Whale Optimization Algorithm for Requirements Prioritization

Abdullah Alzaqebah, Raja Masadeh and Amjad Hudaib

April 18, 2018

Whale Optimization Algorithm for Requirements Prioritization

Abdullah Alzaqebah
Computer Center
The World Islamic Sciences and Education University
Amman, Jordan
Abdullah.zaqebah@wise.edu.jo

Raja Masadeh
Computer Science Department
The World Islamic Sciences and Education University
Amman, Jordan
raja.masadeh@wise.edu.jo

Amjad Hudaib
Computer Information Systems Department
The University of Jordan
Amman, Jordan
ahudaib@ju.edu.jo

***Abstract*— In software projects, the requirement engineering phase is the base of any project and this phase is concerned in software requirements processes. Stakeholders are the main source of these requirements with respect of constraints and regulations. Requirements prioritization is considered as one of the most significant approach in the requirement engineering process because it is used to prioritize the execution order of requirements with respect to stakeholders' viewpoints. On the other hand, whale optimization algorithm is recently used in optimization problems since it mimics the Humpback whale hunting behavior by employing bubble net hunting technique. In this paper the Whale Optimization Algorithm (WOA) is employed in order to prioritize the software requirements by assuming the requirements in the search space and the hunting behavior is implemented to prioritize these requirements, the proposed technique is evaluated in term of running time with analytical hierarchy Process (AHP) to prioritize different size of requirement's sets, the results shows the RP_WOA outperforms the AHP technique by approximately (40%).**

Keywords*— *Requirement prioritizations (RP), Whale Optimization Algorithm (WOA), Analytical Hierarchy Process (AHP).

I. INTRODUCTION

Software Engineering contains one of the most considerable fields which is (RE). Also software development life cycle treated RE as the most important phase in the life cycle [1]. RE include requirements identification, requirements elicitation, requirements validation and documentation. All software projects, development process have many limitations such as time and budget to achieve market production; this leads us to transmit the software projects as sequential versions. However, in large projects many stakeholders participate to take decision what should be developed firstly. Such issue makes a challenge for stakeholders and participate the software engineers to put the suitable order for the requirements in an effective way to guarantee correct delivery and development of the project [1, 2, 3].

Thus, the important part in the requirement analysis is RE, because it contains the most important section which is (RP). To build software project and deliver good system that meet the customer needs RP is the most helpful process to accomplish the job. If a project is subjected to a tough execution plan, inadequate resources and the consumers' expectations are in a high level, and then the most important characteristics should be highlighted in an early enough time. That leads us to understand the importance of priority ordering of the requirements [1].

The existing of RP process give the stakeholder's height opportunity to be involved in deciding which requirements a software should be included based on the impact of development process and requirement importance, such involvement helps the stakeholders to deal with restrictions and project's resources limitations. Also they can discuss the contentions between opinions and viewpoints which impact in the software development. These opinions are come from various objectives, goals and roles of stakeholders. Thus, priority in projects with large number of requirements becomes a priority for the success or failure of the software project depending on project limitations and constraints [2].

The main objective of this participation is to ordering the requirements based on their importance and execution order [1, 2, 4, 9,22]. Accordingly, clustering technology can be used to classify requirements based on their importance. In other words, each cluster includes the same data objects, moreover, the clusters are various from others [15, 16]. Thus, the cluster analysis is one of the most important technique that used in data exploration process, image segmentation, neural computing and other engineering [17, 18]. Nowadays, many techniques are proposed by authors such as heuristic [18] and meta-heuristic [16] mechanisms to solve the issues that occur as an outcome of complex datasets. However, the most mechanisms that suggested in order to solve the problems of the optimizations depend on meta-heuristic algorithms; which purposes to determine the optimal solutions for achieving data classes and minimizes the problems of local minima. One of the latest meta-heuristic algorithm is the WOA which proposed by Mirjalili and Lewis in 2016 [5]. Which imitates the foraging behavior of humpbacks whales that is called bubble-net technique which indicates to the hunting process to locate and attack the prey. In addition to that, the main uniqueness between this algorithm and others meta-heuristic algorithms is the principles which improves the solution selection in each iteration.

