



## Article

# An Integrated Decision Approach with Probabilistic Linguistic Information for Test Case Prioritization

A. D. Shrivathsan <sup>1</sup>, R. Krishankumar <sup>1</sup>, Arunodaya Raj Mishra <sup>2</sup> , K. S. Ravichandran <sup>1</sup> , Samarjit Kar <sup>3,\*</sup> and V. Badrinath <sup>1</sup>

<sup>1</sup> School of Computing, SASTRA University, Thanjavur 613401, TN, India; shrivathsan.a.d@mca.sastra.edu (A.D.S.); krishankumar@sastra.ac.in (R.K.); raviks@sastra.edu (K.S.R.); deanbadri@sastra.edu (V.B.)

<sup>2</sup> Department of Mathematics, Government College, Jaitwara 485221, MP, India; arunodaya87@outlook.com

<sup>3</sup> Department of Mathematics, National Institute of Technology, Durgapur 713209, WB, India

\* Correspondence: samarjit.kar@maths.nitdgp.ac.in

Received: 7 September 2020; Accepted: 22 October 2020; Published: 23 October 2020



**Abstract:** This paper focuses on an exciting and essential problem in software companies. The software life cycle includes testing software, which is often time-consuming, and is a critical phase in the software development process. To reduce time spent on testing and to maintain software quality, the idea of a systematic selection of test cases is needed. Attracted by the claim, researchers presented test case prioritization (TCP) by applying the concepts of multi-criteria decision-making (MCDM). However, the literature on TCP suffers from the following issues: (i) difficulty in properly handling uncertainty; (ii) systematic evaluation of criteria by understanding the hesitation of experts; and (iii) rational prioritization of test cases by considering the nature of criteria. Motivated by these issues, an integrated approach is put forward that could circumvent the problem in this paper. The main aim of this research is to develop a decision model with integrated methods for TCP. The core importance of the proposed model is to (i) provide a systematic/methodical decision on TCP with a reduction in testing time and cost; (ii) help software personnel choose an apt test case from the suite for testing software; (iii) reduce human bias by mitigating intervention of personnel in the decision process. To this end, probabilistic linguistic information (PLI) is adopted as the preference structure that could flexibly handle uncertainty by associating occurrence probability to each linguistic term. Furthermore, an attitude-based entropy measure is presented for criteria weight calculation, and finally, the EDAS ranking method is extended to PLI for TCP. An empirical study of TCP in a software company is presented to certify the integrated approach's effectiveness. The strengths and weaknesses of the introduced approach are conferred by comparing it with the relevant methods.

**Keywords:** decision-making; EDAS approach; entropy measure; probabilistic linguistic information; test case prioritization

## 1. Introduction

Multi-criteria decision-making (MCDM) is an attractive concept that involves a set of options selected based on a set of criteria, either linguistically or numerically. Each criterion is associated with an importance value utilized by the ranking approach to form the ranking order [1]. Zadeh [2] introduced the philosophy of linguistic decision-making and discussed its merits and the flexibility offered to decision-makers (DMs) in preference information. Later, Herrera et al. [3] fine-tuned the notion and made it more applicable to MCDM. Rodriguez et al. [4] identified a crucial weakness of linguistic term sets (LTSs) and proposed hesitant fuzzy linguistic term sets (HFLTSSs) to resolve the same. As stated, the HFLTSS has the ability to accept more than one rating for a particular alternative—criterion

pair, which is lacking in the LTS. The core strength of HFLTSS is that they combine the ability of LTSs and hesitant fuzzy sets (HFSs), [5,6] which not only allows for preference elicitation linguistically, but also offers flexibility to handle hesitation better. Attracted by this flexibility, many researchers contribute MCDM methods under HFLTSS [7].

Even though HFLTSS tackle the crucial weakness of LTSs, the idea of associating occurrence probability to each term is missing, and as a result, key information is lost. To tackle this problem, Pang et al. [8] gave a preference structure known as probabilistic linguistic information (PLI), which associates occurrence probability to each term, which indicates the expectation of each term during preference elicitation. This feature of PLI attracted many researchers to propose new decision models under PLI, such as aggregation-based models [9,10], distance/similarity-based models [11], ranking-based models [12–14], and fundamental arithmetic operations-based models [15]. From the detailed survey prepared by Liao et al. [16], it is evident that (i) PLI is a strong structure for handling uncertainty, and (ii) TCP is not explored under the PLI context.

### 1.1. Literature Review on TCP

With the agile paradigm's implementation by many software enterprises, we discern increasing attention in continuous integration (CI) settings. Such settings permit more repetitive integration of software alterations, creating software progression that is more quick and cost-effective [17]. The outcomes are utilized to tackle issues and find faults, and speedy feedback is essential to diminish progress costs [18]. Within an integration cycle, regression testing (RT) is an action that takes a momentous volume of time. Many times, a test set comprises thousands of test cases whose execution grosses numerous hours or days [19].

To assist in the RT task, we discover in the literature various methods, which are generally categorized into three key types [20]: minimization, selection, and prioritization. Models with test case minimization (TCM) generally eliminate redundant test cases, minimizing the test set based on several attributes. Test case selection (TCS) chooses a subset of test cases, the vital ones to test the software. Test case prioritization (TCP) endeavors to re-order a test suite to recognize an "ideal" order of test cases that maximizes precise goals, namely early fault detection. TCP methods are renowned in the enterprises and are the subject of study.

Recent software structures constantly progress because of the fixing of detected bugs, accumulating original functionalities, and refactoring structure architecture. RT is used to confirm that the reformed source code does not familiarize new defects. It can become lavish to run a whole RT suite since its size naturally increases through software maintenance and evolution in an industrial case testified by Rothermel et al. [21]. For instance, the execution time for running the whole test suite could take some weeks. RT case prioritization (RTCP) has become one of the most operative structures to lessen the overheads in regression testing [22–25]. RTCP procedures reorder the execution sequence of RT cases, aiming to execute those test cases and increase the possibility of detecting faults as quickly as possible [26,27]. Tahvili and Bohlin [28] gave a novel decision model TCP under the context of an industrial perspective.

Traditional RTCP techniques [21,29,30] typically utilized a code coverage criterion (CCC) to lead the prioritization procedure. Naturally, a CCC specifies the percentage of selected code units enclosed by a test case. The expectancy is that test cases with greater code coverage degrees have a greater chance of detecting faults [31]. Furthermore, Li et al. [22] gave two search-based RTCP models to discover the search space to assess a sequence with a better fault detection rate. Jiang et al. [23] explored adaptive random models [32] to rank test cases by CCC. To bridge the gap between the two greedy schemes, Zhang et al. [29] suggested an integrated model with the fault detection probability for each test case. Pradhan et al. [33] introduced and conducted an overall empirical assessment of a rule mining and search-based dynamic prioritization methodology with three key components to detect faults earlier. Shrivathsan et al. [34] developed two fuzzy-based clustering methods for TCP by

similarity coefficient and dominance measure. Additionally, they adopted the weighted arithmetic sum product assessment (WASPAS) model for ranking with both inter- and intra-perspectives.

Next, Khatibsyarhini et al. [35] classified and criticized the current state and trend of TCP models, using a systematic literature review (SLR) to conclude the work. With the SLR structure, several applicable research questions (RQs) were framed based on the study's goal. Banias [36] proposed a dynamic programming model for handling TCP assessment problems using low memory consumption in pseudo-polynomial time complexity applicable in the assessment. Chi et al. [37] presented a relation-based TCP technique called additional greedy method call sequence (AGCS) based on method call sequences. The developed approach leverages dynamic relation-based coverage as a measurement to extend the original additional greedy coverage procedure in TCP techniques. Lima and Vergilio [38] presented the outcomes of a systematic mapping investigation on TCP in continuous integration (TCPCI) settings that reported the key features of TCPCI models and their assessment facets. The mapping was used as a plan that contains the definition of RQs, assessment attributes, search string, and evaluation of search engines. Huang et al. [39] proposed a coverage attribute, a code arrangements coverage that groups the conception of code coverage and combination coverage. Mahdiah et al. [40] proposed a model that enhances coverage-based TCP procedures by taking fault-proneness distribution over code units. Additionally, they presented the outcomes of a case study that showed that the approach expressively recovers the additional strategy, which is a broadly utilized coverage-based TCP model.

