# OPTIMAL SOFTWARE RELIABILITY PREDICTION USING CRITERIA WEIGHTS UNDER FUZZY DECISION-MAKING APPROACH

H. D. Arora, Anjali Naithani, Surbhi Gupta
•
Department of Mathematics, Amity Institute of Applied sciences,
Amity University Uttar Pradesh, Noida, U.P., India

hdarora@amity.edu, anathani@amity.edu, sgupta11@amity.edu

**Abstract**

*Multi Criteria Decision Making techniques often face the challenge of determining criteria weights. The weights of criteria can significantly impact the outcomes of the decision-making process. Therefore, it is crucial to pay close consideration to the objectivity characteristics of criteria weights. Many weighting methods were discussed by various authors and utilized to solve various decision-making complications in Analytical Hierarchy Process (AHP), Entropy method, Weighing Score Method (WSM), Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS), Best worse method (MWM), VlseKriterijumska Optimizacija I Kompromisno Resenje (VIKOR), Criteria Importance Through Intercriteria Correlation (CRITIC) method, ELECTRE, etc. This research article gives an overview of various weighting strategies that can be used in multi-criteria optimization and proposes a novel approach to determine criteria weights using Pythagorean fuzzy sets to handle uncertainties in the decision maker's preferences for allocating software reliability. The comparative analysis shows that the proposed weighting method has the advantage of being simple and straightforward in comparison to the existing weighting methods. The evaluation confirms that this novel approach is effective enough to determine objective weights.*

**Keywords:** Criteria Weights, software reliability, MCDM, Entropy, AHP, Pythagorean fuzzy number

## 1. Introduction

Software systems have become an essential part of everyone's daily lives. The number of failures experienced by a specific user of software determines its reliability. The reliability goal is determined by the expectations of users and developers. Reliability refers to the degree to which the developments of software program enable it to perform its exact end-item use. software reliability, like functionality, usability, overall performance, serviceability, capability, maintainability, and documentation, is an important best trait. For software-primarily based systems utilized in protection-crucial packages which include computer relaying of energy transmission lines, software reliability assessment is critical. To give tools for estimating them, numerous software reliability models have been developed [1,2]. Early-stage software reliability prediction models are critical because they enable the early detection of cost overruns, issues with the software development process, and optimal development strategies [3]. Various components influence the overall operation of the software to varying degrees, and therefore information on component prominence can reveal critical information for software designers and test engineers. Additionally, software dependability is estimated utilising antique data collected and an assumed distribution curve, making it inherently unstable. This model specifies how reliable software modules and programmes must be to maximise user utility while considering the system's financial and technical constraints. The reliability goal is determined by the expectations of users and developers. The degree to which the attributes of software enable it to perform its special end-item use is known as software reliability. Along with functionality, usability, overall performance, serviceability, capability, install ability, maintainability, and documentation, software reliability is a crucial quality trait.

Our research has concentrated on the problem of reliability allocation in a software system known as software hierarchy, which includes functions, programmes, and modules [4]. In this field, allocation of reliability is a relatively unexplored domain. To address the issue of reliability allocation, techniques of mathematical programming such as maximisation of reliability, minimization of cost, and multi-criterion decision-making models have been used. Numerous decision-making approaches have been discussed by various researchers for representing complex commercial or technical procedures.

Techniques for multiple-criteria decision-making (MCDM) have recently attained amazing popularity and wide use. Several researchers have employed AHP in testing consistency of software for reliability distribution or for issues with choosing suitable module. Sitorus et al. discussed the problem of choosing suitable method for mining and mineral processing in MCDM problems [5]. Methods such as AHP [6], weighted score method [7], VIKOR-TODIM [8], TOPSIS [9], DEMATEL [10], and other weighting methods have been proposed. Applications of decision making in public transport [11], location preferences of bridges [12], industrial symbiosis [13], and evaluating sustainable manufacturing strategies [14] were also discussed in the literature. In MCDM problem, selecting a suitable weighting method is a difficult task. Pamučar proposed full consistency method in MCDM which has cater smaller number of pairwise comparisons [15]. This method allows to produce realistic criteria weight coefficient values, which helps in reasonable judgement. These weighting methods are divided into several methods for directly and indirectly weighting criteria. A simulation study is provided to evaluate the effectiveness of approaches for transforming the ranks of multiple criteria into weights in multi-criteria decision-making [16,17]. Methods of weighting might be integrative, subjective, or both. Diakoulaki et al. suggested a method for calculating objective weights that is based on the quantification of two key MCDM concepts [18]. A sample of industrial businesses are subjected to the application of the suggested approach. The results demonstrated that the strategy guarantees a better compromise of the evaluated criteria when compared to those achieved by other sets of objective weights. Chatterjee et al. applied fuzzy AHP method for calculating the weights of functions, programmes, and modules to assess the dependability of the elements during the design and model phase of a software project [19]. In industrial health and risk evaluation, a new method has been presented [20]. The fuzzy VIKOR and Pythagorean fuzzy AHP are incorporated into the risk assessment process. To convert qualitative data into quantitative, Saaty proposed a numerical scale of one to nine where one denoting the 'equal importance' and nine 'great importance [21]. FARE (Factor Relationship), a new technique for calculating the weights of the criteria based on the connections between all the criteria describing the phenomenon under consideration, is proposed [22]. An overview of several weighting approaches that can be used with multi-criteria optimization techniques is given in the work of Odu [23]. This approach concentrated on the utility of the various weighing methods for MCDM and suggested that subjective techniques are easy to compute as compared to objective one. Kasim presented subjective and objective weights methods for addressing MCDM problems applicable to real world scenario [24]. An MCDM-approach was proposed by Gupta et al. to rank a variety of SRGMs in software reliability [25]. To demonstrate the viability of the suggested strategy employing an entropy distance-based approach, an exploratory study is being conducted. In a hierarchy relating the expectations of clients, software technologists, and systems analyst for the software system, Neha et al. described an allocation approach where preferences were assigned in terms of Pythagorean fuzzy numbers [26]. An example problem served as the basis for the proposed solution. For rating and assessing the services provided in hotels, Zoraghi et al. introduced a fuzzy MCDM prototype by incorporating both subjective and objective weights [27]. In the context of sustainable energy, Sahin proposed a comprehensive and comparative analysis of weighting MCDM methods [28].

This work proposes a novel technique for estimating the objective weights of criteria based on the elimination impact of criteria. This method employs a novel concept for weighing criteria. This method determines criterion weights based on the elimination impact of each criterion on alternative performance. Criteria with a greater impact on performance are given more weight. We provide some computer assessments to demonstrate effectiveness of our method after introducing it in a systematic manner. In practise, it is difficult for even a single decision-maker to provide numerical relative weights for various decision criteria.

This paper delivers an outline of various weighting methods that can be used with multi-criteria optimization techniques. The remainder of the paper is structured as follows: In section 2, we review the pioneered methods for determining criteria weights which will be helpful in the study for comparison purposes. Proposed methodology and algorithm are described in detail in section 3. Section 4 describes the proposed method's application using a real-life case study. Section 5 explains the comparison analysis of reliability allocation between the proposed method and the existing methods of determining criteria weights. Finally, section 6 concludes the paper by discussing prospects.