The reminder of this paper is organized as follows: section II contains the related work, while section III outlines the Whale Optimization Algorithm (WOA). The proposed algorithm "RP-WOA" is discussed in Section IV. Section V represents the discussion of results of the proposed algorithm. VI Finally, the last section draws the conclusion of this study.

II. RELATED WORK

Many techniques have been developed in order to prioritize the given requirements for a particular project. Some of these techniques are better to applied in case small number of requirements in turn others are more appropriate for complex and huge number of requirements. The stakeholder plays an important role for decision making to analyze the requirements, then specify numbers which reflect their significance using prioritization mechanisms. The most popular RP techniques are classified into three classifications [6,22]; nominal scale, ordinal scale and ratio scale. Nominal scale prioritization techniques produce number of categories to which object can be classified [7]. In other words, the requirements can be categorized into classes based on their importance. MoSCoW (Must have, Should have, Could have, and Won't have) and Numeral assignment techniques are only the techniques that are included to this category .Whereas, ordinal scale prioritization techniques generate sorted lists of requirements. This category can say that requirement is more important than others but not to what amplitude. Priority group, bubble sort and minimal spanning tree (MST) techniques are the most common and used in this classification. Finally, ratio scale prioritization techniques generate ordered lists of requirements too, but this category can supply the relative diversity. Hierarchical cumulative voting (HCV), Analytic hierarchy process (AHP) and Hierarchy AHP are the common techniques in this classification [22].

Numerical assignment technique is considered as the most common and traditional technique. In this technique, all requirements join to one cluster which represent the same priority for all of them, which means does not tell which requirement has more or less primacy than other requirements within the same primacy [8]. This technique is mentioned by many researchers such as Sommerville and Sawyer [1], Karlsson et al. [10], and Berander and Andrews [11].

The priority groups' technique is similar to numerical assignment technique, which allocates each of the given requirements in one of the three main classifications; low class, medium class and high class. However, the variation between these mechanisms is that numerical assignment technique classifies the requirements once and only once but priority groups mechanism categories the requirements repeatedly [7,22].

Karlsson et al. [7] assume the empirical study about this priority classes, which prioritize 13 requirements with these techniques; AHP, MST, binary search, bubble sort and priority groups. This study draws that the worst technique is priority groups' technique in terms of difficult to use and slow to implement. Thus, this technique is not suitable to be applied on small number of requirements. While [7, 12] concludes that AHP technique is time consuming. Based on this conclusion, many techniques are developed in order to solve this issue by reducing the number of comparisons. Hierarchy AHP and MST are improved to solve time consuming. Bubble sort mechanism is compared in this study [7] which is the simplest to use and supplied fault tolerance, as well as reliable. It is concluded that this technique is faster than AHP in terms of time consumption, but is slower than MST technique. Thus, Bubble sort mechanism is suitable for small number of requirements. Based on this study [7], the MST is the fastest technique but it is not suitable for small number of requirements. However, in case of fault tolerance and reliability are more significant than time consuming; it is suitable for small number of requirements. Thus, this mechanism is suitable for large number of requirements. In addition to that, binary search tree is studied for requirement prioritization in order to rank the list of requirements.

Masadeh et al. [21] employed the grey wolf optimization algorithm for prioritizing the requirement, this algorithm mimics the hunting behavior of the grey wolf, the flock of wolves are splitted into groups and each group contains 5 – 12 wolves that surrounds the victim, each group has a leader which responsibles for making a decision and they concern about the nearest agent to the prey, in each group the nearest three wolves to the prey are considered by α, β and δ respectively in order to find the optimal solution. Researchers in [21] applied this algorithm for prioritize the software requirements and they showed in the results how this algorithm overcome the AHP technique in running time.

