

Architecture-based Optimal Software Reliability Allocation under uncertain preferences

Anu G. Aggarwal¹, Vibha Verma², Sameer Anand³

¹Department of Operational Research, University of Delhi, Delhi, India

²Department of Operational Research, University of Delhi, Delhi, India

³Shaheed Sukhdev College of Business Studies, University of Delhi, Delhi, India
¹anuagg17@gmail.com, ²vibhaverma.du.aor@gmail.com, ³sananddu@gmail.com

Abstract—Reliability Allocation in an essential task of the software development process. Increasing complexities in software structure and demand for bug free software has made Reliability Allocation a mandatory task during design and planning phase. So far in the literature several methods and models have been discussed for achieving the reliability target based on user's and developer's point of view. The crucial question that arises is 'How to allocate reliability for a Software system in an uncertain atmosphere where developer's preferences are subjective in nature?'. In this paper, we have proposed the software reliability allocation problem incorporating the decision maker's subjective uncertain preferences using Ordered Weighted Averaging (OWA) approach based on Fuzzy Analytical Hierarchical Process (FAHP). Parameter determination using FAHP through architectural hierarchy of the software system helps in interaction of user's assessment with the software engineers and programmers outlook. The OWA technique ensures complete use of available information and also avoids any kind of biasedness in reliability allocation due to overestimation of developer's inclinations. The proposed MEMV-OWA (Maximum Entropy Minimum Variance) operator is a bi-objective mathematical programming problem that maximizes entropy (deployment of information) along with minimization of the variance in weighting vector in an uncertain environment. Reliability allocation procedure for software system using the anticipated process has been discussed in detail. Also precise demonstration of the procedure has been done with an application example.

Index Terms—Software Reliability Allocation, Architecture-based hierarchy, MEMV-OWA, FAHP, Uncertain preferences

I. INTRODUCTION

THE INCREASING importance of the software systems in control and management of critical functions of various machines in daily life has steered the demand for quality software both at individual and the organizational level. Quality refers to a reliable software i.e. no bug reported by the user during the execution period under stated conditions. Buggy software can lead to loss of time, money and human lives also. The objective of error free software can be attained by maximizing the reliability therefore it is of greater importance for management to perform precise reliability allocation during the planning and design phase of the software development process. Effective reliability allocation is necessary since achieving the target reliability for a software

system involves a lot of cost. It can be defined as method of allocating the pre-defined reliability goal among the subsystems.

The concept of reliability allocation was initially used for hardware systems. Later it was adopted as an important aspect in software reliability as well. Most of the researchers in the past decades have used the architecture-based models for achievement of target reliability of a software system subject to constraints of budget, delivery time etc. Kubat [11] gave one of the initial models for software reliability allocation; he proposed a stochastic model for minimization of cost subject to the reliability constraint. Halendar et. al [9] gave RCCM (Reliability Constrained Cost Minimization) Model for the reliability allocation. Rani and Misra [14] proposed economic model that minimized cost (development and failures cost) for the optimal allocation of reliability. In these papers the reliability was allocated based on one or two criteria like minimization of cost or maximization of reliability. Later on number of researchers proposed multi-attribute based allocation methodologies by taking into consideration key software characteristics for e.g. functionality, criticality, availability, complexity, reliability etc. Zahedi and Ashrafi [20] proposed analytical hierarchical process (AHP) based software reliability allocation with objective of maximizing the user's utility based on cost and price. Leung [18] proposed the operational profile based reliability problem by using AHP and software utility function to form the software utility function defined in terms of reliability measures. Chatterjee et al. [4] established a system hierarchy combining users and developers view and adopted Fuzzy Analytical Hierarchical process (FAHP) to determine the weights required for reliability allocation. Yue et al. [18] used Dempster-Shafer theory for reliability allocation among the Multi-software in multimedia systems subject to the budget constraints.

AHP is a multi-criteria decision making framework to tackle decision maker's preferences for the alternatives. Zadeh [19] extended this into fuzzy environment to take care of uncertainty and vagueness involved in real life application.

Hence due to possible subjective preferences it is better to collect the expert's opinion in terms of Fuzzy numbers. FAHP helps to overcome the non-clarity in ratings provided by experts by combining it with Fuzzy numbers and aggregate the levels of hierarchy with each other, by incorporating the view of user as well as the one's involved in the software development process. In this paper we have proposed integrated framework combining FAHP with OWA to take care of uncertain preference of decision makers namely users, developer and programmer.

The OWA operator was first proposed by Yager [16] as tool for formation of overall decision function after aggregation of the multi-criteria. The OWA weight vector can aggregate the information provided by decision maker in imprecise and complete manner. Fuller and Majinder [7] proposed MV-OWA (Minimum variance-ordered weighted averaging) operator for obtaining weighting vector by minimizing variance for a given level of orness. O' Hagan [13] proposed ME-OWA (Maximum Entropy-ordered weighting averaging) operator for getting the weighting vector by maximizing entropy under the constraint of orness value.

OWA operator has been widely used in the field of hardware reliability for apportionment of target reliability goal among the subsystems. Chang et. al [3] did the reliability allocation for a fighter aircraft airborne radar system using the ME-OWA method. They also conducted a comparative case study to show how ME-OWA operator overcomes the shortcomings of Average weighting allocation method. Feng et. al [6] used the ME-OWA based on AHP for reliability allocation among the modules of the coal mining machine. Chen et. al [5] proposed MEMV-OWA technique based on AHP for reliability allocation for fuel cell vehicle.

In this paper we extend the concept of MEMV-OWA operator in the field of software reliability for this we have considered the architecture-based hierarchy that relates the user's view point to the software engineer's view point and further to the view of developer for a software system [4][20]. The objective is to achieve the target reliability by allocating reliability among the modules which are at the lowest level in the hierarchy. The model parameters i.e. the weights for reliability allocation are obtained through MEMV-OWA (Maximum Entropy Minimal Variance – Ordered Weighted Averaging) operator based on FAHP.

In real world always there is always possibility of some sort of uncertainty involved in preferences made and hence it is difficult for users, engineers and developers to assign precise value for their preferences. This gave the need of moving from crisp judgments to fuzzy judgements. Firstly FAHP is used for determination of weights of modules and then the proposed MEMV-OWA operator is used at developer's level of hierarchy to consider the imprecise preferences of the decision maker which may be subjective in nature. The weighting vector for modules obtained after applying the integrated

MEMV-OWA operator allocates the reliability using all the information available avoiding any kind of biasedness in making fuzzy judgements.