# 2. Methods

## 2.1. Determination of Criteria Weights

It might be challenging to select an appropriate weighting approach to resolve a multi-criteria decision problem. Some researchers have ignored the challenge of evaluating the criteria weights because they think that all decision-makers are aware of their importance. To avoid altering the MCDM models and provide accurate model outputs, it is necessary to take the validity of the acquired criteria weights into account. employing a variety of weighting methods. By enhancing the rationality, efficiency, and clarity of decision-making process, MCDM methodologies can aid in enhancing the quality of decisions. The authors pointed out that MCDM includes a variety of decision criteria and options, is recognised as a key component of recent operational research and decision science which comprises of multiple criteria and multiple decision alternatives. The methods used to establish the criteria weights that can be categorised as subjective and objective methods. While the objective approach chooses weights by arithmetic operations that disregard the decision makers' subjective judgement data, the subjective approach selects weights exclusively based on the considerations or judgements of decision makers. Because both subjective and objective approaches have advantages and disadvantages, an integrated or combined method appears to be preferable in determining weights of criteria [29]. To solve this problem, we investigated a technique for creating MCDM for ranking utilising fuzzy logic that relied on subjective and objective weights.

We put forward few essential objective and subjective methods of determining weights for solving MCDM problems in optimum allocation of software reliability.

### 2.1.1. Entropy Method

Because the decision matrix for a group of potential materials contains a certain amount of data, the entropy approach is used to determine the weight in a specific situation. Based on a predetermined decision matrix, entropy operates. Entropy is a measure of how much uncertainty a discrete probability distribution can convey. Shannon developed an entropy approach that can be applied to determine the weights of the criteria in any MCDM problem [30]. The working algorithm based on Shannon's entropy [30] can be demonstrated in a series of steps:

Step 1: Create the decision matrix $D_{ij}$, which includes the evaluations of the options to be evaluated.

$$D_{ij} = \begin{bmatrix} d_{11} & d_{12} & \dots & d_{1n} \\ d_{21} & d_{22} & \dots & d_{2n} \\ d_{31} & d_{32} & \dots & d_{3n} \\ \dots & \dots & \dots & \dots \\ d_{m1} & d_{m2} & \dots & d_{mn} \end{bmatrix}$$

Step 2: Normalize the decision matrix $D_{ij}$ as

$$\mathcal{P}_{ij} = \frac{D_{ij}}{\sum_{j=1}^{n} D_{ij}} \tag{1}$$

Step 3: Calculate the entropy for each selection criterion $\mathfrak{E}$ as

$$\mathfrak{E}_i = \mathfrak{E}_0 \sum \mathcal{P}_{ij} \times In(\mathcal{P}_{ij}) \tag{2}$$

where $\mathfrak{E}_0$ is the entropy constant computed by $(In\, m)^{-1}$

Step 4: Compute diversity degree of the knowledge involved in the outcomes of the $i^{th}$ criteria as

$$\check{\mathrm{D}}_i = 1 - \mathfrak{E}_i \tag{3}$$

Step 5: Compute the normalised weights of the selected criteria as follows:

$$\theta_i = \frac{1-\mathfrak{E}_i}{\sum_{i=1}^{m}(1-\mathfrak{E}_i)} = \frac{\check{\mathrm{D}}_i}{\sum_{i=1}^{m} \check{\mathrm{D}}_i} \tag{4}$$

### 2.1.2. Analytical Hierarchy Process (AHP) Method

The AHP is a method that prioritises each choice by establishing the objectives or the hierarchy of relevance of attributes [31]. By condensing, dividing, and comparing numerous attributes, the AHP minimises cognitive errors. It can compare both qualitative and quantitative indices. As a result, it is frequently used in many different contexts, such as selection, evaluation, resource allocation, conflict resolution, priority and ranking, and optimization. The general procedure of finding weights using AHP is as follows:

Step 1: Creating a hierarchical structure with a goal at the top, attributes, or criteria at the second, and alternatives at the third.

Step 2: Determine the importance of various attributes or criteria in relation to the goal. A pairwise comparison matrix is created using a scale of relative importance (Table 1).

Step 3: Geometric mean (GM) suggested by Buckley [32] is employed to aggregate the pairwise comparison matrices as

$$\acute{r}ij^{\sim} = \sqrt[n]{(\acute{r}ij^{1\sim} \otimes \acute{r}ij^{2\sim} \dots \otimes \acute{r}ij^{n\sim})} \tag{5}$$

where $n$ stands for the DMs and $\acute{r}ij^{\sim}$ represents triangular values. Equation

$$\mathcal{P} \otimes \mathcal{Q} = (\mu_{\mathcal{P}}, \upsilon_{\mathcal{P}}, \gamma_{\mathcal{P}}) \otimes (\mu_{\mathcal{Q}}, \upsilon_{\mathcal{Q}}, \gamma_{\mathcal{Q}}) = (\mu_{\mathcal{P}} * \mu_{\mathcal{Q}}, \upsilon_{\mathcal{P}} * \upsilon_{\mathcal{Q}},\ \gamma_{\mathcal{P}} * \gamma_{\mathcal{P}},) \tag{6}$$

is used to multiply two fuzzy numbers.

Step 4: Fuzzy weights for all criteria are calculated using the formula

$$\varpi_i = \acute{r}_i \otimes (\acute{r}_1 \otimes \acute{r}_1 \otimes \dots \otimes \acute{r}_1)^{-1} \tag{7}$$

where $\acute{r}_i$ is the vector summation of each $\acute{r}ij^{\sim}$.

Step 5: We need to de-fuzzified these fuzzy weights to get weights in crisp value for all criteria using centre of area formula as

$$\varpi_i = \frac{\mu_{\mathcal{P}} + \upsilon_{\mathcal{P}} + \gamma_{\mathcal{P}}}{3} \tag{8}$$

Step 6: Normalize these weights to get the weight total as 1. These weights can be further used for ranking of alternatives.

**Table 1:** *Saaty's Scale Explanation*

| *Linguistic Term* | *Importance* | *Explanation* |
|---|---|---|
| Strongly low important (SLI) | 0.142857 | Values for inverse comparison |
| Very low important (VLI) | 0.2 | Values for inverse comparison |
| Low important (LI) | 0.333333 | Values for inverse comparison |
| Below average important (BAI) | 0.5 | Values for inverse comparison |
| Above average important (AAI) | 2 | Moderate advantage of the one element compared to the other |
| High important (HI) | 3 | High favouring of one element compared to the other |
| Very high important (VHI) | 5 | One element is given very high importance and has domination in practice compared to the other element |
| Strongly high important (SHI) | 7 | One element is favoured in comparison with the other based on strongly proved evidence and facts |
| Exactly equal (EE) | 1 | Both elements have equal contribution in the objective |

### 2.1.3. Fuzzy AHP using Triangular Fuzzy Numbers (TFN)

Basic AHP has been enhanced by utilising fuzzy logic since it does not include vagueness for subjective judgments. Through the linguistic variables, which are represented as triangular numbers in F-AHP, pairwise comparisons of both criteria and alternatives were carried out [33]. Van Laarhoven and Pedrycz carried out one of the earliest implementations of fuzzy AHP [34]. For pairwise comparisons, they defined the triangle membership functions. Following that, Buckley contributed to the discussion by identifying the fuzziness of comparison ratios with triangle membership functions [32]. A novel technique for using triangular numbers in pair-wise comparisons was also presented by Chang [35].