### 1.2. Research Challenges in TCP

Driven by the inference from the literature review conducted above, some research challenges are put forward.

- Uncertainty in preference elicitation is not properly handled in TCP.
- Preference information from different software personnel is not presented holistically for better TCP. A flexible structure to depict the views holistically is lacking in the state-of-the-art models.
- Weights of criteria that are conflicting and competing with each other are not calculated systematically by capturing DMs' hesitation. Besides, the attitude of DMs is also not taken into consideration during weight calculation.
- The nature of criteria is not considered during the prioritization of test cases, affecting the decision process. Besides, broad rank values are missing in the state-of-the-art models that could promote proper backup management.

### 1.3. Contributions of the Integrated Approach

Driven by the research challenges presented in Section 1.2, certain key contributions are put forward.

- PLI [8] is adopted as the preference structure that handles uncertainty better and provides a holistic view of the data from different software personnel. This concept resolves the first and second challenges.
- An attitude-based entropy measure is proposed with PLI for criteria weight calculation that would capture DMs' hesitation and consider the attitude of DMs during preference elicitation. This resolves the third challenge.
- Further, an evaluation based on distance from an average solution (EDAS) approach is extended to PLI for rational prioritization of test cases. The approach considers the nature of criteria during the ranking of test cases and produces broad rank values that promote effective backup management. This resolves the fourth challenge.
- Finally, the integrated approach is exemplified with a real case study of test case prioritization in a software project. The advantages and weaknesses of the introduced method are discussed by comparing it with diverse TCP models.

### 1.4. Outline of This Paper

The paper is constructed with the following outline. Section 2 provides the basic concepts that form the base for the research. Section 3 describes the core idea of the research, which begins with data collection and transformation, followed by criteria weight calculation and ranking of test cases. To clearly understand the usefulness of the proposed work, Section 4 presents a numerical example of test case prioritization in a software company. Section 5 focuses on comparative analysis that clarifies the strengths and weaknesses of the proposed work. Finally, Section 6 offers the concluding remarks and future directions for the research.

## 2. Preliminaries

This section is committed to describing certain elementary concepts related to the LTS and its generalized structures.

**Definition 1.** (Herrera and Herrera-Viedma, [3])  $T$  is an LTS of the form  $\{s_c | c = 0, 1, \dots, \beta\}$ . Here,  $\beta + 1$  is the cardinality of  $T$ ,  $s_0$  is the first element, and  $s_\beta$  is the last element of  $T$ . Certain features of  $T$  are

If  $c_1 > c_2$  then,  $s_{c_1} > s_{c_2}$ ;

$neg(s_{c_1}) = s_{c_2}$  with  $c_1 + c_2 = \beta$  is called the negation operation.

**Definition 2.** (Rodriguez et al. [4])  $T$  is defined as before. HFLTS is an ordered finite subset of  $T$  and it is given by

$$TH = \{xx, h_{TH}(xx) | x \in XX\} \quad (1)$$

where  $h_{TH}(xx) = h(xx)$  has terms from  $T$  and it is represented as  $h(xx) = \{s_c^k | c = 0, 1, \dots, \beta; k = 1, 2, \dots, \#h(xx)\}$ . Here,  $\#h(c)$  refers to total number of instances.

**Definition 3.** (Pang et al. [8])  $T$  is defined as before. Probabilistic linguistic term set (PLTS) is an ordered finite subset of  $T$  with associated probability for each term, and it is given by

$$TH(p) = \left\{ TH^k(p^k) | TH^k \in T, p^k \in [0, 1], k = 0, 1, \dots, \#th(p), \sum_k p^k \leq 1 \right\} \quad (2)$$

where  $p^k$  is the associated probability value for each term  $TH^k$  and  $\#th(p)$  refers to the total number of instances. For convenience,  $th(p) = th = \{s_c^k(p^k)\}$  is the PLI, and the collection of PLI yields the PLTS.

**Definition 4.** (Gou et al. [15]) Two PLI,  $th_1$  and  $th_2$ , are as stated above. Some operational laws are given by

$$th_1 \oplus th_2 = f^{-1}(f(th_1) + f(th_2)) \quad (3)$$

$$th_1 \otimes th_2 = f^{-1}(f(th_1) \times f(th_2)) \quad (4)$$

where  $f$  and  $f^{-1}$  are functions described in [15].

$$f : \{0, 1, \dots, \beta\} \rightarrow [0, 1]; f(s_c) = \frac{c}{\beta} = \eta$$

$$f^{-1} : [0, 1] \rightarrow \{0, 1, \dots, \beta\}; f^{-1}(\eta) = \eta \cdot \beta = s_c$$

## 3. Proposed Integrated Model for TCP

### 3.1. Data Transformation to PLI

This section focuses on the process of transforming Likert scale-based rating information into PLI without loss of generality. To achieve this, data from different people/personnel/experts are collected in a Likert-scale rating. These are linguistic ratings that are natural and easy from the human point of

view. Opinions on each test case based on a specific criterion are obtained from different personnel that are categorical in nature.

To form a holistic decision/preference matrix without generality loss, the occurrence probability of linguistic terms is determined. The instances are formed based on the descending order of probability values. For ease of understanding, let us consider an example. Six personnel (experts) rate a car with respect to its safety measures, and they adopt a five-point Likert scale rating, where E1 = good gives the values:  $E_1 = \text{good}$ ,  $E_2 = \text{fair}$ ,  $E_3 = \text{good}$ ,  $E_4 = \text{bad}$ ,  $E_5 = \text{fair}$ , and  $E_6 = \text{fair}$ . The occurrence probability of each linguistic term is calculated as  $\text{good} = 2/6 = 0.3333$ ,  $\text{fair} = 3/6 = 0.5$ , and  $\text{bad} = 1/6 = 0.1667$ . Since the committee has planned to use two instances for analysis, PLI is constructed with two instances, and it is given by  $\{\text{good}, (0.3333); \text{fair}(0.5)\}$ . Clearly, it follows Definition 3. Likewise, the entire preference matrix is constructed.

### 3.2. Attitude-Based Entropy Measure

This section focuses on a new approach for criteria weight calculation by properly capturing the hesitation/confusion during the elicitation of preferences. Further, DMs' attitude is also gathered from the top officials that are used in the formulation for the determination of weights. Generally, weights are determined either with completely unknown information or partially known information. The latter context requires additional information on each criterion, which is sometimes difficult to obtain, creating additional overhead. To resolve the issue, the former context is developed. The popular method under the former context is the analytical hierarchy process [41], which faces the problem of consistency maintenance and follows a pairwise comparison that complicates the weight calculation process.

In this study, a novel attitude-based entropy measure is suggested for rational computation of criteria weights to avoid such issues. The Shannon entropy measure is extended to PLI for rational weight calculation, and it is the measure of the expected information of content. Additionally, entropy is the measure of the degree of uncertainty in the form of a probability distribution. In general, MCDM concepts can effectively adopt entropy [42,43] as there is an intrinsic average information transfer between DMs, and variation in the preference information can be captured effectively. Driven by the effectiveness of entropy measures, in this section, a stepwise process for assessing the criteria weight by extending the Shannon entropy measure to PLI is provided as below.

**Step 1:** Generate a matrix of order  $de \times n$  with PLI that is called the criteria weight calculation matrix, where  $de$  and  $n$  are the number of DMs and criteria, respectively.

**Step 2:** Convert the PLI into a single value by applying Equation (5), and it forms a matrix of order  $de \times n$ .

$$v_{lj} = \sum_{k=1}^{\#th} s_c^{*k} \times p^{*k} \quad (5)$$

where  $c$  is the subscript of the linguistic part, and  $p$  is the probability value associated with each term. Here,  $s_c^{*k}$  is a weighted linguistic value, and  $p^{*k}$  is a weighted probability associated with linguistic value. It is calculated as  $s_c^{*k} = s_{\zeta_l c}^k$  and  $p^{*k} = 1 - (1 - p^k)^{\zeta_l}$ . Here  $\zeta_l$  is the attitude value associated with the  $l^{th}$  expert, and its value is in the unit interval with the sum of values equal to unity.