The paper is divided into six sections. Section II describes the MEMV-OWA operator in detail followed by description of Fuzzy Analytical hierarchical process in section III. In section IV the integrated framework of MEMV-OWA operator and FAHP for reliability allocation have been explained. In section V the proposed methodology is illustrated by an example pertaining to a multi-module software system. Finally section VI presents conclusion of the paper.

II. MATHEMATICAL BACKGROUND OF MEMV-OWA OPERATOR

A MEMV-OWA operator is used to determine the weights for reliability allocation to overcome the uncertainty involved in developer's preferences and biasedness in determination of weights. The weights are obtained by solving a bi-objective optimization model of minimizing the variance and maximization of entropy. The operator helps to maximize the use of available information (by maximization of entropy) along with reduction of biasedness in developer's inclination for different alternatives (by minimization of variance). Before we discuss the MEMV-OWA operator we will briefly describe OWA operator.

An **OWA operator** of dimension m is a function $L: \mathbb{R}^m \rightarrow \mathbb{R}$ with associated weighting vector $W = (w_1, w_2, \dots, w_m)^T \in \mathbb{R}^m$ such that it satisfies the following properties,

$$w_1 + w_2 + \dots + w_m = 1 \quad ; \quad 0 \leq w_j \leq 1 \quad j = 1, 2, \dots, m \quad (1)$$

and

$$L(k_1, k_2, \dots, k_m) = \sum_{j=1}^m w_j b_j \quad (2)$$

where b_j is the j th largest element of the aggregates k_1, k_2, \dots, k_m arranged in the descending order ($b_1 \geq b_2 \geq \dots \geq b_m$) before calculation of scalar product with W .

Yager [16] proposed characteristic measures related with weight vector of an OWA operator namely the orness of the aggregation and measure the entropy of the aggregated objects. Also Yager [17] introduced operator called the measure of variance. Now we discuss these characteristic in detail.

Orness Measure: It is defined as [16]

$$orness(W) = \sum_{j=1}^m \frac{m-j}{m-1} w_j = \gamma \quad (3)$$

and

$$orness(W) - \gamma \in [0, 1]; \quad andness = 1 - \gamma \quad (4)$$

for the weighting vector W . In decision maker's view it helps to represent the relationship between the attributes. For

instance γ being close to zero represents a higher andlike operation implying that decision maker is highly non-committal. On contrary if orness(W) is nearer to 1 then the relationship of aggregates is orlike and decision maker is highly optimistic. Similarly if decision maker has moderate assessment for attributes than $\gamma = 0.5$ and the weights will be $\frac{1}{m}$.

Entropy Measure: It is also called as measure of dispersion and is defined as [16]

$$disp(W) = -\sum_{j=1}^m w_j \ln w_j \quad (5)$$

$disp(W)$ is the degree of information utilized related to the attributes in an unspecified environment.

- If $w_j = 1$ and $w_i = 0$ ($i \neq j$) the entropy of W is minimum ($disp(W) = 0$) and hence one of the two attributes is used for the aggregation process.
- Also if $w_j = \frac{1}{m}$, $j = 1, 2, \dots, m$ the entropy of weighting vector W is maximum ($disp(W) = \ln(m)$) and hence all attributes are used in the aggregation.

Variance Measure: It is defined as [17]

$$D^2(W) = \frac{1}{m} \sum_{j=1}^m [w_j - E(W)]^2 = \frac{1}{m} \sum_{j=1}^m w_j - \frac{1}{m^2} \quad (6)$$

It determines the stretch in values of weighting vector W for a given value of orness and thus helps in avoiding overestimation of any single attribute over other attributes. This makes involvement of attributes in decision making a totally fair process.

In this paper we used the anticipated characteristic of OWA operator to form a bi-objective mathematical programming program that incorporates in itself the benefits of these characteristics [5].

Firstly we discuss the ME-OWA operator proposed by O'Hagan[13]. This operator maximizes the entropy under orness value constraint. The approach was as follows:

$$Max \quad -\sum_{j=1}^m w_j \ln w_j \quad (7)$$

$$subject \ to \ \sum_{j=1}^m \frac{m-j}{m-1} w_j = \gamma \quad (0 \leq \gamma \leq 1) \quad (8)$$

$$\sum_{j=1}^m w_j = 1, \quad 0 \leq w_j \leq 1, \quad j = 1, 2, \dots, m \quad (9)$$

Then Fuller and Majinder [7] solved the above optimization problem using Lagrange multiplier method and determined the optimal weighting vector. The optimal weighting vector can be calculated by using the underneath equations:

$$\ln w_j = \frac{j-1}{m-1} \ln w_m + \frac{m-j}{m-1} \ln w_1 \quad (10)$$

$$w_j = \frac{m-1}{\sqrt{w_1^{m-j} w_m^{j-1}}} \quad (11)$$

and

$$w_m = \frac{((m-1)\gamma - m)w_1 + 1}{(m-1)\gamma + 1 - mw_1} \quad (12)$$

then

$$w_1 [(m-1)\gamma + 1 - mw_1]^m = ((m-1)\gamma)^{m-1} [(m-1)\gamma - m] w_1 + 1 \quad (13)$$

Also Fuller and Majinder [8] proposed the MV-OWA operator to minimize variance of W the weighting vector associated with the attributes under the orness value constraint. The approach is as followed:

$$Minimize \quad \frac{1}{m} \sum_{j=1}^m w_j^2 - \frac{1}{m^2} \quad (14)$$

$$subject \ to \ \sum_{j=1}^m \frac{m-j}{m-1} w_j = \gamma, \quad (0 \leq \gamma \leq 1) \quad (15)$$

$$\sum_{j=1}^m w_j = 1, \quad 0 \leq w_j \leq 1, \quad j = 1, 2, \dots, m \quad (16)$$

Now we mathematically depict the MEMV-OWA model used in the present paper:

$$Maximize \quad -\sum_{j=1}^m w_j \ln w_j \quad (17)$$

$$Minimize \quad \frac{1}{m} \sum_{j=1}^m w_j^2 - \frac{1}{m^2} \quad (18)$$

$$subject \ to \ \sum_{j=1}^m \frac{m-j}{m-1} w_j = \gamma, \quad (0 \leq \gamma \leq 1) \quad (19)$$

$$\sum_{j=1}^m w_j = 1, \quad 0 \leq w_j \leq 1, \quad j = 1, 2, \dots, m \quad (20)$$