A fuzzy number $\mathcal{F} = (l, m, u)$ is defined as TFN [36] if its membership function

$$\mu_A(z) = \begin{cases} \frac{z-l}{m-l}, & l \le z \le m \\ \frac{u-z}{u-m}, & m \le z \le u \\ 0, & otherwise \end{cases} \tag{9}$$

Graphically, TFN has been presented in Figure 1.

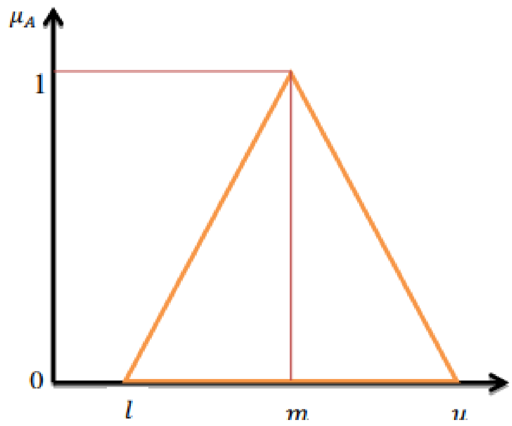


**Figure 1:** *TFN*

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

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

The membership function $\delta_{\mathcal{M}}$ of the fuzzy number $\mathcal{M}$ can also be expressed [37] as

$$\mu_A(z) = \begin{cases} \mu_{\mathrm{A}}^L(z), & a \le z \le b \\ 1, & b \le z \le c \\ \mu_{\mathrm{A}}^U(z), & c \le z \le d \\ 0, & otherwise \end{cases} \tag{10}$$

where $\mu_{\mathrm{A}}^L(z)$ and $\mu_{\mathrm{A}}^U(z)$ are the lower and upper membership functions of fuzzy number A, respectively and $a \le b \le c \le d$.

The procedure of fuzzy AHP suggested by Cheng's extent analysis is as follows:

(1) Draw the hierarchal diagram.

(2) Describe pair-wise comparisons in the form of fuzzy numbers.

(3) Gather data as a fuzzy pairwise comparison matrix based on DMs judgement and expressed as

$$\begin{pmatrix} (1,1,1) & (l_{12}, m_{12}, u_{12}) & \dots & (l_{1n}, m_{1n}, u_{1n}) \\ (l_{21}, m_{21}, u_{21}) & (l_{22}, m_{22}, u_{22}) & \dots & (l_{2n}, m_{2n}, u_{2n}) \\ \dots . & \dots . & & \dots . \\ (l_{n1}, m_{n1}, u_{n1}) & (l_{n2}, m_{n2}, u_{n2}) & & (1,1,1) \end{pmatrix}$$

(4) Calculate $\mathring{\check{S}}_i$ for every row of pair-wise comparison matrix using the expression

$$\mathring{\check{S}}_i = \sum_{j=1}^{m} \acute{M}_{gi}^j \otimes \left[\sum_{i=1}^{n} \sum_{j=1}^{m} \acute{M}_{gi}^j\right]^{-1} \tag{11}$$

where $\acute{M}_{gi}^j$ is TFN of pairwise comparison patterns. The values of $\sum_{j=1}^{m} \acute{M}_{gi}^j$ , $\sum_{i=1}^{n} \sum_{j=1}^{m} \acute{M}_{gi}^j$ and $\left[\sum_{i=1}^{n} \sum_{j=1}^{m} \acute{M}_{gi}^j\right]^{-1}$ are obtained by the expressions

$$\sum_{j=1}^{m} \acute{M}_{gi}^j = \left(\sum_{j=1}^{m} l_j,\ \sum_{j=1}^{m} m_j,\ \sum_{j=1}^{m} u_j\right) \tag{12}$$

$$\sum_{i=1}^{n} \sum_{j=1}^{m} \acute{M}_{gi}^j = \left(\sum_{i=1}^{n} l_i,\ \sum_{i=1}^{n} m_i,\ \sum_{i=1}^{n} u_i\right) \tag{13}$$

$$\left[\sum_{i=1}^{n} \sum_{j=1}^{m} \acute{M}_{gi}^j\right]^{-1} = \left(\frac{1}{\sum_{i=1}^{n} u_i}, \frac{1}{\sum_{i=1}^{n} m_i}, \frac{1}{\sum_{i=1}^{n} l_i}\right) \tag{14}$$

(5) Compute the magnitude of $\mathring{\check{S}}_i$ *w.r.t.* each other. If $\mathbb{M}_1 = (l_1, m_1, u_1)$ and $\mathbb{M}_2 = (l_2, m_2, u_2)$ are two TFNs, then, the magnitude of $\mathbb{M}_1$ *w.r.t.* $\mathbb{M}_2$ is defined as follows

$$\mathcal{V}(\mathbb{M}_2 \ge \mathbb{M}_1) = \begin{cases} 1, & if\ m_2 \ge m_1 \\ 0, & if\ l_1 \ge u_2 \\ \frac{l_1 - u_2}{(m_2 - u_2) - (m_1 - l_1)}, & otherwise \end{cases} \tag{15}$$

On the other hand, the magnitude of a TFN from k as another TFN is obtained by the following expression

$$\mathcal{V}(\acute{M} \ge \acute{M}_1, \acute{M}_2, \dots, \acute{M}_k) = \mathcal{V}[(\acute{M} \ge \acute{M}_1)\ and\ (\acute{M} \ge \acute{M}_2)\ and \dots (\acute{M} \ge \acute{M}_k)]$$
$$= Min\ \mathcal{V}(\acute{M} \ge \acute{M}_i),\ i = 1,2, \dots k \tag{16}$$

(6) Calculate the criteria and alternatives weight in pairwise comparison format using the expression

$$đ^{\dagger}(A_i) = Min\ \mathcal{V}(\mathring{\check{S}}_i \ge \mathring{\check{S}}_k),\ \ k = 1,2, \dots, n\ and\ k \ne i \tag{17}$$

where $A_i$ are n elements. The weight vector is now given by

$$\omega = [đ^{\dagger}(A_1), đ^{\dagger}(A_2), \dots, đ^{\dagger}(A_n)]^T \tag{18}$$

(7) Compute the final weight vectors after normalization as

$$\theta = [đ(A_1), đ(A_2), \dots, đ(A_n)]^T \tag{19}$$

where $\theta$ is a non-fuzzy number.

### 2.1.4. Pythagorean Fuzzy Number (PFN) Method

We have evaluated the scale suggested by Gul [20] to determine the significance weight at every level of the hierarchical structure for the interval valued PFN. Let us assume that there are $i$ alternatives.

Step 1: The compromised pairwise comparison matrix $A = (a_{ij})_{n\times n}$ is structured based on linguistic evaluations of experts using the scale proposed [38] in table 2.