**Step 3:** Define the deviation value for each criterion by applying Equation (6) that forms a matrix of order  $de \times n$ .

$$D_{lj} = |v_{lj} - \bar{v}_j| \quad (6)$$

where  $\bar{v}_j$  is the mean value determined for the  $j^{th}$  criterion.

**Step 4:** Determine the information entropy by adopting Equation (7) that produces a vector of order  $1 \times n$ .

$$En_j = -\frac{1}{\ln(de)} \left( \sum_{l=1}^{de} \frac{D_{lj}}{D_j^{tot}} \ln \left( \frac{D_{lj}}{D_j^{tot}} \right) \right) \quad (7)$$

where  $D_j^{tot}$  is the total deviation value determined for the  $j^{th}$  criterion.

**Step 5:** Normalize these entropy values by using Equation (8) to obtain a vector of order  $1 \times n$ , which denotes the criteria weights.

$$wt_j = \frac{En_j}{\sum_j En_j} \quad (8)$$

where  $wt_j$  is the  $j^{th}$  attribute weight and  $\sum_j wt_j = 1$  with the weights in the unit interval.

**Note:** The entropy measure provided in Equation (7) is inspired from a popular measure called the Shannon entropy whose validity in PLI is verified by acquiring theoretical foundation from [44,45]. Readers are requested to refer these articles for clarity. As a novelty, in this section, we adopted the measure for calculating weights of each criterion by considering not only the deviation in the distribution, but also the attitude of each expert. Readers may refer to Appendix A for theoretical aspects of the entropy measure.

### 3.3. PLI-Based EDAS Method

This section focuses on the issue of prioritization of test cases. For this reason, in this study, a new extension is made to the EDAS method with PLI. Ghorabae et al. [46] presented the EDAS approach with the core idea of assessing alternatives based on average values and applied it for inventory classification. Karaşan and Kahraman [47] extended EDAS to the neutrosophic set and used it for ranking sustainable goals. Peng and Liu [48] came up with a neutrosophic soft EDAS approach with a new similarity measure for solving a software project's investment problem. Mishra et al. [49] also used intuitionistic fuzzy EDAS for healthcare waste management. Liang et al. [50] used the intuitionistic fuzzy EDAS approach to select energy-saving green building projects. Feng et al. [51] proposed an HFLTS-based EDAS approach with a weighted arithmetic operator for project evaluation in a company.

Inspired by the literature, it is observed that (i) EDAS is a powerful ranking method that uses a distance measure in the formulation; (ii) TCP by using the EDAS approach is not done to the best of our knowledge; and finally, (iii) EDAS is a simple and straightforward approach that could be effectively integrated with PLI for rational decision-making.

The step by step procedure for the PLI-based EDAS approach is given by

**Step 1:** Achieve the holistic decision matrix of order  $m \times n$  with PLI from Section 3.1, where  $m$  and  $n$  are the number of test cases and the fault identified by the test case, respectively.

**Step 2:** Obtain the weight vector of order  $1 \times n$  by utilizing the process in Section 3.2.

**Step 3:** Transform the matrix values as weighted single-value elements by using Equation (9) with the same order.

$$wsv_{ij} = \sum_{k=1}^{\#th} s_{wt_j c}^k \times \left(1 - (1 - p^k)^{wt_j}\right) \quad (9)$$

where  $c$  is the subscript of the term,  $p$  is the probability associated with the term.

**Step 4:** Determine the average of values for each criterion to form a vector of order  $1 \times n$ . Use this average and the values to analyze the positive and negative distance from average (PDA and NDA) by using Equations (10) and (11).

$$PDA_i = \sum_{j=1}^n d(wsv_{ij}, \overline{wsv_j}) \quad (10)$$

$$NDA_i = \sum_{j=1}^n d(\overline{wsv_j}, wsv_{ij}) \quad (11)$$

where  $\overline{wsv_j}$  is the average value for the  $j^{th}$  criterion.

Though Equations (10) and (11) look similar, the value in the matrix is normalized and then transformed to adhere to the nature of criteria rationally. For this purpose, benefit type criteria are complemented in Equation (11), and cost type criteria are complemented in Equation (10), respectively, before calculating the distance measure.  $d(a, b)$  is given by  $|a - b|$  where  $a$  is the normalized value, and  $b$  is the mean of the set of normalized values.



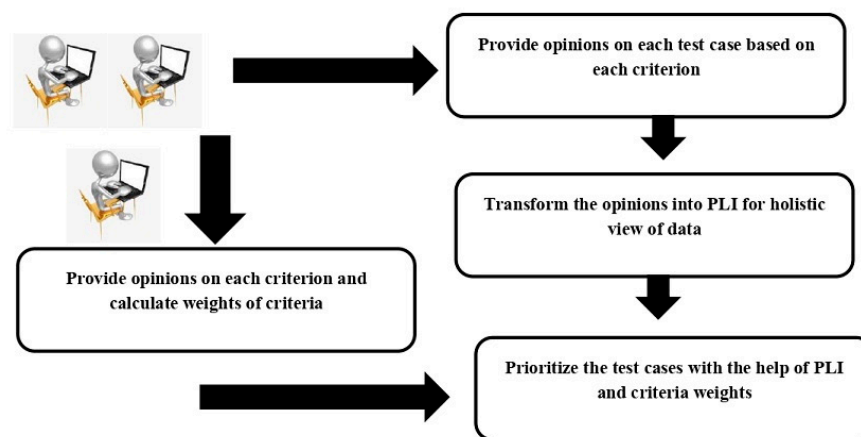
**Step 5:** Test cases are prioritized by taking a linear combination of the vector values from Equations (10) and (11), and it is given in Equation (12).

$$RTCP_i = \theta PDA_i + (1 - \theta) NDA_i \quad (12)$$

where  $\theta$  is a strategy value in the unit interval.

Arrange the values from  $RTCP_i$  in descending order to form the prioritization order of test cases.

Figure 1 depicts the proposed decision model for the rational prioritization of test cases. Initially, complex linguistic expressions are obtained as opinions from experts (software test personnel) on each test case over the criteria. These values are transformed into PLI by using the procedure proposed in Section 3.1. Experts also share their opinions on each criterion used as input for criteria weight calculation by utilizing the method proposed in Section 3.2. Finally, the method presented in Section 3.3 is used for the prioritization of test cases by acquiring input from Sections 3.1 and 3.2. A vector is obtained as output that denotes the test cases' order and aids software test personnel to make rational judgments.



**Figure 1.** A proposed research model for test case prioritization (TCP).

#### 4. Numerical Example—TCP in an SME

In this section, the reasonableness and usefulness of an integrated decision model are revealed by considering a software company's real case study. PTX (name anonymous) is a popular software company in Tamil Nadu that develops software related to the banking and commercial sector with core financial concepts. The company is a small and medium enterprise (SME) that has 52 personnel working across various platforms of software development and actively participate in the horizontal and vertical growth of the company. The company's main office is in Chennai, and it is spread around Tamil Nadu with varying workforces and projects. Around eight private commercial sectors are customers of PTX, and they have built a harmonious relationship with the company for around a decade.

A team of seven members works on the software development life cycle testing phase and has five to six years of experience in software testing and maintenance. The new software is close to its launch, and the company decided to conduct a comprehensive software testing before it is delivered to the market (customer) for live use. Based on the experience, the software personnel create a test suite with five crucial test cases that could identify faults that are critical with varying grades. Though the software needs many such test cases to identify different faults, these five test cases are substantial for the software under consideration. The faults identified by these test cases hinder the reliability aspect of the software. To make an apt call of test cases from the test suite, six criteria are put forward by the software test team. We conducted a detailed discussion with the personnel and based on the voting strategy, six potential criteria for evaluating the test cases are finalized. As a

part of our research work, we requested seven personnel for an interview slot, of which five personnel agreed to provide a slot for the interview. During this session, we asked the five members to share their preferences/grades over these five test cases' ability over the six criteria. Each expert/personnel linguistically shared his/her grade. We clarified the core reason behind data collection and how their data would be used in the research without deviating from the company's ethical policies.