This is bi-objective problem with orness level as the constraint. The above problem can be solved in two steps. Firstly ideal point method (IPM) is employed to translate the bi-objective to a single objective and then lagrange multiplier method is used to determine the weights. Chen et. Al [5] has solved this problem considering that the two objectives given in equation (7),(14) that have their optimal values as g_1^0 and g_2^0 . Then IPM can be implemented as:

$$Minimize: \quad ||G - G^0|| = |g_1^0 - g_1| + |g_2^0 - g_2| - \left(g_1^0 - \frac{1}{m} \sum_{j=1}^m w_j^2 - \frac{1}{m^2} \right) + \left(g_2^0 - \sum_{j=1}^m w_j \ln w_j \right) \quad (21)$$

Applying lagrange multiplier method assuming ≥ 3 , we get;

$$L(W, \lambda_1, \lambda_2) = \left(g_1^0 - \frac{1}{m} \sum_{j=1}^m w_j^2 - \frac{1}{m^2} \right) + \left(g_2^0 - \sum_{j=1}^m w_j \ln w_j \right) + \lambda_1 \left(\sum_{j=1}^m \frac{m-j}{m-1} w_j - \gamma \right) + \lambda_2 \left(\sum_{j=1}^m w_j - 1 \right) \quad (22)$$

where λ_1 and λ_2 are real numbers.

The weighting vector for $m = 4$ calculated by Chen et. al. [5] for different orness levels has been shown in Table 1.

Table 1: Weighting Vector for Maximum Entropy Minimum Variance Ordered Weighted Averaging Operator

w_j	0.5	0.6	0.7	0.8	0.9	1
w_1	0.2500	0.3466	0.4580	0.5898	0.7568	1
w_2	0.2500	0.2732	0.2800	0.2619	0.1945	0
w_3	0.2500	0.2138	0.1659	0.1067	0.0407	0
w_4	0.2500	0.1664	0.0961	0.0416	0.0080	0

III. MATHEMATICAL BACKGROUND OF FAHP

The proposed MEMV-OWA approach is based on FAHP since reliability allocation is to be implemented on a software system with a structured architectural hierarchy. The hierarchy has been discussed later in the paper. Firstly FAHP is used to allocate the reliability among the attributes through the proposed hierarchy and then MEMV-OWA operator is applied to deal with the uncertain preferences of the software developer.

AHP introduced by Thomas Saaty [15] is a multi-criteria decision framework based on the crisp judgements, but since the real world problems encompass a lot of uncertainty and complexity, there is a need to move towards fuzzy environment.

Now we discuss briefly the mathematical concepts involved in the study of FAHP:

Fuzzy Set: A fuzzy set assigns to each object of its class a grade ranging between 0 and 1 through a membership function. Let A be the fuzzy subset of universal set Z for $z \in Z$. Then $\mu_A(z) \in [0,1]$ called membership function represents membership of z to A .

Fuzzy Number theory: The theory of Fuzzy set was initially given by Zadeh [19] to incorporate in itself the ambiguity involved in decisions due to inaccuracy in the data and the system.

Definition: A Fuzzy number ' A ' is the normal and convex subset of the universe fuzzy set Z i.e. mathematically $\forall z_1, z_2 \in Z$ and $\forall \lambda \in [0,1]$ [19]

$$\mu_A(\lambda z_1 + (1 - \lambda)z_2) \geq \min(\mu_A(z_1), \mu_A(z_2))$$

where $\mu_A(z) \in [0,1]$ is the membership function of A .

Triangular Fuzzy number: A Fuzzy number $A = (l, m, u)$ is defined as Triangular Fuzzy number (TFN) (Kauffmann and Gupta 1991) if its membership function $\mu_A(z)$ is equal to:

$$\mu_A(z) = \begin{cases} \frac{z-l}{m-l} & l \leq z \leq m \\ \frac{u-z}{u-m} & m \leq z \leq u \\ 0 & \text{otherwise} \end{cases} \quad (23)$$

Graphically TFN has been presented in Figure 1.

The operation laws of for addition, multiplication and inverse are defined as follows:

$$\begin{aligned} (l_1, m_1, u_1) + (l_2, m_2, u_2) &= (l_1 + l_2, m_1 + m_2, u_1 + u_2) \\ (l_1, m_1, u_1) \times (l_2, m_2, u_2) &= (l_1 \times l_2, m_1 \times m_2, u_1 \times u_2) \\ (l_1, m_1, u_1)^{-1} &= \left(\frac{1}{u_1}, \frac{1}{m_1}, \frac{1}{l_1}\right) \end{aligned}$$

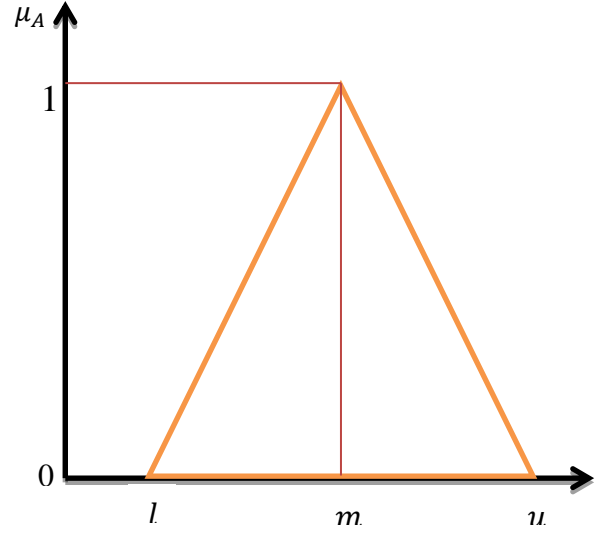


Figure 1: Triangular Fuzzy Number

The preferences of decision makers are expressed through triangular fuzzy numbers. The fuzzy scale is shown in Table 2. Then the goal is to finally obtain relative weights of modules and thus allocate reliability. The FAHP is used for the purpose of combining the user, engineers and development view of assessment. The steps involved in FAHP are:

- 1) Develop a hierarchy after defining the goal of the problem.
- 2) Establishment of fuzzy judgement matrix and the fuzzy weighting vector by employing pairwise comparisons among elements.
- 3) Relative weights of elements are aggregated and optimal one is selected by obtaining the overall rating of the element.