The work of [21] is limited to the number of members in each group. In other words, an increased number of clusters according to agents in the search space, so find an algorithm with less number of clusters will perform better theoretically; thus, the WOA was used in this research.

III. WHALE OPTIMIZATION ALGORITHM (WOA)

WOA is one of the latest randomly optimization techniques by Mirjalili and Lewis in 2016 [5]. This technique aims to imitate the hunting behavior of humpback whales by following the location of prey and encircle it. The hunting journey of humpback whales contains two main stages; exploration stage and exploitation stage. Thus, during searching in the search space for prey, WOA deems the current best candidate solution as target prey until the best solution is known because the optimal one in unknown a priori. Based on that, the other whales update their locations according to the current best solution. This behavior is modeled mathematically by the following Eq. (1) and Eq. (2) [5]:

$$\vec{D} = |\vec{C}.\overrightarrow{X*}(t) - \overrightarrow{X(t)}| \tag{1}$$

$$\vec{X}(t+1) = \overrightarrow{X*}(t) - \vec{A}.\vec{D} \tag{2}$$

Where (t) is denoted as current solution, $\vec{X}*$ indicates to the position vector of the best solution obtained so far and $\vec{X}$ is the position vector. Thus, $\vec{X}^*$ should be updated in each iteration accordingly to the best solution. $\vec{A}$ and $\vec{C}$ are coefficient vectors, which computed by the following Eq. (3) and Eq. (4) [5]:

$$\vec{A} = 2\vec{a} \cdot \vec{r} - \vec{a} \tag{3}$$

$$\vec{C} = 2 \cdot \vec{r} \tag{4}$$

Where $\vec{a}$ is linearly decreased from 2 to 0 over the range of iterations which plays the important role in order to control the exploration and exploitation phases. While $\vec{r}$ denotes as a random vector in [0, 1], which has significant role to make each search agent reaching any position during its search space to ensure the exploration [5].

The main hunting behavior of humpback whales is bubble-net technique [13] which represents the exploitation strategy in the WOA. The following approaches are modeled for the foraging mechanism:

- Shrinking encircling mechanism: This technique is modeled by minimizing $\vec{A}$ value, which is reduced by minimizing $\vec{a}$ value as shown in Eq. (3). Thus, the search agent updates its value during search space with a new value that is among the original location and the location of the current best.

- Spiral updating position: This mechanism firstly computes the distance between the humpback whale location (that represents the agent location in the search space) and the prey location (which represents the current best solution). Then, a spiral equation is performed between the whale location and the prey location to imitate the humpback whale on their spiral shaped motion. This behavior is modelled as following Eq.(5) [5]:

$$\vec{X}(t+1) = \vec{D} \cdot e^{bl} \cdot \mathrm{Cos}\,(2\pi l) + \vec{X^*}(t) \tag{5}$$

Where $\vec{D}$ denotes as the distance between the whale and the prey, b indicates the shape of a spiral, and l is a random number in [−1, 1].

It has been observed that the Humpback whales swim around the prey within a shrinking range and along a helix shaped path at the same time to swoop it. Thus, two probabilities are considered in order to choose between the Shrinking encircling mechanism and the spiral model. This idea is modelled mathematically as following Eq. (6) [5]:

$$\vec{X}(t+1) = \begin{cases} \vec{X} * (t) - \vec{A}.\vec{D} & \text{if } p < 0.5 \\ \vec{D} \cdot e^{bl} \cdot \mathrm{Cos}\,(2\,\pi\, l) + \vec{X} * (t) & \text{if } p \geq 0.5 \end{cases} \tag{6}$$

Where $\vec{A}$ plays an important role in order to guarantee the exploration phase of WOA, the other search agents update their locations based on the best solution when $\vec{A}$ >1 which means the algorithm is still in the exploration phase. The following Eq. (7) and Eq. (8) represent the mathematical model of this idea [5]:

$$\vec{D} = |\, \vec{C} \cdot \overrightarrow{Xrand} - \vec{X}\, | \tag{7}$$

$$\vec{X}(t+1) = \overrightarrow{Xrand} - \vec{A} \cdot \vec{D} \tag{8}$$

Where $\overrightarrow{Xrand}$ represents a random whale which is selected from the current population.