To achieve integrity in the data collection process, we had two interview sessions and clarified our doubts with the software test team. In addition, the test team personnel clarified their queries related to our research and how the inferences would benefit the test team. For the sake of confidentiality, the names associated with the test cases and faults are kept anonymous. Let  $TC = (TC_1, TC_2, TC_3, TC_4, TC_5)$  be a set of five test cases that are evaluated with six criteria, viz., *reliance*, *fault coverage*, *the agility of execution*, *tractability*, *memory utilization*, and *cost* from the set  $F = (F_1, F_2, F_3, F_4, F_5, F_6)$ . The first four criteria are benefit type, and the rest are cost type. Here  $SP = (SP_1, SP_2, SP_3, SP_4, SP_5)$  is a set of five software test personnel who provided their grades for TCP. Each personnel shared the grade value, which is transformed to PLI using the procedure presented in Section 3.1. This procedure provides a holistic view of the data and retains the data integrity and adheres to ethical practices.

Figure 2 elaborates the proposed research model depicted in Figure 1. A clear view of the workflow of the proposed model is presented. Initially, a holistic data matrix (decision matrix) is obtained from the data collected from software personnel. This decision matrix adopts PLI as the preference structure as it reduces information loss and provides flexibility to software personnel during preference elicitation. Then, the criteria weight vector is determined with the help of the data provided by the personnel on each criterion. By using the holistic decision matrix and the weight vector, ranking is performed to determine the suitable test case for the process. The popular EDAS approach is extended to PLI for TCP. Average values from each criterion are determined, which are further used to calculate the PDA and NDA values for each test case. Finally, by adopting the linear combination principle, the rank values are calculated for each test case the prioritization/ranking order is obtained. A detailed stepwise workflow is show below for clarity to readers.

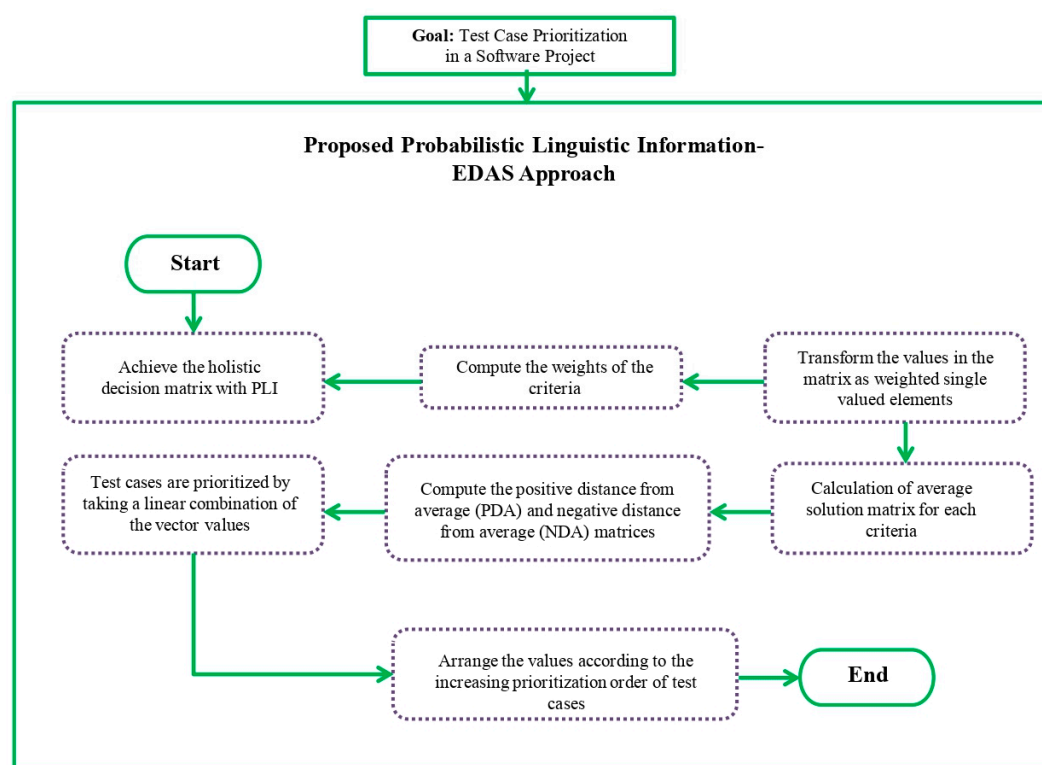


Figure 2. Flowchart depicting the workflow of the proposed model.



A stepwise procedure for rational TCP is given below.

**Step 1:** Collect data related to the test cases' performance with respect to each fault from the five personnel involved in the interview session. They provided data linguistically in the form of a seven-point Likert scale.

**Step 2:** The procedure described in Section 3.1 is adopted to transform the data to PLI to clearly gain potential information with a holistic data view.

Table 1 shows the PLI-based preference matrix obtained by transforming the linguistic data from experts by using the procedure provided in Section 3.1. In this way, a holistic view of the data is obtained for TCP.

**Table 1.** Transformed probabilistic linguistic information (PLI) for decision-making—TCP data.

Test Cases	Criteria for Evaluation					
	$F_1$	$F_2$	$F_3$	$F_4$	$F_5$	$F_6$
$TC_1$	$\left\{ \begin{matrix} s_3(0.4) \\ s_4(0.4) \end{matrix} \right\}$	$\left\{ \begin{matrix} s_5(0.4) \\ s_4(0.3) \end{matrix} \right\}$	$\left\{ \begin{matrix} s_3(0.35) \\ s_4(0.4) \end{matrix} \right\}$	$\left\{ \begin{matrix} s_4(0.4) \\ s_3(0.45) \end{matrix} \right\}$	$\left\{ \begin{matrix} s_4(0.35) \\ s_3(0.35) \end{matrix} \right\}$	$\left\{ \begin{matrix} s_3(0.35) \\ s_4(0.35) \end{matrix} \right\}$
$TC_2$	$\left\{ \begin{matrix} s_3(0.3) \\ s_5(0.45) \end{matrix} \right\}$	$\left\{ \begin{matrix} s_4(0.35) \\ s_3(0.55) \end{matrix} \right\}$	$\left\{ \begin{matrix} s_4(0.4) \\ s_5(0.3) \end{matrix} \right\}$	$\left\{ \begin{matrix} s_5(0.45) \\ s_3(0.25) \end{matrix} \right\}$	$\left\{ \begin{matrix} s_4(0.4) \\ s_5(0.5) \end{matrix} \right\}$	$\left\{ \begin{matrix} s_5(0.5) \\ s_4(0.25) \end{matrix} \right\}$
$TC_3$	$\left\{ \begin{matrix} s_5(0.4) \\ s_4(0.4) \end{matrix} \right\}$	$\left\{ \begin{matrix} s_5(0.45) \\ s_3(0.25) \end{matrix} \right\}$	$\left\{ \begin{matrix} s_4(0.35) \\ s_5(0.5) \end{matrix} \right\}$	$\left\{ \begin{matrix} s_2(0.35) \\ s_3(0.4) \end{matrix} \right\}$	$\left\{ \begin{matrix} s_5(0.6) \\ s_3(0.3) \end{matrix} \right\}$	$\left\{ \begin{matrix} s_4(0.35) \\ s_5(0.25) \end{matrix} \right\}$
$TC_4$	$\left\{ \begin{matrix} s_3(0.4) \\ s_4(0.5) \end{matrix} \right\}$	$\left\{ \begin{matrix} s_4(0.4) \\ s_3(0.3) \end{matrix} \right\}$	$\left\{ \begin{matrix} s_4(0.4) \\ s_5(0.35) \end{matrix} \right\}$	$\left\{ \begin{matrix} s_4(0.5) \\ s_3(0.45) \end{matrix} \right\}$	$\left\{ \begin{matrix} s_4(0.4) \\ s_3(0.35) \end{matrix} \right\}$	$\left\{ \begin{matrix} s_3(0.4) \\ s_5(0.45) \end{matrix} \right\}$
$TC_5$	$\left\{ \begin{matrix} s_5(0.4) \\ s_3(0.3) \end{matrix} \right\}$	$\left\{ \begin{matrix} s_4(0.4) \\ s_5(0.4) \end{matrix} \right\}$	$\left\{ \begin{matrix} s_3(0.45) \\ s_4(0.4) \end{matrix} \right\}$	$\left\{ \begin{matrix} s_4(0.6) \\ s_3(0.25) \end{matrix} \right\}$	$\left\{ \begin{matrix} s_5(0.35) \\ s_3(0.4) \end{matrix} \right\}$	$\left\{ \begin{matrix} s_4(0.5) \\ s_3(0.35) \end{matrix} \right\}$

**Step 3:** Three software personnel (out of five) also provided their data as complex verbal expressions for describing each criterion's importance. In the interview session, we asked the experts (software personnel) to share their opinions on each criterion's importance. By heuristics and discussion with experts, these expressions are transformed to PLI.