Table 2: Fuzzy Scale

Linguistic Scale	Triangular scale	Triangular fuzzy reciprocal scale
Just Equal	(1,1,1)	(1,1,1)
Equally Important	(1/2,1,3/2)	(2/3,1,2)
Weakly Important	(1,3/2,2)	(1/2,2/3,1)
Moderately Important	(3/2,2,5/2)	(2/5,1/2,2/3)
Moderately More Important	(2,5/2,3)	(1/3,2/5,1/2)
More Important	(5/2,3,7/2)	(2/7,1/3,2/5)
Strongly Important	(3,7/2,4)	(1/4,2/7,1/3)
Strongly More Important	(7/2,4,9/2)	(2/9,1/4,2/7)

It is important to test the consistency of all the pairwise comparison matrices for calculation of weights. It can be done easily by following the defuzzification process. The triangular fuzzy matrix is firstly defuzzified and then on the same lines as for consistency check in traditional AHP's consistency is checked for the defuzzified matrix. The matrix is accepted and we proceed for weight vector calculation if consistency is less than 0.1. For a given TFN $A = (l, m, u)$ the defuzzified real

number is $= \frac{l+2m+u}{4}$. Using the obtained real numbers the consistency is calculated according to underneath equations:

$$CI = \frac{\lambda_{max} - k}{k - 1}$$

$$CR = \frac{CI}{RI}$$

where k is the order of the pairwise matrix, RI is random index and CI is consistency index.

In this paper we follow the Chang's extent analysis [2] for the determination of parameters although many other ways have been developed in literature for the purpose. Let $Z = (z_1, z_2, \dots, z_m)$ be an object set and $X = (x_1, x_2, \dots, x_n)$ be the goal set. Following the Chang's extent analysis approach each object is taken one by one and the analysis is performed over every goal individually. Therefore n values of extent analysis are obtained for each object. i.e. $A_{x_i}^1, A_{x_i}^2, \dots, A_{x_i}^n$ where $A_{x_i}^j$ are TFN's, i is $1, 2, \dots, m$ and $j = 1, 2, \dots, n$.

The steps of performing the Chang's extent analysis are:

1. The Fuzzy synthetic extent value of i th object for n extent values is given as

$$S_i = \sum_{j=1}^n A_{x_i}^j \times \left[\sum_{i=1}^m \sum_{j=1}^n A_{x_i}^j \right]^{-1} \quad (24)$$

where

$$\sum_{j=1}^n A_{x_i}^j = \left(\sum_{j=1}^n l_j, \sum_{j=1}^n m_j, \sum_{j=1}^n u_j \right),$$

$$\sum_{i=1}^m \sum_{j=1}^n A_{x_i}^j = \left(\sum_{i=1}^m l_i, \sum_{i=1}^m m_i, \sum_{i=1}^m u_i \right) \text{ and}$$

$$\left[\sum_{i=1}^m \sum_{j=1}^n A_{x_i}^j \right]^{-1} = \left(\frac{1}{\sum_{i=1}^m u_i}, \frac{1}{\sum_{i=1}^m m_i}, \frac{1}{\sum_{i=1}^m l_i} \right)$$

2. The degree of possibility of $A_1 \geq A_2$ is given as:

$$V(A_1 \geq A_2) = \sup_{z_1 \geq z_2} [\min(\mu_{A_1}(z_1), \mu_{A_2}(z_2))] \quad (25)$$

$$V(A_1 \geq A_2) = 1 \quad \text{if } \mu_{A_1}(z_1) = \mu_{A_2}(z_2) = 1 \quad (26)$$

Since A_1 and A_2 are fuzzy numbers (normal and convex by nature), Therefore we have:

$$V(A_1 \geq A_2) = 1 \quad \text{iff } m_1 \geq m_2 \quad (27)$$

$$V(A_1 \geq A_2) = hgt(A_1 \cap A_2) \mu_{A_1}(d) \quad (28)$$

where d represents the ordinate of point D and V is given as (Figure 2)

$$= \frac{l_1 - u_2}{(m_2 - u_2) - (m_1 - l_1)} \quad (29)$$

It is the point of highest intersection between μ_{A_1} and μ_{A_2} .

Both $V(A_1 \geq A_2)$ and $V(A_2 \geq A_1)$ are required for the comparison.

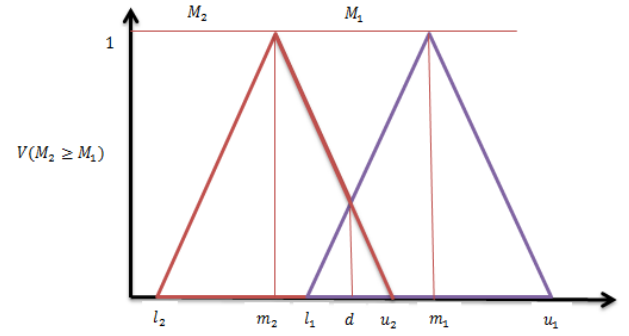


Figure 2: The intersection between M_1 and M_2 .

The possibility degree for a fuzzy number to be greater than l fuzzy numbers $A_i (i = 1, 2, \dots, l)$ is given as:

$$V(A \geq A_1, A_2, \dots, A_l) = V[(A \geq A_1) \text{ and } (A \geq A_2) \text{ and } \dots (A \geq A_l)] = \min V(A \geq A_i) \quad i = 1, 2, \dots, l \quad (30)$$

It is assumed that $d'(A_i) = \min V(S_i \geq S_j)$ for $l = 1, 2, \dots, m$ $l \neq i$.

Therefore weight vector is $W' = (d'(A_1), d'(A_2), \dots, d'(A_m))^T$.

The obtained weight vector needs to be normalized. The normalized weights are determined by dividing the each value of vector W' by the sum of values in W' . Mathematically the normalized weighting vector is given as:

$$= \left(\frac{d'(A_1)}{d'(A_1) + d'(A_2) + \dots + d'(A_m)}, \frac{d'(A_2)}{d'(A_1) + d'(A_2) + \dots + d'(A_m)}, \dots, \frac{d'(A_m)}{d'(A_1) + d'(A_2) + \dots + d'(A_m)} \right)$$

IV. METHODOLOGY OF RELIABILITY ALLOCATION

There is always a possibility of some sort of uncertainty, subjectivity and incompleteness involved in the preferences made by developers in real situations. Therefore it is not just enough to use traditional AHP's for reliability allocation. Practically it is difficult for users, engineers and developers to assign a specific value of importance for functions, programs and modules respectively. Therefore one moved to fuzzy conclusion from the crisp conclusion.

In this paper MEMV-OWA operator based on FAHP has been used for reliability allocation of a software system. Firstly the weights for the modules at the fifth level of hierarchy are derived using FAHP then MEMV-OWA operator is applied at the developer's level i.e. at the fourth level of hierarchy to combat the uncertainty involved in the developer's decision during making preferences for modules. Step by step procedure for the methodology is as follows:

- 1) Set reliability target for the software system.
- 2) Build the Software hierarchical model for the reliability apportionment process.
- 3) Construct the fuzzy pairwise comparison matrix for each level in hierarchy.