Step 2: The difference matrices $Đ = [đ_{ij}]_{n\times n}$ between the lower and upper values of the membership and non-membership functions are calculated using (20) and (21):

$$đ_L = \mu_L^2 - \nu_U^2 \quad (20)$$

$$đ_U = \mu_U^2 - \nu_L^2 \quad (21)$$

**Table 2**: *Weighing scale for PFN*

| *Linguistic Term* | *Lower value of membership degree ($\mu_L$)* | *Upper value of membership degree ($\mu_U$)* | *Lower value of non-membership degree ($\nu_L$)* | *Upper value of non-membership degree ($\nu_U$)* |
|---|---|---|---|---|
| Strongly low important (SLI) | 0 | 0 | 0.9 | 1 |
| Very low important (VLI) | 0.1 | 0.2 | 0.8 | 0.9 |
| Low important (LI) | 0.2 | 0.35 | 0.65 | 0.8 |
| Below average important (BAI) | 0.35 | 0.45 | 0.55 | 0.65 |
| Above average important (AAI) | 0.55 | 0.65 | 0.35 | 0.45 |
| High important (HI) | 0.65 | 0.8 | 0.2 | 0.35 |
| Very high important (VHI) | 0.8 | 0.9 | 0.1 | 0.2 |
| Strongly high important (SHI) | 0.9 | 1 | 0 | 0 |
| Exactly equal (EE) | 0.1965 | 0.1965 | 0.1965 | 0.1965 |

Step 3: Calculate relative multiplicative matrix $\mathfrak{M} = [\mathfrak{m}_{ij}]_{n\times n}$ with the help of (22) and (23):

$$\mathfrak{m}_L = \sqrt{1000^{đ_L}} \quad (22)$$

$$\mathfrak{m}_U = \sqrt{1000^{đ_U}} \quad (23)$$

Step 4: The determinacy value $\delta = [\delta_{ij}]_{n\times n}$ is calculated with the help of (24):

$$\delta_{ij} = 1 - (\mu_L^2 - \mu_L^2) - (\nu_U^2 - \nu_L^2) \quad (24)$$

Step 4: Compute matrix of weights $\omega = [\omega_{ij}]_{n\times n}$ is calculated by multiplying the relative multiplicative matrix with the determinacy value as

$$\omega_{ij} = \left(\frac{\mathfrak{m}_L + \mathfrak{m}_U}{2}\right)\delta_{ij} \quad (25)$$

Step 5: Normalize weights $\theta_i$ with the help of (26) as

$$\theta_i = \frac{\sum_{j=1}^{n} \omega_{ij}}{\sum_{i=1}^{n}\sum_{j=1}^{n} \omega_{ij}} \quad (26)$$

These weights can be further used for ranking of alternatives.

## 3. Proposed Methodology and Algorithm

The assessment of the criteria weights is one of the most crucial phases of multicriteria evaluation. As criteria weights plays a significant role in MCDM, it is crucial to pay close attention to the factors associated with the criteria weights. Most of the current evaluation techniques for determining the weights of the criteria are based on the expert's subjective opinions. The selection of the criteria weights has a significant impact on the accuracy of the findings produced by multicriteria evaluation methods. In this segment, a technique based on the approach proposed by Keshavarz-Ghorabaee et al. is used to establish the weights of the criteria in a multi-criteria decision-making problem for optimal allocation of software reliability [39]. This technique is part of the objective weighting techniques used to determine criteria weights. The criteria weights in this method are determined by the exclusion effect of each criterion on the implementation of alternatives. In this analysis, the performances of the alternatives are calculated using a fundamental use of logarithmic measure with equal weights. We utilize the absolute deviation measure to define the effects of eliminating every condition. This metric indicates the variance between the performance of the alternative as a whole and its performance when a criterion is removed. The procedures for determining objective weights are as follows:

- The compromised pairwise comparison matrix $\mathcal{X} = (x_{ij})_{m\times n}$ is built using expert linguistic evaluations.
- Define fuzzy numbers to be used for pair-wise comparisons.
- Normalize the decision matrix as

$$\mathcal{N}_{ij} = \frac{x_{ij}}{\sum_{j=1}^{n} x_{ij}}; \tag{27}$$

$i = 1,2, \dots, m \; and \; j = 1,2, \dots, n$

- Calculate the overall performance as follows:

$$\mathcal{P}_i = ln\left[1 + \left\{\frac{1}{n}\sum_{i=1}^{n}\left|ln\left(\mathcal{N}_{ij}\right)\right|\right\}\right] \tag{28}$$

- Using the formula, compute the performance of the alternatives by removing the impact of each criterion.

$$\mathcal{P}_{ij}^{r} = ln\left[1 + \left\{\frac{1}{n}\sum_{k,k\neq i}^{n}\left|ln\left(\mathcal{N}_{ij}\right)\right|\right\}\right] \tag{29}$$

- Evaluate the sum of absolute deviations between the overall performance and performance of the alternatives by eliminating impact of each criterion as

$$\omega_i = \sum_{j=1}^{n}\left|\mathcal{P}_{ij}^{r} - \mathcal{P}_i\right| \tag{30}$$

- Determine weights $\theta_i$ with the help of (31) as

$$\theta_i = \frac{\omega_i}{\sum_{i=1}^{n}\omega_i} \tag{31}$$

These weights can be further used for ranking of alternatives.

## 4. Allocation of System Reliability

The user is the ultimate arbiter of a system's performance and dependability. Before designing any strategy, it is essential to base it on users' opinions and perceptions of the dependability of distinct functions. The views of the user, the software manager and programmer may disagree. To accomplish the goal, it is necessary to incorporate all perspectives on how to assign reliability values to different software modules, programmes, and functions. First, we identify our problem and establish the system's reliability objective. This objective is based on what the user anticipates from the software. We consider the system's target reliability as 90% for the sake of our research. Based on this value, we allocate reliability to these modules, functions, and programmes of the system.

A hierarchy structure (figure 2) is formed based on the problem's objective. There are different tiers of modules, programmes, and functions in this structure. The user's decision-making judgement is converted into a fuzzy numerical value. Following user feedback on function, software technologists express their favourites for the programmes, and finally, computer programmer assign the inclinations to the autonomous modules. The total system reliability is shown at the top of this hierarchical structure. The user's perspective on the functions is the focus of the hierarchy's second level. Users give their opinions based on software's ability to produce the desired results. We have taken up four functions and denoted as $F_1, F_2, F_3, and\ F_4$. The viewpoint of a software engineer is shown at the third stage of the hierarchical structure. The programmes built for the user-preferred functions are shown at this level. Each function is allowed to have a different number of programmes. We assume programs at this level as $P_1, P_2, and\ P_3$. The perspective of the programmer on the software system's modules is included in the hierarchy's bottom level. The system consists of 4 modules namely $M_1, M_2, M_3, and\ M_4$.