IV. THE PROPOSED ALGORITHM (RP-WOA)

This section presents a proposed algorithm which called Requirement Prioritizations- Whale Optimization Algorithm (RP-WOA), figure.1 shows Flowchart of RP-WOA. RP-WOA aims to prioritize the given requirements for specific project. In this work, the proposed algorithm is theoretically analyzed and tested on various datasets. To get the optimal priorities for the requirements, the whale optimization algorithm is applied. Figures (2 – 5) Figures.2 present the pseudo code for "RP-WO" algorithm, Figure.3 shows **the** Fitness functions of proposed algorithm by representing the WOA algorithm since Figure.4 represents the clustering mechanism that applied in the WOA algorithm, finally Figure. 5 shows the prioritization technique inside each cluster in order to prioritize the overall requirements.

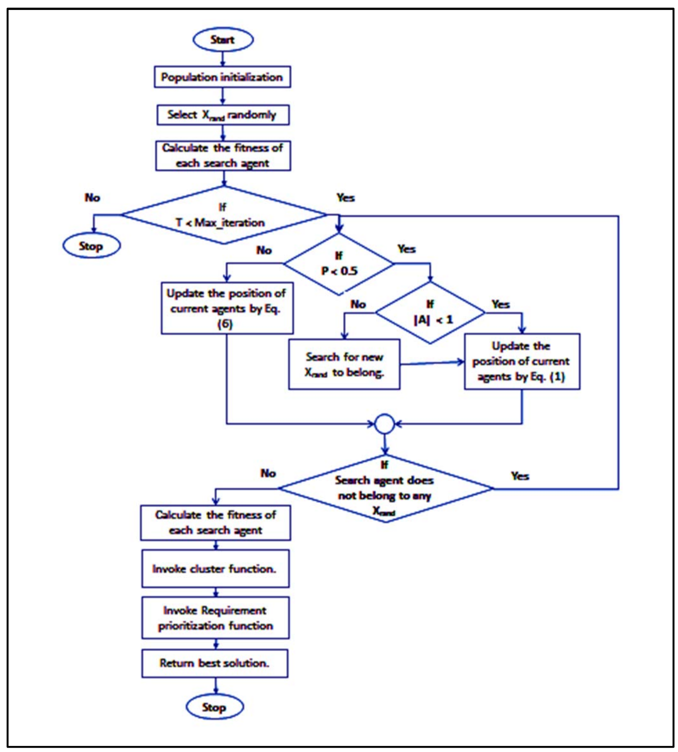

Fig.1. Flowchart of the proposed "RP-WOA"

Pseudo – Code for "RP-WOA"

Begin

1. Initialize the whales population X_i (i = 1, 2, 3…..… n) for n whales
2. Initialize C, r and a/ (C) is coefficient vector, (r) is random vector in [0, 1], and (a) is linearly decreased from 2 to 0 over the course of iterations.
3. Calculate the distance between each whale (i) and all X_{rand} by equation (1)
4. If whale (i) is not assigned
5. Assign whale (i) to its closest X_{rand}
6. Calculate the Fitness for each whale (i)
7. Invoke Cluster function
8. Invoke Requirement prioritization function
9. Return the best solution

End

Fig.2. Pseudo – Code for "RP-WOA"

A. Initialization Stage

At the first, initialize the agents' population as illustrated in Fig.2. Then, number of agents is chosen randomly in order to perform the bubble which presents a helix shape. After that, measure the distance between all Xrands and all the agents to designate to the nearest Xrand and join this cluster.