- 4) Calculate the λ_{max} (maximum Eigen value) and the corresponding normalized Eigen vectors after defuzzification of the Triangular Fuzzy numbers.
- 5) Check the consistency of the comparison matrix. If consistency not satisfied (i.e. $CI > 0.1$) then repeat steps 3 and 4.
- 6) On the basis of orness value γ obtain the OWA weight $W = (w_1, w_2, \dots, w_m)$ at developers level.
- 7) Obtain the value of OWA operator L using equation (2).
- 8) At last the reliability allocated to module is calculated according to the equation (37).

Architectural Hierarchy of the Software System

In our model we have developed the hierarchy based on [4][20] that links the user's view about functionalities to the software engineer's view of programs and further it is linked to the developer's or programmer's view of modules. The hierarchy uses the top-down approach for the reliability apportionment by combining the users, software engineers and programmer's assessment.

The first level of hierarchy is the overall system reliability target (R) based on user's view of various functionalities of software. The software system is developed to perform certain functions (given at second level of hierarchy) as per user's expectation. This level represents the user's view in the software system. The functions are denoted by $F_i (i = 1, 2, \dots, f)$ and third level of hierarchy consists of programs designed by the software engineers (SE) to accommodate the various functions specified by the user. It is denoted by P_j 's ($j = 1, 2, \dots, p$). While this level is from the SE's point of view. At the fourth level i.e. the last level of hierarchy is the modules written by programmers representing programmers view. It is denoted by $M_k (k = 1, 2, \dots, m)$. The hierarchy has been depicted diagrammatically through Figure 3. In the network we can observe that a program and a module may be connected to more than one functions and program respectively i.e. there exist many-to-many connections between them. In other words we can say that a program designed can be beneficial in performing one or more than one function simultaneously. Similarly a module can also be connected to one or more than one programs. The modules are independent in nature but they may have submodules with one-to-one connection.

The aim is to find the relative weights of each function F_i at the second level, program P_j at the third level and module M_k at the fourth level of hierarchy from the knowledge of reliability goal set by the user for the software system based on their assessment of the functions. In this paper we have calculated the alternative weights of modules considering the uncertain preferences of the developer while making decisions using Minimum Variance Maximum Entropy – Ordered weighted averaging operator.

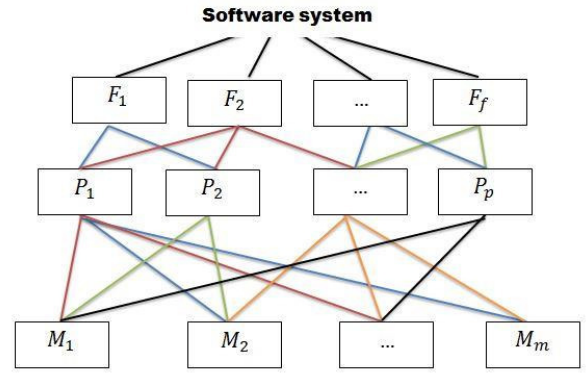


Figure 3: Architectural hierarchy of the software system

Relative Reliability Weight of Module

Once the architectural hierarchy has been formed and the reliability target is set, the next step is to form pairwise comparison matrix based on user's opinion. The data has been collected in terms of fuzzy numbers. Let it be given as follows:

$$\begin{bmatrix} (1,1,1) & (l_{12}, m_{12}, u_{12}) & \dots & (l_{1f}, m_{1f}, u_{1f}) \\ (l_{21}, m_{21}, u_{21}) & (1,1,1) & \dots & \dots \\ \dots & \dots & \dots & \dots \\ (l_{n1}, m_{n1}, u_{n1}) & (l_{n2}, m_{n2}, u_{n2}) & \dots & (1,1,1) \end{bmatrix} \quad (31)$$

where $(l_{ij}, m_{ij}, u_{ij}) = \left(\frac{1}{u_{ji}}, \frac{1}{m_{ji}}, \frac{1}{l_{ji}} \right)$ for $i, j = 1, 2, \dots, n$ and $n \neq j$

At the second level users were asked to compare the functions (F_i 's) pairwise. The total number of pairwise evaluation will be $\frac{f!}{2 \times (f-2)!}$ where ' f ' the total number of functions specified by the user. Similarly we go for the other levels of the hierarchy. The results of comparison of functions are stored in $A_f = [a_{fij}]$ a $f \times f$ matrix. The element a_{fij} gives the importance of i element over the j th element, the diagonal elements are obviously (1,1,1) and the lower part of the matrix are inverse of the corresponding upper triangular matrix elements. As mentioned earlier the relative weights at this stage are derived using Chang's extent analysis [1996].

Let relative weights for Function be denoted as $(WF_1, WF_2, \dots, WF_f)$. Allocated reliability for the function provided they are executed every time is given by [1].

$$RF_i = (R)^{WF_i} \quad (32)$$

Similarly the input matrix A_p and A_m are formed for programs and modules by taking opinion from the system engineer and programmer. Let the relative weight matrix for the program may be given as follows:

$$\begin{bmatrix} WP_1^1 & WP_2^1 & \dots & WP_p^1 \\ WP_1^2 & WP_2^2 & \dots & WP_p^2 \\ \dots & \dots & \dots & \dots \\ WP_1^f & WP_2^f & \dots & WP_p^f \end{bmatrix} \quad (33)$$

WP_j^i 's are the relative weights at the software engineer's level indicating weight for the j th program of the i th function. At the developer's level of hierarchy we need p number of pairwise matrices for comparison. The target reliability at this level will be RF_i for the programs connected to the function i . For the i th function the relative weights of related programs are given by

$$WP^i = (WP_j^i) \quad \forall i = 1, 2, \dots, f \text{ and } j \in P \\ \subset \{1, 2, \dots, p\} \text{ program related to } F_i\}$$

Then the reliability allocated to the programs is

$$RP_j^i = (RF_i)^{WP_j^i} \quad \forall i = 1, 2, \dots, f \quad j \in P \subset \\ \{1, 2, \dots, p\} \text{ program connected to } F_i\} \quad (34)$$

Doing this we obtain different reliability values of same program, since a function can be connected to more than one function. Therefore it will be best to choose the one with highest reliability.

$$RP_j = \text{Maximum}(RP_j^i) \quad \forall j = 1, 2, \dots, p \text{ and } i \in F \\ \subset \{1, 2, \dots, f\} \\ \text{for all functions in which } j \text{th program is called} \} \\ (35)$$

Here RP_j is the allocated reliability of the j th program. Similar approach is used for other levels of the hierarchy. At fourth level the reliability value allocated to modules is calculated.