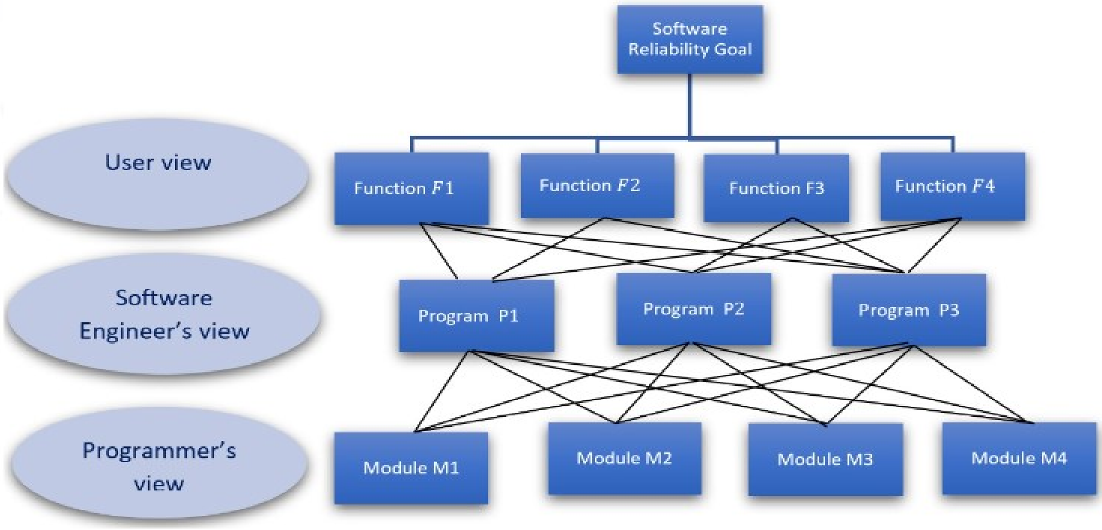


**Figure 2:** *The software hierarchy of the proposed system for reliability allocation*

We allocate the reliabilities after determining the weights for the alternatives based on the processes. We employ the following mathematical formula to determine the reliability to the priority weights:

$$\mathfrak{R}\mathcal{T}_i = \mathfrak{R}\mathcal{T}^{\theta_i}$$

where $\mathfrak{R}\mathcal{T}_i$: Reliability of $the\ i^{th}$ element, $\mathfrak{R}\mathcal{T}$: Objective reliability of the system and $\theta_i$: weights acquired.

Each element in the hierarchy is interconnected with multiple elements at the top levels. So, we choose the one that has the highest level of reliability. For example, if a program $P_1$ relates to four modules $M_1, M_2, M_3, and\ M_4$ then reliability allocated will be

$\mathfrak{R}\mathcal{T}_{P_1} = Max\ \{\mathfrak{R}\mathcal{T}_{M_1}, \mathfrak{R}\mathcal{T}_{M_2}, \mathfrak{R}\mathcal{T}_{M_3}, \mathfrak{R}\mathcal{T}_{M_4}\}$.

where $\mathfrak{R}\mathcal{T}_{P_1}$ represents reliability allocated to program $P_1$ and $\mathfrak{R}\mathcal{T}_{M_1}, \mathfrak{R}\mathcal{T}_{M_2}, \mathfrak{R}\mathcal{T}_{M_3}, and\ \mathfrak{R}\mathcal{T}_{M_4}$ are the reliabilities of modules $M_1, M_2, M_3, and\ M_4$ associated with $P_1$.

## 4.1. Application of the proposed framework

In this section, a simplified example has been used to illustrate how the suggested solution can be used to the reliable allocation problem. Figure 2 depicts the hierarchical structure of reliability of software.

It is assumed that the goal of our overall system reliability of the system to be 0.90. Assume a software with four functions has been created $F_1, F_2, F_3, and\ F_4$. The fuzzy values were assigned by DMs while performing pairwise comparing. As shown in table 1, linguistic statements were used for pairwise comparisons while collecting expert opinions using a scale of relative importance. The pairwise comparison matrix is attained as

**Table 3:** *Comparison matric of functions with linguistic statement using expert opinion*

| Software | $F_1$ | $F_2$ | $F_3$ | $F_4$ |
|---|---|---|---|---|
| $F_1$ | EE | SLI | VLI | LI |
| $F_2$ | SHI | EE | AAI | HI |
| $F_3$ | VHI | BAI | EE | HI |
| $F_4$ | HI | LI | LI | EE |

The relative importance using Saaty scale is shown in table 4 as

**Table 4:** *Comparison matric with relative importance using expert opinion*

| Software | $F_1$ | $F_2$ | $F_3$ | $F_4$ |
|---|---|---|---|---|
| $F_1$ | 1 | 0.1429 | 0.5 | 1 |
| $F_2$ | 7 | 1 | 0.333333 | 7 |
| $F_3$ | 2 | 3 | 1 | 2 |
| $F_4$ | 1 | 0.1429 | 0.5 | 1 |

The priority weights and allocation of reliabilities to these functions using proposed methodology is demonstrated in table 5 as

**Table 5:** *Reliability allocation for functions*

| Functions | Weights ($\theta_i$) | Reliability provision ($\Re\mathcal{T}_i = \Re\mathcal{T}^{\theta_i}$) |
|---|---|---|
| $F_1$ | 0.017643 | $(0.90)^{0.017643} = \mathbf{0.998143}$ |
| $F_2$ | 0.401141 | $(0.90)^{0.401141} = 0.958616$ |
| $F_3$ | 0.315821 | $(0.90)^{0.315821} = 0.967272$ |
| $F_4$ | 0.265395 | $(0.90)^{0.265395} = 0.972425$ |

The weights and reliabilities determined for all offered functions in relation to the target reliability are shown in the table 5. It has been observed from the above table that the maximum reliability is assigned to the function $F_1. i.e.,$ 0.998143 followed by $F_4$, $F_3$, and $F_2$ in that order. Furthermore, we assign reliability to the programmes associated to each individual function using the target reliability of the functions as our benchmark. To meet the needs of each function, relative weights of the programmes are computed for reliability criteria. Programs $P_1$ and $P_2$ are required for function $F_1$, $P_1$ and $P_3$ are required for function $F_2$, $P_2$ and $P_3$ are required for function $F_3$, and $P_1$, $P_2$, $P_3$ are required for function $F_4$,as shown in table 6.

After converting linguistic statements to relative importance, we find weights and allocate reliability to these programs using our methodology as described in table 7.

The table 7 shows that programme $P_1$ has been given the highest reliability rating, which is 0.999606, followed by $P_2$ and $P_3$ in that order. When a single programme is linked to multiple functions, we choose the highest level of reliability possible. In a similar way, we assign reliability to four proposed modules. To determine the weights of these modules, evaluation patterns as studied are developed. Here, all these programs relate to these four modules such as $M_1, M_2, M_3, and\ M_4$ and assessment pattern of these modules *w.r.t.* each program in linguistic form is depicted in table 8.

**Table 6:** *Comparison matric of programs with linguistic statement using expert opinion*

(a)

| $F_1$ | $P_1$ | $P_2$ |
|---|---|---|
| $P_1$ | EE | SLI |
| $P_2$ | SHI | EE |

(b)

| $F_2$ | $P_1$ | $P_3$ |
|---|---|---|
| $P_1$ | EE | BAI |
| $P_3$ | AAI | EE |

(c)

| $F_3$ | $P_2$ | $P_3$ |
|---|---|---|
| $P_2$ | EE | BAI |
| $P_3$ | AAI | EE |

(d)

| $F_4$ | $P_1$ | $P_2$ | $P_3$ |
|---|---|---|---|
| $P_1$ | EE | SLI | BAI |
| $P_2$ | SHI | EE | LI |
| $P_3$ | AAI | HI | EE |

**Table 7**: *Reliability allocation for programs*

| *Programs* | *Weights* ($\theta_i$) | *Reliability allocation* ($\Re\mathcal{T}_i = \Re\mathcal{T}^{\theta_i}$) |
|---|---|---|
| $P_1$ | 0.00374, 0.107793,0.072205 | $Max\,\{(0.90)^{0.00374}, (0.90)^{0.99626}, (0.90)^{0.072205}\} = \mathbf{0.999606}$ |
| $P_2$ | 0.99626, 0.035899, 0.356939 | $Max\,\{(0.90)^{0.99626}, (0.90)^{0.035899}, (0.90)^{0.356939}\} = 0.996225$ |
| $P_3$ | 0.892207, 0.964101, 0.570856 | $Max\,\{(0.90)^{0.892207}, (0.90)^{0.964101}, (0.90)^{0.570856}\} = 0.941627$ |