B. Fitness Function

When the X_{rand} sees the prey, it generates bubble-net to represents a cluster. Thus, all the search agents that see the bubble-net decide to join that cluster. In turn, the other search agents that see another bubble will join it to be members at that clusters. Thus, these agents that joined the clusters must update their position toward the position of Xrand as shown in Fig.3. In other meaning, these agents should update their positions based on $|\overrightarrow{A}|$ value. When $|\overrightarrow{A}| < 1$ means that agent is still joining the cluster and update its position by applying Eq. (5). Otherwise, the agent does not belong to the cluster and searches for another cluster to join by applying Eq. (8). This idea is modelled mathematically by Eq. (5, 8 and 9).

$$V = h * r * \pi \left(\frac{1}{3}\right) \qquad (9)$$

Where r indicates the radius of bubble-net which represents a constant value (r =20) [14], the height of bubble-net is denoted as h which is chosen randomly between 6 and 12 [13] and π represents a constant value that approximately equals 3.14.

Since each X_{rand} creates a bubble net in a spiral shape and this represent a cluster, so if we take a look to this spiral from 3D view with connecting each corner of that spiral by an imaginary line the result will be a cone, for that the Eq. (9) was used to represents the spiral volume by assuming this shape as a cone and this equation is used to find the cone's volume (V).

Fitness Function

1. Xrand generates bubble-net by Eq. (8)
2. For each search agent (i)
3. Update a, A, C and L; ***a*** is linearly decreased from 2 to 0 over the course of iterations, ***A*** and ***C*** are coefficient vectors and ***L*** is a random number in [−1, 1]. Compute the distance between each agent (i) and Xrand by Eq. (1)
4. If (|A|<1)
5. Update the location of the current agent (i) by the Eq. (5)
6. Else If (|A| >=1)
7. Chose new Xrand randomly
8. Update the location of the current agent (i) by the Eq. (8)
9. End If
10. End If

Fig.3. Fitness function

C. Clustering function

Each X_{rand} forms a cluster M when it sees the prey and creates bubbles. The fitness function is computed for each search agent in order to inspect if the current agent sees the bubble which created by X_{rand}. This means, that current search agent is still a member in this spiral (cluster). However, the current search agent did not view the bubble; it should search for new Xrand which means for new cluster to join. Fig.4. illustrates the pseudo-code of the cluster behavior.

Clustering Function

Begin

1. For each cluster N
2. Chose X_{rand} randomly
3. While (t< = max_iteration)
4. Invoke the fitness function
5. Invoke the RP function
6. End While
7. Return best solution
8. End for

End

Fig.4. Clustering function.

D. Requirements Prioritization Function

As mentioned in the cluster function, each search agent indicates to a requirement in terms of particular factors and weight of importance; as it refers to the importance of the requirements in the development operation of the given project. In this work, the weight of the given requirements is denoted as the ratio of cost over value that obtained from the given requirements. In other words, cost value based method is used in this study such as AHP technique because this type of technique had been found precise which relied on pairwise comparison. Thus, the arranged collection of these requirements ascending depends on the computed ratio that discussed before. Eq. (10) presents the ratio formula. Moreover Fig.5. Shows the pseudo-code of the requirements prioritization function.

$$\text{Ratio} = \text{Cost} / \text{Value} \tag{10}$$

As clearly shown in Fig.5, the RP function works by iterating on the clusters until they will be empty and in each iteration the requirement that has minimum ratio in each cluster will be chosen from that cluster as local min ratio requirement, as Eq. (11); the requirement that has minimum ratio among the set of local min ratio requirements will be prioritized as most important one and remove it from the search space in order to verify the clusters will be empty after M iterations.

$$\text{Global Min Ratio} = \text{MIN}\ \{\text{Local Min Ratio }(1), \ldots, \text{Local Min Ratio}(M)\} \tag{11}$$

Where the variable (M) indicates the number of clusters.

