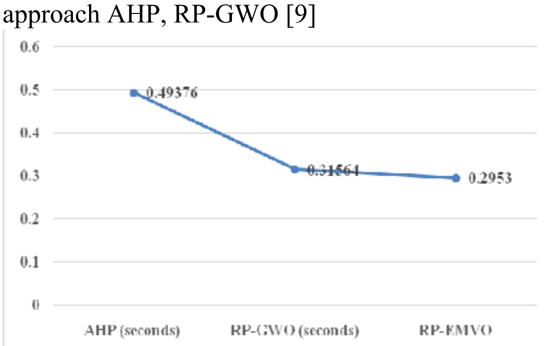


Figure 8: Comparison For dataset 500

Figure 8 shows the time comparison of 500 dataset with existing techniques

4.2.6 For dataset 600:

The following figure shows, the comparison of dataset 600. This was compared with existing approach AHP, RP-GWO

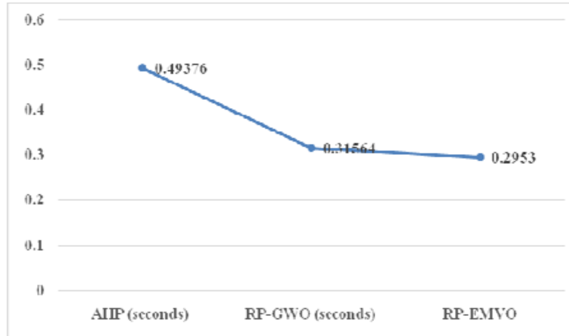


Figure 9: Comparison For dataset 600

Figure 9 shows the time comparison of 600 dataset with existing techniques

4.2.7 For dataset 700:

The following figure shows, the comparison of dataset 600. This was compared with existing approach AHP, RP-GWO.

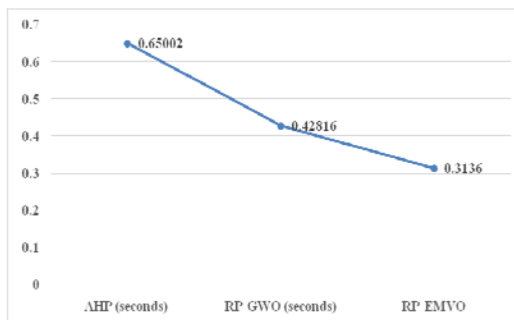


Figure 10: Comparison For dataset 700

Figure 10 shows the time comparison of 600 dataset with existing techniques

4.2.8 For dataset 800:

The following figure shows, the comparison of dataset 600. This was compared with existing approach AHP, RP-GWO.

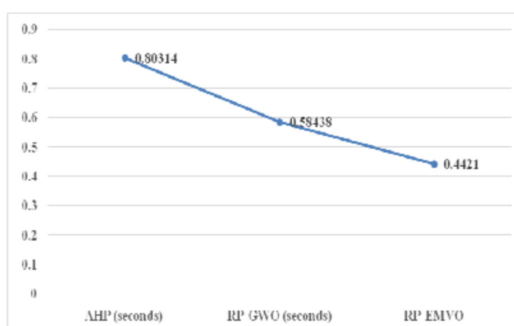


Figure 11: Comparison For dataset 800

Figure 11 shows the time comparison of 600 dataset with existing techniques

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

4.2.9 For dataset 900:

The following figure shows, the comparison of dataset 600. This was compared with existing approach AHP, RP-GWO.

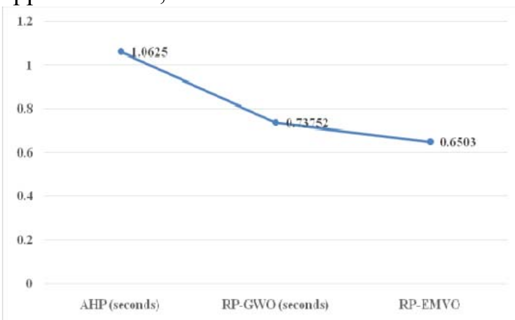


Figure 12: Comparison For dataset 900

Figure 12 shows the time comparison of 900 dataset with existing techniques

4.2.10 For dataset 1000:

The following figure shows, the comparison of dataset 700. This was compared with existing approach AHP, RP-GWO.