We further calculate the weights of modules as follows:

$$WM^j = (WM_k^j) \quad \forall j = 1, 2, \dots, p \text{ and } k \in M \subset \\ \{1, 2, \dots, m\} \text{ for the module related to } P_j\}$$

Hence the reliability allocated to the programs is

$$RM_k^j = (RP_j)^{WM_k^j} \quad \forall j = 1, 2, \dots, p \quad k \in M \subset \\ \{1, 2, \dots, m\} \text{ module connected to } P_j\} \quad (36)$$

Doing this we obtain different reliability values of same module, since a module can be connected to more than one program. Therefore it will be best to choose the one with highest reliability.

$$RM_k = \text{Maximum}(RM_k^j) \quad \forall k = 1, 2, \dots, m \text{ and } j \in P \\ \subset \{1, 2, \dots, p\} \\ \text{for all programs in which } k \text{th module is called} \} \\ (37)$$

Hence RF_j , RP_k and RM_l are allocated reliability values of the functions, programs and modules respectively.

Relative weight of modules applying MEMV-OWA operator at program level of hierarchy

In section two of this paper we have shown the weight vector of MEMV-OWA operator against different orness values ranging from 0.5 to 1 (Table 1). Now these weights are used to obtain the final weights assigned to each module at the lowest level of hierarchy. The OWA function L_i is given as:

$$L_i(WP_1, WP_2, \dots, WP_p) = \sum_{k=1}^p w_k \times b_{ik} \quad (38)$$

where L_i is the weight of the i th module and b_{ik} is the k th largest element in vector $(WM_1, WM_2, \dots, WM_m)$ for a program. Similarly OWA weights for other modules are calculated. The obtained weight incorporates in itself the developer's uncertain preferences which are always subjective to outside fact.

V. APPLICATION EXAMPLE OF PROPOSED RELIABILITY ALLOCATION MODEL

We exhibit the FAHP based MEMV-OWA technique of Reliability apportionment with an example. Let us consider software architecture with three functions (F_1, F_2, F_3). The software engineer and programmers established four programs (P_1, P_2, P_3 and P_4) and six modules ($M_1, M_2, M_3, M_4, M_5, M_6$) respectively to accomplish the functions expected by the users. The linkages between functions, programs and modules are as follows:

- 1) Software engineers designs the program to execute the functions expected by the user. P_1, P_2 and P_4 serve the function F_1 , P_1, P_2 and P_3 serve function F_2 and P_2, P_3 and P_4 serve function F_3 .
- 2) Further developers or programmers use six modules to implement the 4 programs. P_1 appeals for modules M_1, M_2 and M_5 , P_2 appeals for modules M_2, M_3 and M_4 , P_3 appeals for modules M_4, M_5 and M_6 and program P_4 appeals for M_1, M_3, M_6 .

The target reliability for the software system is set to 0.90 based on the necessities of users. The total number of triangular fuzzy pairwise comparison matrices at different levels of hierarchy is 1, 3 and 4 respectively. After obtaining weights of modules using FAHP we apply the MEMV-OWA operator at software developer's level to find the weights for modules and thus allocate the reliability.

Based on the fuzzy scale given in Table 2 the fuzzy pairwise comparison matrices have been constructed according to views of users, SE (Software Engineers) and developers.

Reliability allocation for Functions:

Fuzzy matrix obtained for system after the comparison of functions is given by:

$$\begin{matrix} & F_1 & F_2 & F_3 \\ \begin{matrix} F_1 \\ F_2 \\ F_3 \end{matrix} & \begin{pmatrix} (1,1,1) & (3/2,2,5/2) & (3/2,2,5/2) \\ (2/5,1/2,2/3) & (1,1,1) & 2/3,1,2) \\ (2/5,1/2,2/3) & (1/2,1,3/2) & (1,1,1) \end{pmatrix} \end{matrix}$$

Now we use the FAHP method discussed earlier to determine the relative weights for the functions and thus the corresponding reliabilities. The weights and reliabilities so determined are as follows:

The normalized weighting vector is $WF = (0.523077, 0.271111, 0.205805)$ and reliabilities allocated to functions using expression (32) are:

$$\begin{aligned} RF_1 &= (R)^{WF_1} = 0.946380697 \\ RF_2 &= (R)^{WF_2} = 0.9718309859 \\ RF_3 &= (R)^{WF_3} = 0.97854954 \end{aligned}$$

The reliability allocated to a function is used as the reliability goal to allocate reliability among the programs associated with that particular function.

Reliability Allocation for Programs:

Fuzzy Pairwise comparison matrix for programs associated with function F_1 are :

$$\begin{matrix} & P_1 & P_2 & P_4 \\ \begin{matrix} P_1 \\ P_2 \\ P_4 \end{matrix} & \begin{pmatrix} (1,1,1) & (3/2,2,5/2) & (1,3/2,2) \\ (2/5,1/2,2/3) & (1,1,1) & (2/5,1/2,2/3) \\ (1/2,2/3,1) & (3/2,2,5/2) & (1,1,1) \end{pmatrix} \end{matrix}$$

The normalized weighting vector is $WF_1 = (0.428928, 0.142145, 0.428928)$ and reliabilities allocated to programs of function F_1 using expression (34) are:

$$\begin{aligned} RP_1 &= (RF_1)^{WP_1} = 0.976635514 \\ RP_2 &= (RF_1)^{WP_2} = 0.9921962096 \\ RP_4 &= (RF_1)^{WP_4} = 0.976635514 \end{aligned}$$

For Function F_2

$$\begin{matrix} & P_1 & P_2 & P_3 \\ \begin{matrix} P_1 \\ P_2 \\ P_3 \end{matrix} & \begin{pmatrix} (1,1,1) & (2/3,1,3/2) & (2/9,1/4,2/7) \\ (2/3,1,3/2) & (1,1,1) & (2/5,1/2,2/3) \\ (7/2,4,9/2) & (3/2,2,5/2) & (1,1,1) \end{pmatrix} \end{matrix}$$

The normalized weights are $WF_2 = (0.41629, 0.291855, 0.291855)$

$$RP_1 = (RF_2)^{WP_1} = 0.988175676$$

$$\begin{aligned} RP_2 &= (RF_2)^{WP_2} = 0.991694352 \\ RP_3 &= (RF_2)^{WP_3} = 0.991694352 \end{aligned}$$