Requirements Prioritization Function

Begin

1. For each cluster N
2. Sort the requirements based on Eq. (10)
3. Choose the requirement which has minimum ratio
4. End For
5. Choose Minimum of minimums ratio based on Eq.(11)
6. Remove the chosen one from search space
7. End While
8. While each cluster is not empty (1,2,3,…N)

End

Fig.5. Requirements Prioritization function.

V. RESULTS AND DISCUSSION

MATLAB program based as simulation program was improved to estimate the performance of "RP-WOA" by using several dataset sizes. Moreover, the dataset size was between 100 and 1000 requirements. Each scenario was reiterated 10 experiments and increasing the data set size by 100, same as prior studies [20]. Moreover, these datasets are selected randomly in terms of cost and value for each requirement. The simulation program that used in this study was implemented on portable computer with the following specifications; Intel (R) core (TM) i7-4510U CPU with 2.40 GHz, 16 GB RAM and Windows 8.1, 64-bit operating system.

AHP technique is chosen in this work in order to compare its performance with the performance of the proposed algorithm in term of time consuming because it is one of the common and traditional techniques which utilized to prioritize the given requirements [19]. Moreover, this technique is employed because pairwise comparison based prioritization technique had been found accurate by [7, 10, 19]. Fig.6 shows the average run time in seconds for "RP-WOA" algorithm compared to AHP technique for various datasets. According to the comparison of the both techniques' results, it is obvious from Fig.6 that the proposed algorithm is fulfilled better performance than the AHP's performance, especially for large dataset size.

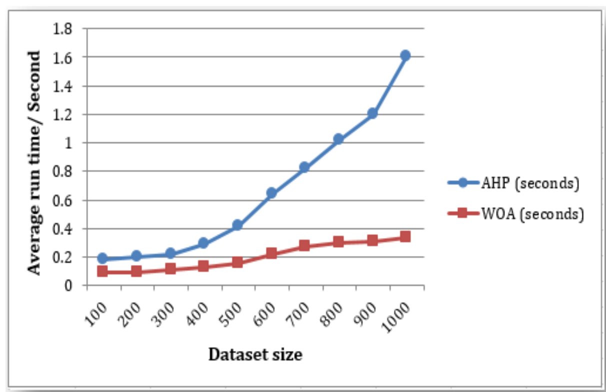

Fig. 6. Average running time for experimental results for "RP-WOA" compared with AHP technique in seconds

VI. CONCLUSION

Requirement Engineering is the basic and complex phase in software development because it contains various activities and dealing with stakeholders. While requirements prioritization is the basic and significant step in order to deliver good system with satisfying the customer needs. Based on that, in this paper the WOA as a meta-heuristic approach was used in order to prioritize the software requirements. The findings in this research show that the RP-WOA outperforms the AHP approach by approximately 40% in term of required time to prioritize different size sets of requirements.

REFERENCES

[1] I. Sommerville and P. Sawyer, Requirements engineering: a good practice guide. John Wiley & Sons, Inc.., 1997.

[2] R,B. Svensson, T. Gorschek, B. Regnell, R. Torkar, A. Shahrokni, R. Feldt and A. Aurum, "Prioritization of quality requirements: State of practice in eleven companies.", In Requirements Engineering Conference (RE), 2011 19th IEEE International, pp. 69-78, August 2011.

[3] D. Greer, & D. W. Bustard. “SERUM-Software engineering risk”, Understanding and management. The International Journal of Project & Business Risk, vol. 1(4), pp. 373-388, 1997.

[4] P. Kousalya, G. M. Reddy,S. Supraja and V. S. Prasad, Analytical “Hierarchy Process approach–An application of engineering education”, Mathematica Aeterna ,pp. 861, October 2012.

[5] S. Mirjalili, and A. Lewis. "The whale optimization algorithm." Advances in Engineering Software, pp. 51-67, May 2016.

[6] M. Vestola. “A comparison of nine basic techniques for requirements prioritization. Helsinki University of Technology.M. Young, The Technical Writer’s Handbook. Mill Valley, CA: University Science, 2010.