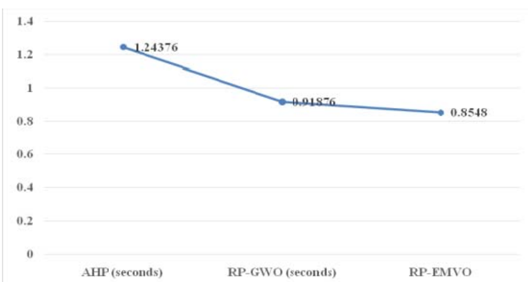


Figure 13: Comparison For dataset 1000

Figure 13 shows the time comparison of 1000 dataset with existing techniques

4.3 Average time comparison with existing:

Figure 11 shows the average time of the proposed method with existing RP-GWO and AHP methods.

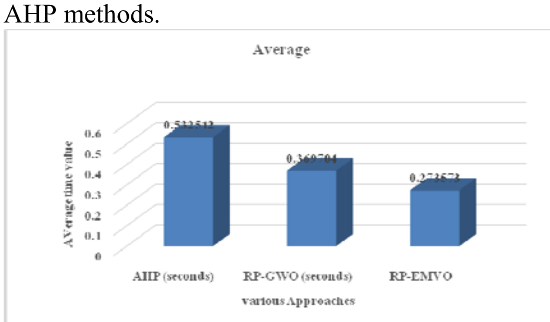


Figure 14: Overall comparison of the average time

Figure 14 explains about average time of proposed RP-EMVO method, and proves it takes minimum amount of time for requirement prioritization

4.4 Accuracy comparison with existing:

The accuracy is compared with the existing stake QP[18] and Lim et. al.[19]. The graph clearly illustrates the proposed was outperformed than existing approaches.

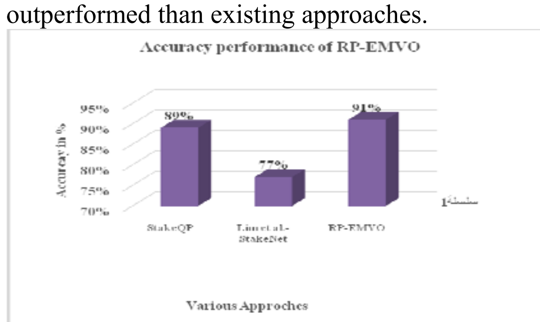


Figure 15 Overall comparison of Accuracy

Figure 15 defines the accuracy of various approaches. Accuracy is the important measure, and the evaluation results proved higher accuracy compared than other techniques. It shows 8% higher than existing methods.

5 CONCLUSION

The requirement prioritization is very much essential for the exact arrangement of the data on the basis of priority in the requirement engineering process. Various prioritization methodologies have been implied to develop software with greater quality. The proposed work used the RP EMVO (Requirement Prioritization Enhanced Multi-Verse Optimizer) is used to implicit the software plan. The importance of the EMVO algorithm which is a benchmark algorithm is highly depending on the three components which are a wormhole, black hole, and a white hole. When comparing the RP EMVO algorithm with the prevailing methods with respect to requirement prioritization, it shows a higher efficiency of time and accuracy that reaches up to ninety-one percentage.

REFRENCES:

[1] N. Riegel and J. Doerr, "A systematic literature review of requirements prioritization criteria," in *International Working Conference on Requirements*

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Engineering: Foundation for Software Quality, 2015, pp. 300-317.

[2] I. Morales-Ramirez, D. Munante, F. Kifetew, A. Perini, A. Susi, and A. Siena, "Exploiting user feedback in tool-supported multi-criteria requirements prioritization," in *2017 IEEE 25th International Requirements Engineering Conference (RE)*, 2017, pp. 424-429.

