A VIKOR-based group decision-making approach to software reliability evaluation

Chuan Yue

Accepted: 16 May 2022 / Published online: 14 July 2022
© The Author(s), under exclusive licence to Springer-Verlag GmbH Germany, part of Springer Nature 2022

Abstract
Software reliability evaluation is important to attribute of software quality. A new group decision-making is provided to evaluate software reliability, where the model is based on an extended VIsekriterijumska optimizacija i Kompromisno Resenje technique. The individual utility and individual regret in general VIKOR technique are extended to group utility and group regret. A specific regret matrix is provided in new VIKOR-based GDM method. A new ranking method is provided in a static environment. Another new ranking method is provided in a dynamic environment. To implement a user-based evaluation for software reliability, the picture fuzzy set is used for handling the questionnaire information. It is noted that the existing projection measure is not always reasonable in picture fuzzy setting. To solve this problem, a new normalization projection measure is provided in picture fuzzy setting. And a new GDM method is established for software reliability evaluation. The feasibility and practicability developed method in this work are illustrated by an experimental analysis.

Keywords Group decision-making · Extended VIKOR method · Group regret matrix · Normalization projection · Picture fuzzy number · Software reliability

1 Introduction
Software reliability is one of the most important factors in the software development process, which directly impacts on software quality (Utkin and Coolen 2018). It is defined as the probability of failure-free software operation for a specified period of time in a specified environment (Lyu 1996). Software reliability is important to attribute of software quality, together with functionality, usability, performance, serviceability, capability, installability, maintainability, and documentation. Software reliability is hard to achieve, because the complexity of software tends to be high. Many software reliability researches have contributed to literature. Assessing software reliability is an important issue in the modern software development process. However, to the best of the author’s knowledge, no good quantitative methods were developed to represent software reliability without excessive limitations. The subsequent study in this work finds that the following some questions need further exploration.

Question 1 The existing literature is lack for the users-based models and methods for software reliability. In particular, the existing literature is lack for the users’ satisfaction and expectations.

If the evaluated software is viewed as an alternative, then the preference of software reliability can be identified by multi-attribute decision-making (MADM) (Yue 2016) method. If the users of software are viewed as experts, then the preference of software reliability can be identified by group decision-making (GDM) (Yue and Yue 2019; Büyüközkan and Güler 2020; Yue 2014, 2011b, a) method. So, this work attempts to solve this question by using a more comprehensive GDM method.

Question 2 It is noted that user survey is the most common method for knowing the status of users’ satisfaction. However, there is less information from questionnaires. The information from questionnaires may only be “agree”, “disagree”, “abstain” and “refusing to survey”.

For Question 2, picture fuzzy set can just solve this problem.

Chuan Yue
yuechuan-1988@163.com

1 College of Mathematics and Computer Science, Guangdong Ocean University, Zhanjiang, China
Projection measure is a comprehensive measurement, which can measure not only the distance but also the angle between two decision objects (Wei et al. 2018). However, this research finds the following Question 3.

**Question 3** The projection measure is not always reasonable in picture fuzzy (Opricovic 1998; Cuong 2014) setting (see the following Example 2 in Sect. 4).

The VIKOR (VIsekriterijumska optimizacija i KOmpro-mismo Resenje) (Opricovic and Tzeng 2004) is a very useful method for MADM and GDM problems. However, this work finds the following Question 4.

**Question 4** The VIKOR method is not very convenient to use in GDM environment. The individual utility and individual regret are two important references in existing VIKOR-based GDM method. In theory, the reference of a GDM problem should be a group’s reference. In other words, they should be the group utility and the group regret.

Based on above four questions, corresponding research contributions in this paper are raised as follows:

**Contribution 1** For the Questions 1, a new evaluation method of software reliability is developed. This new method is based on GDM and users’ expectations.

**Contribution 2** For the Question 2, a GDM approach with picture fuzzy information (Cuong 2014) is developed in order to quantify the users’ satisfaction and expectations.

**Contribution 3** For the Question 3, a new normalization projection measure is developed in picture fuzzy setting.

**Contribution 4** For the Question 4, an extended VIKOR method is developed, where the ideal decision is a decision matrix. The individual utility vector is extended to a group utility matrix; the individual regret vector is extended to a group regret matrix in the developed VIKOR method. A specific regret matrix is provided in the new VIKOR-based GDM method. A new ranking method under a static environment and a new ranking method in a dynamic environment are provided in this work. And the new GDM method is applied to software reliability evaluation.