Table 2 depicts the opinions of experts on each criterion. This is in the form of PLI. By applying the procedure depicted in Section 3.2, entropy values are determined as shown above, and diversification values are obtained, which are normalized to get the weight vectors as 0.174, 0.169, 0.164, and 0.169, 0.152, and 0.172, respectively.

$$En_j = \begin{pmatrix} 0.3466 & 0.3017 & 0.2669 & 0.2999 & 0.1993 & 0.3605 \\ 0.3466 & 0.3465 & 0.3466 & 0.3467 & 0.3641 & 0.3466 \\ 0.3465 & 0.3658 & 0.3679 & 0.3660 & 0.3465 & 0.3238 \end{pmatrix}$$

**Table 2.** Weight calculation matrix for criteria with PLI.

Experts	Criteria					
	$F_1$	$F_2$	$F_3$	$F_4$	$F_5$	$F_6$
$SP_1$	$\left\{ \begin{matrix} s_4(0.4) \\ s_5(0.35) \end{matrix} \right\}$	$\left\{ \begin{matrix} s_4(0.45) \\ s_3(0.4) \end{matrix} \right\}$	$\left\{ \begin{matrix} s_4(0.5) \\ s_3(0.4) \end{matrix} \right\}$	$\left\{ \begin{matrix} s_5(0.35) \\ s_4(0.35) \end{matrix} \right\}$	$\left\{ \begin{matrix} s_4(0.45) \\ s_5(0.35) \end{matrix} \right\}$	$\left\{ \begin{matrix} s_3(0.35) \\ s_4(0.4) \end{matrix} \right\}$
$SP_2$	$\left\{ \begin{matrix} s_4(0.45) \\ s_3(0.4) \end{matrix} \right\}$	$\left\{ \begin{matrix} s_3(0.35) \\ s_5(0.5) \end{matrix} \right\}$	$\left\{ \begin{matrix} s_5(0.4) \\ s_4(0.25) \end{matrix} \right\}$	$\left\{ \begin{matrix} s_2(0.25) \\ s_3(0.5) \end{matrix} \right\}$	$\left\{ \begin{matrix} s_4(0.25) \\ s_3(0.45) \end{matrix} \right\}$	$\left\{ \begin{matrix} s_5(0.5) \\ s_4(0.25) \end{matrix} \right\}$
$SP_3$	$\left\{ \begin{matrix} s_5(0.35) \\ s_4(0.4) \end{matrix} \right\}$	$\left\{ \begin{matrix} s_4(0.25) \\ s_3(0.5) \end{matrix} \right\}$	$\left\{ \begin{matrix} s_2(0.5) \\ s_4(0.35) \end{matrix} \right\}$	$\left\{ \begin{matrix} s_4(0.45) \\ s_3(0.4) \end{matrix} \right\}$	$\left\{ \begin{matrix} s_5(0.35) \\ s_2(0.4) \end{matrix} \right\}$	$\left\{ \begin{matrix} s_3(0.35) \\ s_4(0.45) \end{matrix} \right\}$

**Step 4:** From Step 2, we obtain a matrix of order  $5 \times 6$ , which is used by the ranking method in Section 3.3 to form a prioritization order for the test cases. Similarly, Step 3 yields a matrix of order  $3 \times 6$  for the weight calculation of criteria where five software personnel provide their grades on six criteria as complex linguistic expressions.

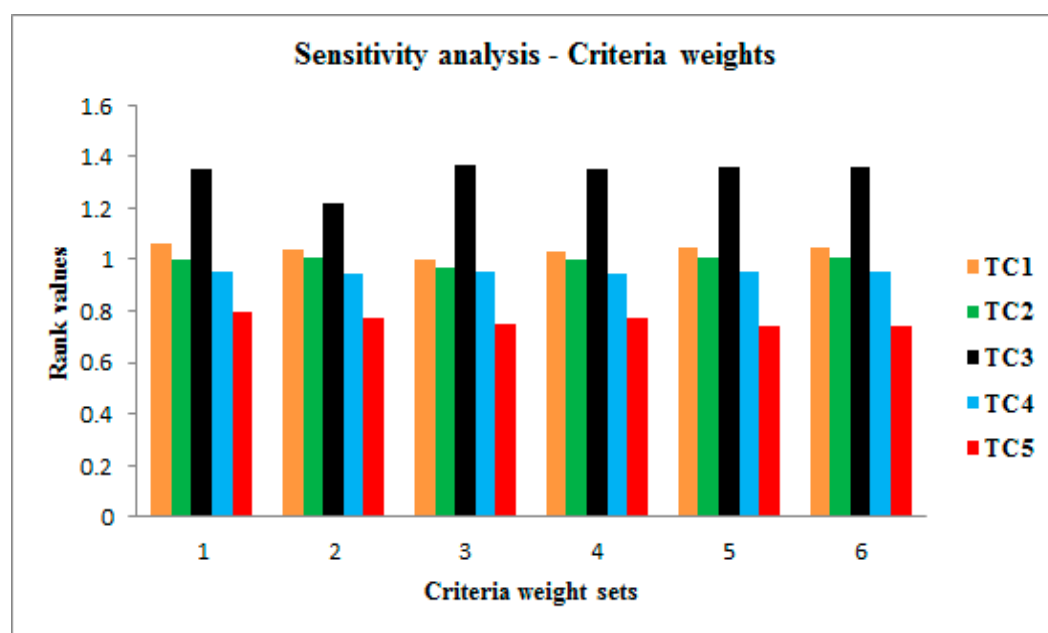
**Step 5:** A vector of order  $1 \times 6$  is obtained from Step 4, along with a matrix of order  $5 \times 6$ . By applying the procedure proposed in Section 3.3, test cases are prioritized to obtain a vector of order  $1 \times 5$ .

Table 3 clearly shows the parameter values of the EDAS approach under the PLI context. We obtain three vectors of order  $1 \times 5$ , and the vector from the last column in Table 3 depicts the rank value, and the order is given by  $TC_3 > TC_1 > TC_2 > TC_4 > TC_5$ .

**Table 3.** Parameter values of an evaluation based on distance from the average solution (EDAS) approach with PLI.

Test Cases	PDA	NDA	Rank Values
$TC_1$	1	0.05515	1.0552
$TC_2$	0	1	1
$TC_3$	0.9204	0.4290	1.3494
$TC_4$	0.7415	0.2089	0.9504
$TC_5$	0.7973	0	0.7973

**Step 6:** Conduct sensitivity analysis of criteria weight values by generating six sets of weight vectors with a single left shift operation. Figure 3 (*x-axis—different weight sets from shift operation*) depicts test cases' ranking order for all six sets. The figure shows that the ranking order does not change, and the proposed model is unaffected by criteria weight alteration with *high robustness* to weight changes.



**Figure 3.** Sensitivity analysis with six sets of criteria weights.

## 5. Comparative Investigation with Existing Models

This section tackles the comparative investigation of the introduced model with extant models. We conduct a comparison with respect to TCP and PLI models. State-of-the-art TCP models from Pradhan et al. [33], Shrivatsan et al. [34], and Baniyas [36] are considered for investigation with the proposed model. Table 4 provides the investigation with respect to different characteristics gathered from experts' intuition and the literature.