**Table 8**: *Comparison matric of modules with linguistic statement using expert opinion*

(a)

| $P_1$ | $M_1$ | $M_2$ | $M_3$ | $M_4$ |
|---|---|---|---|---|
| $M_1$ | EE | BAI | SLI | VLI |
| $M_2$ | AAI | EE | BAI | SHI |
| $M_3$ | SHI | AAI | EE | HI |
| $M_4$ | VHI | SLI | LI | EE |

(b)

| $P_2$ | $M_1$ | $M_2$ | $M_3$ | $M_4$ |
|---|---|---|---|---|
| $M_1$ | EE | VLI | BAI | SLI |
| $M_2$ | VHI | EE | AAI | LI |
| $M_3$ | AAI | BAI | EE | SLI |
| $M_4$ | SHI | HI | SHI | EE |

(c)

| $P_3$ | $M_1$ | $M_2$ | $M_3$ | $M_4$ |
|---|---|---|---|---|
| $M_1$ | EE | VLI | SLI | SHI |
| $M_2$ | VHI | EE | AAI | BAI |
| $M_3$ | SHI | BAI | EE | LI |
| $M_4$ | SLI | AAI | HI | EE |

After converting linguistic statements to relative importance, we find weights and allocate reliability to these modules using our methodology as described in table 9.

**Table 9:** *Reliability allocation for modules*

| *Programs* | *Weights* ($\theta_i$) | *Reliability allocation* ($\Re\mathcal{T}_i = \Re\mathcal{T}^{\theta_i}$) |
|---|---|---|
| $M_1$ | 0.048414, 0.034373, 0.220843 | $Max\,\{(0.90)^{0.048414}, (0.90)^{0.034373}, (0.90)^{0.220843}\} = \mathbf{0.996385}$ |
| $M_2$ | 0.287318, 0.28954, 0.33712 | $Max\,\{(0.90)^{0.287318}, (0.90)^{0.28954}, (0.90)^{0.33712}\} = 0.970182$ |
| $M_3$ | 0.374067, 0.134255, 0.260876 | $Max\,\{(0.90)^{0.374067}, (0.90)^{0.134255}, (0.90)^{0.260876}\} = 0.985954$ |
| $M_4$ | 0.290201, 0.541832, 0.181161 | $Max\,\{(0.90)^{0.290201}, (0.90)^{0.541832}, (0.90)^{0.181161}\} = 0.981094$ |

From the above table, it has been observed that the module $M_1$ has been given the highest reliability rating, which is 0.996385, followed by $M_3$, $M_4$ and $M_2$ in that order. Before developing the real system, all software components are given reliability targets using this allocation technique. These objectives must consider the normal user expectations as well as the specifications for the software's structure and reliability to be practical and significant. For software reliability allocation, this strategy enhances communication between users, software managers, and programmers.

## 5. Comparative Analysis

In this section, comparison has been carried out by finding criteria weights using existing methods as discussed in section 2. Weights calculation and reliability allocation for functions, programs and modules has been demonstrated in table 10-13 as shown below:

**Table 10**: *Weight calculation and reliability allocation using entropy method*

| *Functions* | *Weights* ($\theta_i$) | *Max. Reliability allocation* ($\Re_i = \Re^{\theta_i}$) |
|---|---|---|
| $F_1$ | 0.207905 | **0.978333** |
| $F_2$ | 0.299162 | 0.968972 |
| $F_3$ | 0.217526 | 0.977342 |
| $F_4$ | 0.275407 | 0.97140 |
| *Programs* | *Weights* ($\theta_i$) | *Max. Reliability allocation* ($\Re_i = \Re^{\theta_i}$) |
| $P_1$ | 0.49999, 0.5000, 0.291107 | 0.969794 |
| $P_2$ | 0.50000, 0.5000, 0.524813 | 0.948683 |
| $P_3$ | 0.50000, 0.5000, 0.184080 | **0.980792** |
| *Modules* | *Weights* ($\theta_i$) | *Max. Reliability allocation* ($\Re_i = \Re^{\theta_i}$) |
| $M_1$ | 0.180953, 0.222048, 0.305404 | 0.981115 |
| $M_2$ | 0.254397, 0.343202, 0.308463 | 0.973553 |
| $M_3$ | 0.158464, 0.259382, 0.254434 | 0.983443 |
| $M_4$ | 0.406186, 0.175368, 0.131699 | **0.98622** |

**Table 11:** *Weight calculation and reliability allocation using AHP method*

| *Functions* | *Weights* ($\theta_i$) | *Max. Reliability allocation* ($\Re_i = \Re^{\theta_i}$) |
|---|---|---|
| $F_1$ | 0.059249 | **0.993777** |
| $F_2$ | 0.482803 | 0.950404 |
| $F_3$ | 0.313851 | 0.967473 |
| $F_4$ | 0.144097 | 0.984933 |
| *Programs* | *Weights* ($\theta_i$) | *Max. Reliability allocation* ($\Re_i = \Re^{\theta_i}$) |
| $P_1$ | 0.125016, 0.3333, 0.075064 | **0.992122** |
| $P_2$ | 0.874984, 0.2500, 0.591723 | 0.974004 |
| $P_3$ | 0.66660, 0.7500, 0.3332130 | 0.965502 |
| *Modules* | *Weights* ($\theta_i$) | *Max. Reliability allocation* ($\Re_i = \Re^{\theta_i}$) |
| $M_1$ | 0.066274, 0.059672, 0.16054 | **0.993733** |
| $M_2$ | 0.31181, 0.233218, 0.358978 | 0.975727 |
| $M_3$ | 0.488009, 0.106113, 0.249495 | 0.988882 |
| $M_4$ | 0.133907, 0.60099, 0.230987 | 0.985991 |

**Table 12:** *Weight calculation and reliability allocation using fuzzy AHP (TFN) method*

| *Functions* | *Weights* ($\theta_i$) | *Max. Reliability allocation* ($\Re_i = \Re^{\theta_i}$) |
|---|---|---|
| $F_1$ | 0.056585 | **0.994056** |
| $F_2$ | 0.376814 | 0.961077 |
| $F_3$ | 0.348631 | 0.963934 |
| $F_4$ | 0.21797 | 0.977296 |
| *Programs* | *Weights* ($\theta_i$) | *Max. Reliability allocation* ($\Re_i = \Re^{\theta_i}$) |
| $P_1$ | NA, 0.50, 0.212266 | **0.977884** |
| $P_2$ | NA, 0.3163, 0.433994 | 0.9672 |
| $P_3$ | 0.50, 0.6837, 0.35374 | 0.963416 |
| *Modules* | *Weights* ($\theta_i$) | *Max. Reliability allocation* ($\Re_i = \Re^{\theta_i}$) |
| $M_1$ | 0.130874, 0.072106, 0.233092 | **0.992432** |
| $M_2$ | 0.317613, 0.265214, 0.277064 | 0.972444 |
| $M_3$ | 0.343306, 0.156194, 0.285197 | 0.983678 |
| $M_4$ | 0.208207, 0.506487, 0.204647 | 0.978669 |