[3] M. M. Hasan, P. Loucopoulos, and M. Nikolaidou, "Classification and qualitative analysis of non-functional requirements approaches," in *Enterprise, Business-Process and Information Systems Modeling*, ed: Springer, 2014, pp. 348-362.

[4] N. Chitsaz and M. E. Banihabib, "Comparison of different multi criteria decision-making models in prioritizing flood management alternatives," *Water Resources Management,* vol. 29, pp. 2503-2525, 2015.

[5] F. Shao, R. Peng, H. Lai, and B. Wang, "DRank: A semi-automated requirements prioritization method based on preferences and dependencies," *Journal of Systems and Software,* vol. 126, pp. 141-156, 2017.

[6] M. Dabbagh, S. P. Lee, and R. M. Parizi, "Functional and non-functional requirements prioritization: empirical evaluation of IPA, AHP-based, and HAM-based approaches," *Soft Computing,* vol. 20, pp. 4497-4520, 2016.

[7] M. I. Babar, M. Ghazali, D. N. Jawawi, S. M. Shamsuddin, and N. Ibrahim, "PHandler: an expert system for a scalable software requirements prioritization process," *Knowledge-Based Systems,* vol. 84, pp. 179-202, 2015.

[8] J. McZara, S. Sarkani, T. Holzer, and T. Eveleigh, "Software requirements prioritization and selection using linguistic tools and constraint solvers—a controlled experiment," *Empirical Software Engineering,* vol. 20, pp. 1721-1761, 2015.

[9] R. Masadeh, A. Alzaqebah, A. Hudaib, and A. Rahman, "Grey Wolf Algorithm for Requirements Prioritization," *Modern Applied Science,* vol. 12, p. 54, 2018.

[10] V. Sachdeva, "Requirements prioritization in agile: use of planning poker for maximizing return on investment," in *Information Technology-New Generations*, ed: Springer, 2018, pp. 403-409.

[11] X. Devroey, G. Perrouin, M. Cordy, H. Samih, A. Legay, P.-Y. Schobbens, *et al.*, "Statistical prioritization for software product line testing: an experience report," *Software & Systems Modeling,* vol. 16, pp. 153-171, 2017.

[12] Y. V. Singh, B. Kumar, S. Chand, and D. Sharma, "A Hybrid Approach for Requirements Prioritization Using Logarithmic Fuzzy Trapezoidal Approach (LFTA) and Artificial Neural Network (ANN)," in *International Conference on Futuristic Trends in Network and Communication Technologies*, 2018, pp. 350-364.

[13] A. Salado and R. Nilchiani, "Adaptive requirements prioritization (ARP): Improving decisions between conflicting requirements," *Systems Engineering,* vol. 18, pp. 472-490, 2015.

[14] N. Afreen, A. Khatoon, and M. Sadiq, "A taxonomy of software's non-functional requirements," in *Proceedings of the second international conference on computer and communication technologies*, 2016, pp. 47-53.

[15] M. Kassab and N. Kilicay-Ergin, "Applying analytical hierarchy process to system quality requirements prioritization," *Innovations in Systems and Software Engineering,* vol. 11, pp. 303-312, 2015.

[16] Y. Li, M. Zhang, T. Yue, S. Ali, and L. Zhang, "Search-based Uncertainty-wise Requirements Prioritization," in *2017 22nd International Conference on Engineering of Complex Computer Systems (ICECCS)*, 2017, pp. 80-89.

[17] "https://soolinglim.wordpress.com/datasets/."

[18] F. Hujainah, R. B. A. Bakar, and M. A. Abdulgabber, "StakeQP: A semi-automated stakeholder quantification and prioritisation technique for requirement selection in software system projects," *Decision Support Systems*, 2019.

[19] S. L. Lim, D. Quercia, and A. Finkelstein, "StakeNet: using social networks to analyse the stakeholders of large-scale software projects," in *Proceedings of the 32Nd ACM/IEEE International Conference on Software Engineering-Volume 1*, 2010, pp. 295-304.