**Table 4.** Analysis of theoretical factors for different TCP models.

Factors	TCP Models			
	Proposed	Pradhan et al. [33]	Shrivatsan et al. [34]	Banias [36]
Data	PLI	Crisp data	Fuzzy data	Objective and subjective data
Weight calculation	Entropy measure	n/a	Not calculated; directly obtained	n/a
Prioritization method	EDAS approach	Rule mining and MO search	WASPAS method	Dynamic programming
Uncertainty	Handled effectively	Not captured	Slightly handled	Slightly handled
Data view	A holistic view of data from different experts	n/a	Fuzzified view of data	Human decisions are considered with objective values
Information overhead	No extra information is required Captured	Partial information is needed—constrained	No extra information is required	Partial information is needed—constrained
Experts' hesitation	effectively by using deviation measure	n/a	Not captured	Not captured
Information loss	Handled effectively	n/a	Not properly handled as linguistic data is transformed	n/a

Note: MO is multi-objective; WASPAS is a weighted arithmetic sum product assessment.

To better recognize the strength of the introduced work from the method perspective, we compare the framework with extant models like Sivagami et al.'s [13] model, Krishankumar et al.'s [12] model, and Zhang and Xing's [14] model. The preference order of the test cases is  $TC_3 > TC_1 > TC_2 > TC_4 > TC_5$ . Similarly, the prioritized order from Sivagami et al.'s [13] model is given by  $TC_3 > TC_1 > TC_4 > TC_2 > TC_5$ , Krishankumar et al.'s [12] model is given by  $TC_3 > TC_1 > TC_4 > TC_2 > TC_5$ , and Zhang and Xing's [14] model is given by  $TC_3 > TC_2 > TC_1 > TC_4 > TC_5$ . Using Spearman correlation, the coefficient values and the two-tailed significance value are determined for the ranks, and they are given by (1,0; 0.6, 0.285; 0.6, 0.285; 0.9, 0.037) for the proposed work versus other methods. From the values it is inferred that for the second and third pair needs additional samples to make further arguments that is planned for future.

Figure 4 shows that the introduced work is *consistent* with the extant models with the correlation, and rho values are shown above. On the x-axis, the labels 1, 2, 3, and 4 represent proposed vs. proposed, proposed vs. Sivagami et al.'s [13] model, proposed vs. Krishankumar et al.'s [12] model, and proposed vs. Zhang and Xing's model [14], respectively.

Some key outcomes of the introduced framework are given by

- The preference structure used in the paper for TCP is an innovative and flexible structure that only allows experts to share complex linguistic expressions and associate occurrence probability to each term. This enhances uncertainty handling and rational MCDM by providing a holistic view of the data from different experts.
- Criteria considered for evaluating TCP are competing and conflicting with each other, hence, weights are systematically calculated to mitigate bias and capture hesitation better.
- Moreover, test cases are prioritized systematically with broad rank values to make backup plans easily.
- Information loss is mitigated by avoiding the transformation of data that promotes rational prioritization of test cases.
- The proposed model also reduces the computational overhead by not acquiring additional data from experts in the form of constraints.

The model's limitations are (i) matrices are assumed to be the complete and systematic imputation of missing values are not considered; and (ii) cross-functional views of test cases from diverse software test teams are not considered in the present study.

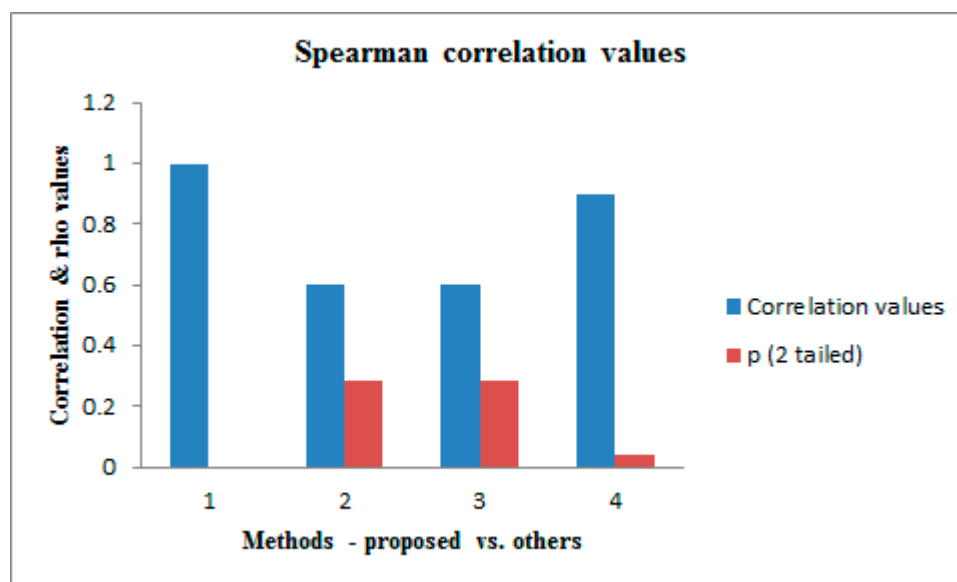


Figure 4. Correlation plot for consistency analysis.

## 6. Conclusions

This paper develops a new model for TCP by properly handling uncertainty with the help of PLI, which is a flexible structure to handle complex linguistic expressions. Opinions from different software test personnel are holistically represented in a preference matrix, and systematic prioritization of test cases is carried out. Entropy measures are proposed to calculate weights with reduced bias and proper handling of hesitation. Test cases are prioritized by using the EDAS approach to promote backup management during catastrophes. The proposed model is *highly robust* to weight alteration and produces *consistent ranking* compared to other models. These are evident from sensitivity analysis and Spearman correlation. Furthermore, the model produces *broad rank values* for effective backup plans that are evident from deviation analysis.

Some managerial implications of the study are (i) the proposed model is a readymade framework for deployment that could rationally perform TCP; (ii) experts involved in the process must be trained with the PLI structure and framework for effective utilization of the systematic tool; (iii) the model not only helps in TCP but also allows for the effective building of test cases for designing high-quality software; and finally, (iv) the tool could be flexibly adapted by the IT sector for other crucial decision-making problems such as strategy and risk management. As a future research direction, the model's limitations are planned to be addressed and machine learning approaches could be integrated with the proposed framework for enhanced decision-making in terms of test case evaluation and management. Additionally, the idea of comparative linguistic expressions [52,53] could be integrated with PLI for efficient handling of uncertainty during preference elicitation with ease and flexibility for experts to provide their preferences in a natural cognitive way that would promote rational decision-making.

**Author Contributions:** All authors have read the paper and accept its submission to the journal. The following are the authors' contributions. The first three authors A.D.S., R.K., and A.R.M. prepared the initial design of the proposed research model that was fine-tuned by the K.S.R. and S.K. Furthermore, K.S.R. and S.K. provided valuable advice and suggestions for coding, and A.D.S., R.K., and A.R.M. developed the complete code for the model. V.B. gathered the data needed for the validation of the code and helped us in refining the data with sufficient pre-processing. K.S.R., S.K., and V.B. discussed the model's overall workflow and provided crucial

improvements undertaken by A.D.S., R.K., and A.R.M. A.D.S., R.K., and A.R.M. prepared an initial draft of the paper that was completely refined with presentation improvement by K.S.R and V.B. S.K. and V.B. gave suggestions for improving the results section and presentation of tables and figures. K.S.R. and V.B. edited the language of the paper along with the fifth author. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no funding.

**Conflicts of Interest:** The authors declare no conflict of interest.

## Appendix A

**Table A1.** List of abbreviations with expansions.

Abbreviations	Expansions
MCDM	Multi-criteria decision-making
DM	Decision-makers
LTS	Linguistic term set
HFLT	Hesitant fuzzy linguistic term set
PLTS	Probabilistic linguistic term set
PLI	Probabilistic linguistic information
RT	Regression testing
TCM	Test case minimization
TCP	Test case prioritization
TCS	Test case selection
RTCP	Regression test case prioritization
AGCS	Additional greedy method call sequence
TCPCI	Test case prioritization in continuous integration
SLR	Systematic literature review
EDAS	Evaluation based on distance from the average solution

It must be noted that the base idea for the proofs are adapted from [44,45].