**Table 13:** *Weight calculation and reliability allocation using PFN method*

| *Functions* | *Weights* ($\theta_i$) | *Max. Reliability allocation* ($\Re_i = \Re^{\theta_i}$) |
|---|---|---|
| $F_1$ | 0.020534 | **0.997839** |
| $F_2$ | 0.542303 | 0.944464 |
| $F_3$ | 0.31764 | 0.967087 |
| $F_4$ | 0.119524 | 0.987486 |
| *Programs* | *Weights* ($\theta_i$) | *Max. Reliability allocation* ($\Re_i = \Re^{\theta_i}$) |
| $P_1$ | 0.03444, 0.316381, 0.040192 | **0.995774** |
| $P_2$ | 0.965556, 0.14101, 0.709697 | 0.985253 |
| $P_3$ | 0.683619, 0.85899, 0.250111 | 0.973992 |
| *Modules* | *Weights* ($\theta_i$) | *Max. Reliability allocation* ($\Re_i = \Re^{\theta_i}$) |
| $M_1$ | 0.020222, 0.020222, 0.333048 | **0.997872** |
| $M_2$ | 0.366725, 0.202953, 0.206594 | 0.978844 |
| $M_3$ | 0.437906, 0.047376, 0.34180 | 0.995021 |
| $M_4$ | 0.175147, 0.729448, 0.118557 | 0.987586 |

Based on the comparison shown in table 10 to table 13 several finding can be made: first, function $F_1$ is assigned with maximum reliability by all the methods considered such as entropy method, AHP method, fuzzy AHP, PFN and our proposed technique. Secondly, maximum reliability is assigned to program $P_1$ by all except the entropy method and thirdly, for modules $M_1$ is chosen for the optimal allocation of software reliability. The above findings suggest that the suggested methodology provides software designers a productive and strategic method for producing highly robust software. The achievement of the best system reliability goal requires having a suitable reliability allocation method because development of a software is a significant cost factor of computer system. By incorporating users' opinions with those of software engineers and programmers, it also emphasises how crucial it is to comprehend users' roles in the software industry. As a result, this study improves communication between users, software engineers, and programmers. Also, employing this study DMs have more liberty and convenience in conveying their thoughts.

## 6. Conclusions and Future Scope

In MCDM, ascertaining criteria weights is a key issue. Weights are designed to convey the comparative importance of certain criteria. Real-world applications always involve varied degrees of criterion contribution to the outcomes being considered. Ignoring the issue will lead to wrong decisions. Pairwise comparisons are frequently employed as intermediate decision support when the DM finds it difficult to order the options as a whole and immediately regarding a criterion. Numerous techniques have been presented, and their efficacy has been compared, to estimate the preference values from the pairwise comparison matrix. It is essential to include users' perception while developing and designing software because they play a significant influence in the software market. This paper provides selective pioneered methods of computing priority weights and discusses the allocation of reliability of a software problem that arises during the designing and development phases of a programme. We compare the function based on the user's perception, compute the weights for each function, and assign reliability to them using a pairwise comparison matrix. We create a comparison matrix for modules considering the opinions of the programmers and a comparison matrix for programmes based on the decisions made by software engineers after assigning reliability to the functions. Then, based on the developed processes in the technique, reliability to programmes and modules is assigned using suggested approach. The study compares priority weights determined by entropy method, AHP, fuzzy AHP and Pythagorean fuzzy approach with the weights computed from proposed methodology for validation. We may assert that incorporating fresh MCDM techniques based on innovative viewpoints could guarantee the reliability of outcomes. Decision-makers can get weights that are more trustworthy by incorporating weighting procedures. The success of the software system reliability objective is assured by the reliability allocation model proposed in this work. For reliability allocation, this strategy enhances communication between consumers, software engineers, and programmers. The study broadens developers' comprehension of the impact of users' opinions during the software development phase as well as engineers' and programmers' perspectives at various levels of the hierarchy. However, the study is only capable of considering at one decision maker's perspective at each level. Furthermore, because the study only considers a few methods for computing the weighing criteria for functions, programmes, and modules, future investigation will focus on different approaches of decision-making in the allocation of reliability to software and can incorporate more objective and subjective methods for determining priority weights. The suggested approach can enhance future study in a vague defined environment, such as fuzzy, Neutrosophic and Fermatean environments.

## References

[1] Brooks, W. D. & Motley, R. W. (1980). Analysis of discrete software reliability Models, *Technical Report RADC-TR-80-84*, Rome Air Development Center, New York.

[2] Angus, J. E., Schafer, R. E., & Sukert, A. (1980). Software reliability model validation, *In Proc. Annu. Reliability and Maintainability Symp*., San Francisco, CA, Jan., pp. 191-193.

[3] Aggarwal, A. G., Dhaka, V., & Nijhawan, N. (2017). Reliability analysis for multi-release open-source software systems with change point and exponentiated Weibull fault reduction factor. *Life Cycle Reliability and Safety Engineering*,*6*(1),3–14.https://doi.org/10.1007/s41872-017-0001-0

[4] Zahedi, F., & Ashrafi, N. (1991). Software reliability allocation based on structure, utility, price, and cost. *IEEE Transactions on Software Engineering*, *17*(4), 345–356. https://doi.org/10.1109/32.90434

[5] Sitorus, F., Cilliers, J. J., & Brito-Parada, P. R. (2019). Multi-criteria decision making for the choice problem in mining and mineral processing: Applications and trends. *Expert Systems with Applications*, *121*, 393–417. https://doi.org/10.1016/j.eswa.2018.12.001

[6] Sehra, S. K., Singh Brar, Y., & Kaur, N. (2012). Multi Criteria Decision Making Approach for Selecting Effort Estimation Model. *International Journal of Computer Applications*, *39*(1), 10–17. https://doi.org/10.5120/4783-6989

[7] Zha, S., Guo, Y., Huang, S., & Wang, S. (2020). A Hybrid MCDM Method Using Combination Weight for the Selection of Facility Layout in the Manufacturing System: A Case Study. *Mathematical Problems in Engineering*, *2020*, 1–16. https://doi.org/10.1155/2020/1320173

[8] Arya, V., & Kumar, S. (2020). A new picture fuzzy information measure based on shannon entropy with applications in opinion polls using extended VIKOR–TODIM approach. *Computational and Applied Mathematics*, *39*(3). https://doi.org/10.1007/s40314-020-01228-1

[9] Xu, Z., & Zhang, X. (2013). Hesitant fuzzy multi-attribute decision making based on TOPSIS with incomplete weight information. *Knowledge-Based Systems*, *52*, 53–64. https://doi.org/10.1016/j.knosys.2013.05.011

[10] Kobryń, A. (2017). DEMATEL as a weighting method in multi-criteria decision analysis. *Multiple Criteria Decision Making*, *12*, 153–167. https://doi.org/10.22367/mcdm.2017.12.11

[11] Ueasin, N. (2020). Decision-making on Public Transportation Services Based on the Socio-economic, Psychological, and Environmental Concern Factors. *The Open Transportation Journal*, *14*(1), 22–31. https://doi.org/10.2174/1874447802014010022

[12] Zagorskas, J., & Turskis, Z. (2020). Location Preferences of New Pedestrian Bridges Based on Multi-Criteria Decision-Making and GIS-Based Estimation. *The Baltic Journal of Road and Bridge*