For function F_3

$$\begin{matrix} & P_2 & P_3 & P_4 \\ \begin{matrix} P_2 \\ P_3 \\ P_4 \end{matrix} & \begin{pmatrix} (1,1,1) & (2/3,1,3/2) & (2/7,1/3,2/5) \\ (2/3,1,3/2) & (1,1,1) & (2/5,1/2,2/3) \\ (5/2,3,7/2) & (3/2,2,5/2) & (1,1,1) \end{pmatrix} \end{matrix}$$

The normalized weights are $WF_3 = (0.423036, 0.288482, 0.288482)$

$$\begin{aligned} RP_2 &= (RF_3)^{WP_2} = 0.99086758 \\ RP_3 &= (RF_3)^{WP_3} = 0.993762994 \\ RP_4 &= (RF_3)^{WP_4} = 0.99372994 \end{aligned}$$

Since a program written is for accomplishment of more than one function of the software so the final reliability allocated to a program using expression (35):

$$\begin{aligned} RP_1 &= \text{Maximum}(0.976635514, 0.988175676) = 0.988175676 \\ RP_2 &= \text{Maximum}(0.9921962096, 0.991694352) = 0.992196209 \\ RP_3 &= \text{Maximum}(0.991694352, 0.993762994) = 0.993762994 \\ RP_4 &= \text{Maximum}(0.976635514, 0.993762994) = 0.993762994 \end{aligned}$$

Weight calculation for Modules:

We only determine weights for modules using FAHP method because reliability allocation for modules is to be done using the weights obtained from MEMV-OWA technique.

For program P_1 the fuzzy pairwise comparison matrix is given as:

$$\begin{matrix} & M_1 & M_2 & M_5 \\ \begin{matrix} M_1 \\ M_2 \\ M_5 \end{matrix} & \begin{pmatrix} (1,1,1) & (2/7,1/3,2/5) & (2/7,1/3,2/5) \\ (2/3,1,3/2) & (1,1,1) & (2/5,1/2,2/3) \\ (5/2,3,7/2) & (3/2,2,5/2) & (1,1,1) \end{pmatrix} \end{matrix}$$

The normalized weights are $WP_1 = (0.3520129, 0.3520129, 0.295943)$

For Program P_2

$$\begin{matrix} & M_2 & M_3 & M_4 \\ \begin{matrix} M_2 \\ M_3 \\ M_4 \end{matrix} & \begin{pmatrix} (1,1,1) & (2,5/2,3) & (5/2,3,7/2) \\ (1/3,2/5,1/2) & (1,1,1) & (1/2,2/3,1) \\ (2/7,1/3,2/5) & (1,3/2,2) & (1,1,1) \end{pmatrix} \end{matrix}$$

The normalized weights are $WP_2 = (0.248739, 0.248739, 0.502523)$

For program P_3

$$\begin{matrix} & M_4 & M_5 & M_6 \\ M_4 & (1,1,1) & (2/7,1/3,2/5) & (1,3/2,2) \\ M_5 & (5/2,3,7/2) & (1,1,1) & (7/2,4,9/2) \\ M_6 & (1/2,2/3,1) & (2/9,1/4,2/7) & (1,1,1) \end{matrix}$$

The normalized weights are $WP_3 = (0.3520129, 0.3520129, 0.295943)$

For program P_4

$$\begin{matrix} & M_1 & M_3 & M_6 \\ M_1 & (1,1,1) & (2/9,1/4,2/7) & (1,3/2,2) \\ M_3 & (7/2,4,9/2) & (1,1,1) & (7/2,4,9/2) \\ M_6 & (1/2,2/3,1) & (2/9,1/4,2/7) & (1,1,1) \end{matrix}$$

The normalized weights are $WP_4 = (0.345638, 0.345638, 0.308725)$

Consistency of each pairwise matrix has been checked by using defuzzification process discussed earlier. The essential details of the consistency have been shown in Table 3.

Table 3: Consistency check for pairwise comparison matrices

Matrix	λ_{max}	n (order of the matrix)	CI	RI	CR
System	3.0764180613	3	0.0382093	0.58	0.065877639
F_1	3.058563675	3	0.029282838	0.58	0.050485927
F_2	3.098245614	3	0.049122088	0.58	0.084693255
F_3	3.063174114	3	0.031586998	0.58	0.054460342
P_1	3.02875	3	0.01437526	0.58	0.024784931
P_2	3.06779661	3	0.033901652	0.58	0.058451124
P_3	3.02875	3	0.01437526	0.58	0.024784931
P_4	3.0417101	3	0.020855165	0.58	0.035957182

The weight vector for different modules corresponding to a program obtained using FAHP is shown in Table 4.

Table 4: Modular weights obtained using FAHP

Allocated Reliability to program	Module	P_1	P_2	P_3	P_4
	M_1	0.3520129	—	—	0.345638
	M_2	0.3520129	0.248739	—	—
	M_3	—	0.248739	—	0.345638
	M_4	—	0.502523	0.3520129	—
	M_5	0.295943	—	0.3520129	—
	M_6	—	—	0.295943	0.308725

Now MEMV-OWA operator is applied at the software developer's level in the hierarchy to obtain the weights for the modules and thus allocate the reliability. The weights for the OWA operator are given in Table 1 for different orness values. We have assigned weight zero for modules which are not connected to the program and then evaluated the relative weights of the modules for different orness values i.e. from 0.5 to 1. The weights obtained at different orness levels are shown in Figure 4, the graphical representation compares modular weights obtained at different orness levels. This clearly

implies that the subjective attitude of the decision maker affects the weights assigned for allocation.

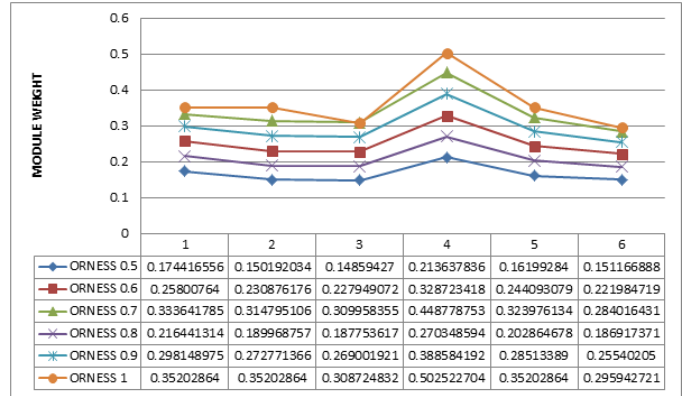


Figure 4: Comparison of Module weights for different orness levels

Now using the weights obtained after application of MEMV-OWA operator we find the reliability allocated to each module under different orness i.e. under different possible perspectives of the developer in an uncertain environment. The graphical representation of allocated reliability is shown in Figure 5. We can see that it is important to take into consideration the developer's behavior. The developer should neither be over-optimistic nor under optimistic. Reliability allocated for orness value is 1 whereas it is maximum for the average optimism behavior i.e. considering every alternative to be equally important.