**Theorem A1.** Formulation in Equation (7) is a valid entropy measure that satisfies the following properties.

- (P1).  $En_j(th(p)) = 0$  if and only if  $h(p) = \{s_0(1)\}$  or  $th(p) = \{s_\beta(1)\}$ ;  $En_j(s_0(p), s_\beta(1-p)) = 0$ ;  
(P2).  $\eta_j(th(p)) = \max(En_j)$  when  $th(p) = (s_{\frac{\beta}{2}}(1))$  and  $\eta_j(th(p)) = En_j(th(p, 1-p))$ ;  
(P3).  $En_j(th^{(1)}(p)) \leq En_j(th^{(2)}(p))$  if  $th^{(1)}(p)$  and  $th^{(2)}(p)$  satisfy  $v_{lj}(th^{(1)}(p)) \leq v_{lj}(th^{(2)}(p)) \leq 0.5$  or  $v_{lj}(th^{(1)}(p)) \geq v_{lj}(th^{(2)}(p)) \geq 0.5$ ;  
(P4).  $\eta_j(th(p)) = \eta_j(th(p)^{(c)})$  if and only if  $\eta_j(th(p)) = En_j(th(p, 1-p))$ .

**Proof.** Let us consider  $v_i = \frac{D_{ij}}{D_{tot}}$  for proof purpose.

Given that  $v_0 = 0$  and  $v_\beta = \beta$  based on Equation (5), normalizing the values gives 0 and 1. In addition,  $En_j(0) = En_j(1) = 0$ . Thus,  $En_j(s_0(1)) = En_j(s_\beta(1)) = En_j(s_0(p), s_\beta(1-p)) = 0$ .

Given that  $v_{\beta/2} = 0.5$  and based on the binary entropic measure,  $\eta_j(s_{\frac{\beta}{2}}(1)) = \max(En_j)$ .

If  $s_i^{(1)} \leq s_i^{(2)} \leq s_{\beta/2}$ ,  $v_i^{(1)} \leq v_i^{(2)} \leq 0.5$ . From the formulation, it is evident that  $v_i^{(1)} \ln v_i^{(1)} \leq v_i^{(2)} \ln v_i^{(2)}$ , hence,  $En_j(th^{(1)}(p)) \leq En_j(th^{(2)}(p))$ . The second part may also be proved in a similar fashion.

By binary entropic measure, it is clear that  $En_j(th(p, 1-p)) = -\sum v_i \ln(v_i) + (-\sum (1-v_i) \ln(1-v_i))$ .

By taking complement, we get  $-\sum (1-v_i) \ln(1-v_i) + (-\sum 1 - (1-v_i) \ln 1 - (1-v_i)) = \eta_j(th(p)^{(c)})$ .

Thus,  $\eta_j(th(p)) = \eta_j(th(p)^{(c)})$ . All four properties are proved. (P2) and (P4) adopt a binary entropic measure formulation for deriving efficient proofs.  $\square$

## References

1. Triantaphyllou, E.; Shu, B. Multi-criteria decision making: An operations research approach. *Encycl. Electr. Electron. Eng.* **1998**, *15*, 175–186.
2. Zadeh, L.A. The concept of a linguistic variable and its application to approximate reasoning-I. *Inf. Sci.* **1975**, *8*, 199–249. [[CrossRef](#)]
3. Herrera, F.; Herrera-Viedma, E. Linguistic decision analysis: Steps for solving decision problems under linguistic information. *Fuzzy Sets Syst.* **2000**, *115*, 67–82. [[CrossRef](#)]
4. Rodriguez, R.M.; Martinez, L.; Herrera, F. Hesitant fuzzy linguistic term sets for decision making. *IEEE Trans. Fuzzy Syst.* **2012**, *20*, 109–119. [[CrossRef](#)]
5. Torra, V. Hesitant Fuzzy Sets. *Int. J. Intell. Syst.* **2010**, *25*, 529–539. [[CrossRef](#)]
6. Alcantud, J.C.R.; Santos-García, G.; Peng, X.; Zhan, J. Dual extended hesitant fuzzy sets. *Symmetry* **2019**, *11*, 714. [[CrossRef](#)]
7. Liao, H.; Xu, Z.; Herrera-Viedma, E.; Herrera, F. Hesitant Fuzzy Linguistic Term Set and Its Application in Decision Making: A State-of-the-Art Survey. *Int. J. Fuzzy Syst.* **2018**, *20*, 2084–2110. [[CrossRef](#)]
8. Pang, Q.; Wang, H.; Xu, Z. Probabilistic linguistic term sets in multi-attribute group decision making. *Inf. Sci.* **2016**, *369*, 128–143. [[CrossRef](#)]
9. Liu, P.; Li, Y. Multi-attribute decision making method based on generalized maclaurin symmetric mean aggregation operators for probabilistic linguistic information. *Comput. Ind. Eng.* **2019**, *131*, 282–294. [[CrossRef](#)]
10. Liu, P.; Teng, F. Some Muirhead mean operators for probabilistic linguistic term sets and their applications to multiple attribute decision-making. *Appl. Soft Comput. J.* **2018**, *68*, 396–431. [[CrossRef](#)]
11. Lin, M.; Xu, Z. Probabilistic linguistic distance measure and their applications in multi-criteria group decision-making. In *Soft Computing Applications for Group Decision-Making and Consensus Modeling*; Springer: Berlin, Germany, 2017; pp. 411–440.
12. Krishankumar, R.; Ravichandran, K.; Ahmed, M.; Kar, S.; Tyagi, S. Probabilistic Linguistic Preference Relation-Based Decision Framework for Multi-Attribute Group Decision Making. *Symmetry* **2018**, *11*, 2. [[CrossRef](#)]
13. Sivagami, R.; Ravichandran, K.S.; Krishankumar, R.; Sangeetha, V.; Kar, S.; Gao, X.Z.; Pamucar, D. A scientific decision framework for cloud vendor prioritization under probabilistic linguistic term set context with unknown/partialweight information. *Symmetry* **2019**, *11*, 682. [[CrossRef](#)]
14. Zhang, X.; Xing, X. Probabilistic Linguistic VIKOR Method to Evaluate Green Supply Chain Initiatives. *Sustainability* **2017**, *9*, 1231. [[CrossRef](#)]
15. Gou, X.; Xu, Z. Novel basic operational laws for linguistic terms, hesitant fuzzy linguistic term sets and probabilistic linguistic term sets. *Inf. Sci.* **2016**, *372*, 407–427. [[CrossRef](#)]
16. Liao, H.; Mi, X.; Xu, Z. A survey of decision-making methods with probabilistic linguistic information: Bibliometrics, preliminaries, methodologies, applications and future directions. *Fuzzy Optim. Decis. Mak.* **2020**, *19*, 81–134. [[CrossRef](#)]
17. Zhao, Y.; Serebrenik, A.; Zhou, Y.; Filkov, V.; Vasilescu, B. The impact of continuous integration on other software development practices: A large-scale empirical study. In Proceedings of the 32nd IEEE/ACM International Conference on Automated Software Engineering, Urbana, IL, USA, 30 October–3 November 2017; pp. 60–71. [[CrossRef](#)]
18. Jiang, B.; Chan, W.K. Testing and debugging in continuous integration with bud- get quotas on test executions. In Proceedings of the IEEE International Conference on Software Quality, Reliability and Security, Vienna, Austria, 1–3 August 2016; pp. 439–447. [[CrossRef](#)]
19. Haghighatkhah, A.; Mäntylä, M.; Oivo, M.; Kuvaja, P. Test prioritization in continuous integration environments. *J. Syst. Softw.* **2018**, *146*, 80–98. [[CrossRef](#)]
20. Yoo, S.; Harman, M. Regression testing minimization, selection and prioritization: A Survey. *Softw. Test. Verif. Reliab.* **2012**, *22*, 67–120. [[CrossRef](#)]
21. Rothermel, G.; Untch, R.H.; Harrold, M.J. Test case prioritization: An empirical study. In Proceedings of the 15th IEEE International Conference on Software Maintenance (ICSM'99), Oxford, UK, 30 August–3 September 1999; pp. 179–188.