A detailed introduction of this work is listed as follows. Section 2 introduces the related work. Section 3 introduces the preliminary knowledge. Section 4 presents a normalization projection measure. Section 5 develops a VIKOR-based GDM method, where the algorithmic pseudocodes are also attached in detail. Section 6 provides a detailed assessment process and an experimental analysis of software reliability. Finally, Sect. 7 draws conclusions and future researches.

### 2 Related work

Three related researches are introduced in this section. One of related researches is software reliability assessment; another work is VIKOR-based GDM methods; the third work is projection measures. The main weaknesses of current work and research motivations of this work are also mentioned.

There are many researches on software reliability (Zhao et al. 2021; Rahman et al. 2021; Mohammazadeh et al. 2021; Mahmoudi et al. 2021; Islam et al. 2021). For example, Kang et al. (2018) developed a Bayesian belief network model for software reliability in nuclear power plants. Sinha et al. (2019) introduced an early prediction method of hardware-software reliability based on functional failures. Cho et al. (2019) proposed an exhaustive test cases for the software reliability of safety-critical digital systems. Huo and Li (2019) modeled a cost-effective-based software defect prediction method. Kaliraj and Bharathi (2019) explored a path testing based reliability analysis framework. Bertsatos et al. (2019) described a testing of 3D-ID software reliability in sex and ancestry estimation. Lanna et al. (2018) focused a feature-family-based reliability analysis of software product line. Greisberger et al. (2019) suggested an interrater reliability of angular measures using TEMPLO two-dimensional motion analysis software. Levitin et al. (2019) focused an optimization of partial software rejuvenation policy. Shan et al. (2019) developed a software structure characteristic measurement method based on weighted network. Guo et al. (2018) presented a software reliability demonstration for nuclear safety-critical Digital Instrumentation and Control system. Mozaveni et al. (2018) proposed a reliability improvement of software-defined networks. Dohi et al. (2018) described an optimal periodic software rejuvenation policies based on interval reliability criteria. Zou et al. (2018) proposed a software reliability hierarchical structure modeling in I&C system Software Life Cycle. Abuta and Tian (2018) addressed the reliability over consecutive releases of a semiconductor Optical Endpoint Detection software system. Using a deep learning model, Wang and Zhang (2018) developed a software reliability prediction based on the RNN encoder-decoder.

These studies have greatly contributed to software reliability researches. However, this work finds that the software reliability assessment is multi-dimensional. MADM method and GDM method able to deal with it better. The software reliability assessments are less based on decision science.

The VIKOR is one of the common decision-making methods, which is often used in MADM (Curiel-Esparza et al. 2019; Štribanović et al. 2019; Chen 2016). A part of VIKOR-based MADM is used in GDM. For example, some VIKOR-based decision support systems are introduced in fuzzy environments (Ploskas and Papathanasiou 2019; Ren et al. 2017). The VIKOR-based GDM methods (Büyüközkan
et al. 2019; Çali and Balaman 2019) are introduced by some scholars under intuitionistic fuzzy environment. Wu et al. (2019) proposed a VIKOR-based GDM approach under interval type-2 fuzzy environment. The VIKOR-based GDM technologies with Pythagorean fuzzy information (Chen 2018; Liang et al. 2019; Wu et al. 2019) are shared by scholars. The scholars (Wu et al. 2019; Büyüközkan and Güler 2020; Wu et al. 2016; You et al. 2015) developed some VIKOR-based GDM methods with linguistic information. The researcher (Gupta et al. 2016) modeled two VIKOR methods for GDM problems with intuitionistic fuzzy information. Yue (2020b) suggested a VIKOR approach to software reliability assessment in GDM setting. However, as mentioned in Question 4 in Introduction section, the group utility and group regret are not present in the traditional VIKOR-based GMD method. This research attempts to solve this question.

Projection measures have been attracted by scholars. Some of them are used in MADM problems. Another work is used in GDM problems. For example, Tsao and Chen (2016) addressed a projection-based compromising method for MADM with interval-valued intuitionistic fuzzy information. Wang et al. (2020) explored a projection-based regret theory method for MADM under interval type-2 fuzzy sets environment. Li et al. (2018) presented a grey correlation projection-based MADM approach to economic emission dispatch. Wang et al. (2020) offered a bi-projection MADM model with linguistic terms. Wu et al. (2018) performed a hesitant fuzzy linguistic projection model to a MADM problem. Tang et al. (2020) contributed a projection-based MADM approaches for evaluating the service quality of Chinese commercial banks.