After obtaining the relative weights for module we allocate reliability using expression (36). Since a module is serving more than one program, therefore maximum obtained reliability is allocated to the module using expression (37).

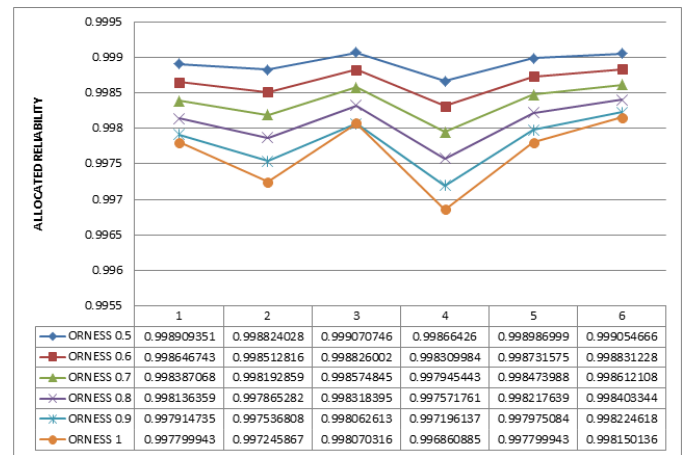


Figure 5: Comparison of Reliability Allocated at different orness levels

VI. CONCLUSIONS

This paper is contributing to the reliability allocation literature of software systems by introduction of a FAHP based MEMV-OWA technique by considering the uncertain preferences of the developer during the planning and design phase. The bi-objective MEMV-OWA technique problem

provides an unbiased way for the allocation of reliability among the modules by minimizing the variance and makes use of maximum information by maximizing the entropy. During reliability allocation in addition to the user's opinion at function level we have also incorporated the system engineers and programmers view at program and module level respectively. An example with the application of methodology has been demonstrated which makes the steps involved more clear. At different orness levels the weight and reliability allocated has been calculated to illustrate the sensitivity of optimal reliability allocation solution to the different level of agreement among the developer's preferences.

REFERENCES

- [1] K.K. Aggrawal and Y. Singh, "Software reliability Apportionment Using Analytical hierarchical process", ACM SIGSOFT software engineering notes, Vol 20, Issue 5, pp. 56-61, 1995.
- [2] D.Y. Chang, "Applications of the extent analysis method on fuzzy AHP", European Journal of Operational Research, Vol 95, Issue 3, pp. 649-655, 1996.
- [3] Yung-Chia Chang, Kuie-Hu Chang and Cheng-Shih Liaw, "Innovative reliability allocation using the maximal entropy ordered weighted averaging method", Computers and Industrial engineering, Vol 57, Issue 4, pp. 1274-1281, 2009.
- [4] S. Chatterjee, J.B. Singh and A. Roy, "A structure-based software reliability allocation using fuzzy analytical hierarchical process", International journal of systems science, Vol 46, Issue 3, pp. 513-525, 2015.
- [5] T. Chen, S. Zheng, H. Liao and J. Feng, "A Multi-Attribute reliability Allocation Method considering Uncertain Preferences", Quality and Reliability engineering international, Vol 32, pp. 2233-2244, 2016.
- [6] Y. Feng, Z. Hong, J. Cheng, G. Tian and H. Zhang, "Environment friendly reliability allocation for product platform based on expert measurement and ICN", Computers and Electrical Engineering, Vol 64, pp. 132-144 2017.
- [7] R. Fuller and P. Majinder, "An analytical approach for obtaining maximal entropy OWA operator weights". Fuzzy Sets and Systems, Vol 124, Issue 1, pp. 53-57, 2001.
- [8] R. Fuller, and P. Majinder, "On obtaining minimal variability OWA operator weights", Fuzzy sets and systems, Vol 136, pp. 203-215, 2003.
- [9] M. E. Halendar, M. Zaho and N. Ohlsson, "Planning models for software reliability and cost", IEEE Trans. Software engineering, Vol 24, Issue 6, pp. 424-434, 1998.
- [10] A. Kaufmann and M. M. Gupta, "Introduction to Fuzzy arithmetic: Theory and Applications", New York: Van Nostrand Reinhold 1986.
- [11] P. Kubat, "Assessing reliability of modular software", Operational research letters, Vol 8, Issue 1, pp.35-41, 1989.
- [12] Leung, Y.W. (1997). Software reliability allocation under an uncertain operational profile, Journal of Operational Research Society, Vol. 48, No. 4 pp. 401-411.
- [13] M. O'Hagan, "Aggregating template or rule antecedents in real time expert systems with fuzzy set logic", in: Proceedings 22nd Annual IEEE Asilomar Conference Signals, Systems and Computers, Pacific Grove, CA, Piscataway, Vol. 8, 1988.
- [14] I. Rani and R. B Misra, "Economic allocation of target reliability in modular software systems", in: Proceedings of the annual Reliability and Maintainability Symposium, pp 428-432, 2005.
- [15] T. L. Satty, "The Analytical Heirarchical Process: Planning, Priority Setting, resource Allocation", McGraw-Hill, Newyork, 1980.
- [16] R. R. Yager, " On ordered weighted averaging aggregation operators in multi-criteria decision making", IEEE Transactions on Systems, Man and Cybernetics, Vol. 18, Issue 1, 183-190, 1988.
- [17] R. R. Yager, "On inclusion of variance in decision making under uncertainty". International journal of uncertainty, Fuzziness and Knowledge-based systems, Vol. 4, Issue 5, pp. 401-419, 1996.
- [18] F. Yue, Goufu Zhang, Zhaopin Su , Yang Lu and Ting Zhang , "Multi-software reliability allocation in multimedia systems with budget constraints using Dempster-Shafer theory and improved differential evolution", Neurocomputing, Vol. 169, pp. 13-22, 2015.
- [19] L. A. Zadeh, "Fuzzy Sets", Information and control, Vol. 8, Issue 3, pp. 338-353, 1965.
- [20] F. Zahedi and N. Ashrafi "Software reliability allocation based on structure, utility, price and cost", IEEE Trans. Software engineering, Vol. 17, Issue 4 , pp. 401-411, 1991.