22. Li, Z.; Harman, M.; Hierons, R.M. Search algorithms for regression test case prioritization. *IEEE Trans. Softw. Eng.* **2007**, *33*, 225–237. [\[CrossRef\]](#)
23. Jiang, B.; Zhang, Z.; Chan, W.K.; Tse, T. Adaptive random test case prioritization. In Proceedings of the 24th IEEE/ACM International Conference on Automated Software Engineering (ASE'09), Auckland, New Zealand, 6–20 November 2009; pp. 233–244.
24. Mei, H.; Hao, D.; Zhang, L.; Zhang, L.; Zhou, J.; Rothermel, G. A static approach to prioritizing junit test Cases. *IEEE Trans. Softw. Eng.* **2012**, *38*, 1258–1275. [\[CrossRef\]](#)
25. Saha, R.K.; Zhang, L.; Khurshid, S.; Perry, D.E. An information retrieval approach for regression test prioritization based on program changes. In Proceedings of the 37th IEEE/ACM IEEE International Conference on Software Engineering (ICSE'15), Florence, Italy, 16–24 May 2015; pp. 268–279.
26. Hao, D.; Zhang, L.; Zhang, L.; Rothermel, G.; Mei, H. A unified test case prioritization approach. *ACM Trans. Softw. Eng. Methodol.* **2014**, *24*, 10:1–10:31. [\[CrossRef\]](#)
27. Hao, D.; Zhang, L.; Zang, L.; Wang, Y.; Wu, X.; Xie, T. To be optimal or not in test-case prioritization. *IEEE Trans. Softw. Eng.* **2016**, *42*, 490–505. [\[CrossRef\]](#)
28. Tahvili, B.S.; Bohlin, M. Test Case Prioritization Using Multi Criteria Decision Making Methods. *Dan. Soc. Oper. Res.* **2016**, *26*, 9–11.
29. Zhang, L.; Hao, D.; Zhang, L.; Rothermel, G.; Mei, H. Bridging the gap between the total and additional test-case prioritization strategies. In Proceedings of the 2013 International Conference on Software Engineering (ICSE'13), San Francisco, CA, USA, 18–26 May 2013; pp. 192–201.
30. Wang, S.; Nam, J.; Tan, L. QTEP: Quality-aware test case prioritization. In Proceedings of the 11th Joint Meeting on Foundations of Software Engineering (ESEC/FSE'17), Paderborn, Germany, 4–8 September 2017; pp. 523–534.
31. Zhu, H.; Hall, P.A.V.; May, J.H.R. Software unit test coverage and adequacy. *ACM Comput. Surv.* **1997**, *29*, 366–427. [\[CrossRef\]](#)
32. Huang, R.; Sun, W.; Xu, Y.; Chen, H.; Towey, D.; Xia, X. A survey on adaptive random testing. *IEEE Trans. Softw. Eng.* **2019**. [\[CrossRef\]](#)
33. Pradhan, D.; Wang, S.; Ali, S.; Yue, T.; Liaaen, M. Employing Rule Mining and Multi-Objective Search for Dynamic Test Case Prioritization. *J. Syst. Softw.* **2019**, *153*, 86–104. [\[CrossRef\]](#)
34. Shrivathsan, A.D.; Ravichandran, K.S.; Krishankumar, R.; Sangeetha, V.; Kar, S.; Ziemba, P.; Jankowski, J. Novel Fuzzy Clustering Methods for Test Case Prioritization in Software Projects. *Symmetry* **2019**, *11*, 1400. [\[CrossRef\]](#)
35. Khatibsyarhini, M.; Isa, M.A.; Jawawi, D.N.A.; Tumeng, R. Test case prioritization approaches in regression testing: A systematic literature review. *Inf. Softw. Technol.* **2018**, *93*, 74–93. [\[CrossRef\]](#)
36. Baniyas, O. Test case selection-prioritization approach based on memoization dynamic programming algorithm. *Inf. Softw. Technol.* **2019**, *115*, 119–130. [\[CrossRef\]](#)
37. Chi, J.; Yu, Q.; Zheng, Q.; Yang, Z.; Jin, W.; Cui, D.; Liu, T. Relation-based test case prioritization for regression testing. *J. Syst. Softw.* **2020**, *163*, 110539. [\[CrossRef\]](#)
38. Lima, J.A.P.; Vergilio, S.R. Test Case Prioritization in Continuous Integration Environments: A Systematic Mapping Study. *Inf. Softw. Technol.* **2020**, 106268. [\[CrossRef\]](#)
39. Huang, R.; Zhang, Q.; Towey, D.; Sun, W.; Chen, J. Regression test case prioritization by code combinations coverage. *J. Syst. Softw.* **2020**, *169*, 110712. [\[CrossRef\]](#)
40. Mahdieh, M.; Mirian-Hosseiniabadi, S.H.; Etemadi, K.; Nosrati, A.; Jalali, S. Incorporating fault-proneness estimations into coverage-based test case prioritization methods. *Inf. Softw. Technol.* **2020**, *121*, 106269. [\[CrossRef\]](#)
41. Xie, W.; Xu, Z.; Ren, Z.; Wang, H. Probabilistic Linguistic Analytic Hierarchy Process and Its Application on the Performance Assessment of Xiongan New Area. *Int. J. Inf. Technol. Decis. Mak.* **2018**, *16*, 1–32. [\[CrossRef\]](#)
42. Yuan, J.; Luo, X. Approach for multi-attribute decision making based on novel intuitionistic fuzzy entropy and evidential reasoning. *Comput. Ind. Eng.* **2019**, *135*, 643–654. [\[CrossRef\]](#)
43. Verma, R. Multiple attribute group decision-making based on order- $\alpha$  divergence and entropy measures under q-rung orthopair fuzzy environment. *Int. J. Intell. Syst.* **2020**, *35*, 718–750. [\[CrossRef\]](#)
44. Lin, M.; Chen, Z.; Liao, H.; Xu, Z.S. ELECTRE II method to deal with probabilistic linguistic term sets and its application to edge computing. *Nonlinear Dyn.* **2019**, *96*, 2125–2143. [\[CrossRef\]](#)

45. Xu, G.L.; Wan, S.P.; Dong, J.Y. An Entropy-Based Method for Probabilistic Linguistic Group Decision Making and its Application of Selecting Car Sharing Platforms. *Informatica* **2020**, *31*, 621–658. [\[CrossRef\]](#)
46. Ghorabae, M.K.; Zavadskas, E.K.; Olfat, L.; Turskis, Z. Multi-Criteria Inventory Classification Using a New Method of Evaluation Based on Distance from Average Solution (EDAS). *Informatica* **2015**, *26*, 435–451. [\[CrossRef\]](#)
47. Karaşan, A.; Kahraman, C. A novel interval-valued neutrosophic EDAS method: Prioritization of the United Nations national sustainable development goals. *Soft Comput.* **2018**, *22*, 4891–4906. [\[CrossRef\]](#)
48. Peng, X.; Liu, C. Algorithms for neutrosophic soft decision making based on EDAS, new similarity measure and level soft set. *J. Intell. Fuzzy Syst.* **2017**, *32*, 955–968. [\[CrossRef\]](#)
49. Mishra, A.R.; Mardani, A.; Rani, P.; Zavadskas, E.K. A novel EDAS approach on intuitionistic fuzzy set for assessment of health-care waste disposal technology using new parametric divergence measures. *J. Clean. Prod.* **2020**, 122807. [\[CrossRef\]](#)
50. Liang, Y. An EDAS method for multiple attribute group decision-making under intuitionistic fuzzy environment and its application for evaluating green building energy-saving design projects. *Symmetry* **2020**, *12*, 484. [\[CrossRef\]](#)
51. Feng, X.; Wei, C.; Liu, Q. EDAS Method for Extended Hesitant Fuzzy Linguistic Multi-criteria Decision Making. *Int. J. Fuzzy Syst.* **2018**, *20*, 2470–2483. [\[CrossRef\]](#)
52. Rodríguez, R.M.; Martínez, L.; Herrera, F. A group decision making model dealing with comparative linguistic expressions based on hesitant fuzzy linguistic term sets. *Inf. Sci.* **2013**, *241*, 28–42. [\[CrossRef\]](#)
53. Liu, Y.; Rodríguez, R.M.; Alcantud, J.C.R.; Qin, K.; Martínez, L. Hesitant linguistic expression soft sets: Application to group decision making. *Comput. Ind. Eng.* **2019**, *136*, 575–590. [\[CrossRef\]](#)

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).