*Engineering*, *15*(2), 158–181. https://doi.org/10.7250/bjrbe.2020-15.478

[13] Alakaş, H. M., Gür, Ş., Özcan, E., & Eren, T. (2020). Ranking of sustainability criteria for industrial symbiosis applications based on ANP. *Journal of Environmental Engineering and Landscape Management*, *28*(4), 192–201. https://doi.org/10.3846/jeelm.2020.13689

[14] Keshavarz-Ghorabaee, M., Govindan, K., Amiri, M., Zavadskas, E. K., & Antuchevičienė, J. (2019). An integrated type-2 fuzzy decision model based on WASPAS and SECA for evaluation of sustainable manufacturing strategies. *Journal of Environmental Engineering and Landscape Management*, *27*(4), 187–200. https://doi.org/10.3846/jeelm.2019.11367

[15] Pamučar, D., Stević, Ž., & Sremac, S. (2018). A New Model for Determining Weight Coefficients of Criteria in MCDM Models: Full Consistency Method (FUCOM). *Symmetry*, *10*(9), 393. https://doi.org/10.3390/sym10090393

[16] Alfares, H. K., & Duffuaa, S. O. (2016). Simulation-Based Evaluation of Criteria Rank-Weighting Methods in Multi-Criteria Decision-Making. *International Journal of Information Technology & Decision Making*, *15*(01), 43–61. https://doi.org/10.1142/s0219622015500315

[17] Németh, B., Molnár, A., Bozóki, S., Wijaya, K., Inotai, A., Campbell, J. D., & Kaló, Z. (2019). Comparison of weighting methods used in multicriteria decision analysis frameworks in healthcare with focus on low- and middle-income countries. *Journal of Comparative Effectiveness Research*, *8*(4), 195–204. https://doi.org/10.2217/cer-2018-0102

[18] Diakoulaki, D., Mavrotas, G., & Papayannakis, L. (1995). Determining objective weights in multiple criteria problems: The critic method. *Computers & Operations Research*, *22*(7), 763–770. https://doi.org/10.1016/0305-0548(94)00059-H

[19] Chatterjee, S., Singh, J. B., & Roy, A. (2013). A structure-based software reliability allocation using fuzzy analytic hierarchy process. *International Journal of Systems Science*, *46*(3), 513–525. https://doi.org/10.1080/00207721.2013.791001

[20] Gul, M. (2018). Application of Pythagorean fuzzy AHP and VIKOR methods in occupational health and safety risk assessment: the case of a gun and rifle barrel external surface oxidation and colouring unit. *International Journal of Occupational Safety and Ergonomics*, *26*(4), 705–718. https://doi.org/10.1080/10803548.2018.1492251

[21] Saaty, T. L. (1977). A scaling method for priorities in hierarchical structures. *Journal of Mathematical Psychology*, *15*(3), 234–281. https://doi.org/10.1016/0022-2496(77)90033-5

[22] Ginevičius, R. (2011). A new determining method for the criteria weights in multicriteria evaluation. *International Journal of Information Technology & Decision Making*, *10*(06), 1067–1095. https://doi.org/10.1142/s0219622011004713

[23] Odu, G. O. (2019). Weighting methods for multi-criteria decision-making technique. *Journal of Applied Sciences and Environmental Management*, *23*(8), 1449. https://doi.org/10.4314/jasem.v23i8.7

[24] Kasim, M. M. (2014). Determination of criteria weights in solving multi-criteria problems. *AIP Conference Proceedings*. https://doi.org/10.1063/1.4903555

[25] Gupta, A., Gupta, N., Garg, R., & Kumar, R. (2018). Evaluation, Selection and Ranking of Software Reliability Growth Models using Multi criteria Decision making Approach. *2018 4th International Conference on Computing Communication and Automation (ICCCA)*. https://doi.org/10.1109/ccaa.2018.8777644

[26] Neha, N. A., Verma, V., Tandon, A., & Aggarwal, A. G. (2021). Software reliability allocation incorporating Pythagorean fuzzy theory and AHP. *International Journal of Industrial and Systems Engineering*, *38*(1), 19. https://doi.org/10.1504/ijise.2021.115128

[27] Zoraghi, N., Amiri, M., Talebi, G., & Zowghi, M. (2013). A fuzzy MCDM model with objective and subjective weights for evaluating service quality in hotel industries. *Journal of Industrial Engineering International*, *9*(1). https://doi.org/10.1186/2251-712x-9-38

[28] Şahin, M. (2020). A comprehensive analysis of weighting and multicriteria methods in the

context of sustainable energy. *International Journal of Environmental Science and Technology*, *18*(6), 1591–1616. https://doi.org/10.1007/s13762-020-02922-7

[29] Liu H, & Kong F. (2005). A new MADM algorithm based on fuzzy subjective and objective integrated weights. *International Journal of Information and Systems Sciences*, 1: 420-427

[30] Shannon, C. E. (1948). A Mathematical Theory of Communication. *Bell System Technical Journal*, *27*(3), 379–423. https://doi.org/10.1002/j.1538-7305.1948.tb01338.x

[31] Saaty, T. L., & Vargas, L. G. (1982). *The Logic of Priorities*. Springer Netherlands. https://doi.org/10.1007/978-94-017-3383-0

[32] Buckley, J. J. (1985). Fuzzy hierarchical analysis. *Fuzzy Sets and Systems*, *17*(3), 233–247. https://doi.org/10.1016/0165-0114(85)90090-9

[33] Kilincci, O., & Onal, S. A. (2011). Fuzzy AHP approach for supplier selection in a washing machine company. *Expert Systems with Applications*, *38*(8), 9656–9664. https://doi.org/10.1016/j.eswa.2011.01.159

[34] van Laarhoven, P. J. M., & Pedrycz, W. (1983). A fuzzy extension of Saaty's priority theory. *Fuzzy Sets and Systems*, *11*(1-3), 229–241. https://doi.org/10.1016/s0165-0114(83)80082-7

[35] Chang, D.-Y. (1996). Applications of the extent analysis method on fuzzy AHP. *European Journal of Operational Research*, *95*(3), 649–655. https://doi.org/10.1016/0377-2217(95)00300-2

[36] Kaufmann, A., & Gupta, M. M. (1991). *Introduction to fuzzy arithmetic: theory and applications.* New York (N.Y.): Van Nostrand Reinhold.

[37] Chu, T.-C., & Lin, Y.-C. (2009). An interval arithmetic based fuzzy TOPSIS model. *Expert Systems with Applications*, *36*(8), 10870–10876. https://doi.org/10.1016/j.eswa.2009.01.083

[38] Ilbahar, E., Karaşan, A., Cebi, S., & Kahraman, C. (2018). A novel approach to risk assessment for occupational health and safety using Pythagorean fuzzy AHP & fuzzy inference system. *Safety Science*, *103*, 124–136. https://doi.org/10.1016/j.ssci.2017.10.025

[39] Keshavarz-Ghorabaee, M., Amiri, M., Zavadskas, E. K., Turskis, Z., & Antucheviciene, J. (2021). Determination of Objective Weights Using a New Method Based on the Removal Effects of Criteria (MEREC). *Symmetry*, *13*(4), 525. https://doi.org/10.3390/sym13040525