[7] J. Karlsson, C. Wohlin and B. Regnell, “An evaluation of methods for prioritizing software requirements”, Information and Software Technology, vol. 39, pp. 14-15, 939-947. 1998.

[8] Q. Ma, “The effectiveness of requirements prioritization techniques for a medium to large number of requirements: a systematic literature review”, Doctoral dissertation, Auckland University of Technology, 2009.

[9] S. Hatton, “Early prioritisation of goals”, In Advances in conceptual modeling – Foundations and applications , pp. 235-244, 2007.

[10] L. Karlsson, M. Host, and B. Regnell, “Evaluating the practical use of different measurement scales in requirements prioritisation”, Proceedings of the 2006 ACM/IEEE International Symposium on Empirical Software Engineering (ISESE’06), pp. 326-335, 2006.

[11] P. Berander,and A. Andrews, “Requirements prioritization”, Engineering and managing software requirements, Springer Berlin Heidelberg, pp. 69-94, January 2005.

[12] J. Karlsson and K. Ryan, “A Cost-Value approach for prioritizing requirements”, IEEE Software, 14(5), pp. 67-74. 1997.

[13] W. A. Watkins and W. E. Schevill, “Aerial observation of feeding behavior in four baleen whales: Eubalaenaglacialis, Balaenoptera borealis, Megapteranovaeangliae, and Balaenopteraphysalus”, Journal of Mammalogy, 60(1), pp. 155-163, February 1979.

[14] https://en.wikipedia.org/wiki/Humpback_whale

[15] A.K. Jain, M.N. Murty and P.J. Flynn, “Data clustering: a review”, CSUR 31 (3) , 1999.

[16] H. Emami and F. Derakhshan, “Integrating fuzzy K-means, particle swarm optimization, and imperialist competitive algorithm for data clustering”, Arab. J. Sci. Eng. 40 (12), pp. 3545–3554, December 2012.

[17] F. Yang, T. Sun and C. Zhang, “An efficient hybrid data clustering method based on K-harmonic means and particle swarm optimization”, Expert Syst. Appl. 36 (6), pp. 9847–9852, August 2009.

[18] Q.H. Zhang, B.L. Li, Y.J. Liu, L. Gao, L.J, Liu, and Shi, X.L., “Data clustering using multivariant optimization algorithm”, International Journal of Machine Learning and Cybernetics, 7(5), pp.773-782, 2016.

[19] A. Perini, A. Susi and P. Avesani, “A machine learning approach to software requirements prioritization”, IEEE Transactions on Software Engineering, 39(4), pp. 445-461, April 2013.

[20] M. Dorigo, M. Birattari, C. Blum, M. Clerc, T. Stützle and A. Winfield, “Ant Colony Optimization and Swarm Intelligence”, 6th International Conference, ANTS 2008, Brussels, Belgium, , Proceedings, Springer, Vol. 5217, pp. 22-24, September 2008.

[21] R. Masadeh, A. Alzaqebah and A, Hudaib. ‘’Grey Wolf Algorithm for Requirements Prioritization’’. Modern Applied Science, 12(2), 54. January 2018.

[22] A. Hudaib, R. Masadeh, R., M.H. Qasem, & A. Alzaqebah. ‘’Requirements Prioritization Techniques Comparison’’. Modern Applied Science, 12(2), 62. January 2018.

[23] Yassien, E., Masadeh, R., Alzaqebah, A., & Shaheen, A. (2017). Grey Wolf Optimization Applied to the 0/1 Knapsack Problem. International Journal of Computer Applications, 169(5).

[24] Masadeh R, Sharieh A, Sliet A. Grey wolf optimization applied to the maximum flow problem. International Journal of Advanced and Applied Sciences. 2017 Jul 1;4(7):95-100.

[25] R. Masadeh, A. Alzaqebah and Sharieh A. Whale Optimization Algorithm For Solving The Maximum Flow Problem. Journal of Theoretical and Applied Information Technology, 2018