Some projection-based GDM researches also attracted the attention of researchers. For example, Liao et al. (2018) focused a projection-based distance measure for GDM in intuitionistic fuzzy setting. Xu and Liu (2013) conducted a projection-based GDM method with interval fuzzy information. The researches (Liu and You 2019; Ye 2017) surveyed two bidirectional projection measures for GDM with neutrosophic numbers. Ju and Wang (2013) introduced a projection method for GDM with incomplete weight information in linguistic setting.

In recent years, some normalized projection measures have contributed to decision science by scholars. For example, Wan et al. (2018) observed a normalized projection-based GDM method with Pythagorean fuzzy information. Yue and Jia (2017) proposed a direct projection-based GDM methodology with crisp values and interval data. Yue (2017b,a, 2019a,b,c, 2020,b) developed some normalized projection measures to GDM problems with different decision information.

These studies have greatly enriched and strengthened the MADM and GDM problems. However, as mentioned in Question 3 in Introduction section, this research finds that the projection measure is not always reasonable in picture fuzzy setting. To solve this problem, this research attempts to develop a new normalized projection measure.

3 Preliminaries

Definition 1 A picture fuzzy set (PFS) (Cuong 2014) on a universe \(X\) is interpreted as \(A = \{(x, \mu_A(x), \eta_A(x), \nu_A(x)) : x \in X\}\), where \(\mu_A(x)\) is positive degree of membership, \(\eta_A(x)\) is the neutral of membership, \(\nu_A(x)\) is the negative degree of membership, such that \(0 \leq \mu_A(x), \eta_A(x), \nu_A(x) \leq 1, \forall x \in X\). Also, the refusal degree is calculated as \(\pi_A(x) = 1 - \mu_A(x) - \eta_A(x) - \nu_A(x), \forall x \in X\). Especially, if \(\pi_A(x) = 0\), then the PFS \(A\) returns to an intuitionistic fuzzy set (Atanassov 1986). If \(\pi_A(x) = 0\) and \(\eta_A(x) = 0\), then the PFS \(A\) returns to a fuzzy set (Zadeh 1965).

For convenience, Wei (2017) called the array of three numbers

\[a = (\mu_a, \eta_a, \nu_a)\]  

a picture fuzzy number (PFN), where \(0 \leq \nu_a, \eta_a, \mu_a \leq 1, 0 \leq \mu_a + \eta_a + \nu_a \leq 1, \pi_a = 1 - \mu_a - \eta_a - \nu_a\).

A PFN can be explained by the following a real-world example.

Example 1 In a democratic election station, the council issues 100 voting papers for a candidate. The voting results are divided into four groups accompanied with the number of papers that are “vote for” (70), “abstain” (6), “vote against” (20) and “refusal of voting” (4). Group “abstain” means that the voting paper is a white paper rejecting both “agree” and “disagree” for the candidate but still takes the vote. Group “refusal of voting” is either invalid voting papers or did not take the vote. The voting results can be written as a PFN \(a\) = (0.7, 0.06, 0.2), where \(\pi_a = 1 - 0.7 - 0.06 - 0.2 = 0.04\).

Some operations of PFNs are provided by scholars as follows.

Definition 2 Let \(a = (\mu_a, \eta_a, \nu_a)\) and \(b = (\mu_b, \eta_b, \nu_b)\) be two PFNs, then

1. \(a \wedge b = (\min\{\mu_a, \mu_b\}, \max\{\eta_a, \eta_b\}, \max\{\nu_a, \nu_b\})\) (Wei 2017);
2. \(a \vee b = (\max\{\mu_a, \mu_b\}, \min\{\eta_a, \eta_b\}, \min\{\nu_a, \nu_b\})\) (Wei 2017);
3. \(a \otimes b = ((\mu_a + \eta_a)\mu_b + \nu_b) - \eta_a\eta_b, \eta_a\eta_b, 1 - (1 - \nu_a)(1 - \nu_b))\) (Zhang et al. 2018);
4. \[ \lambda \alpha = (1 - (1 - \mu_d)^2, (\eta_d)^2, (\eta_d + \nu_d)^2 - (\eta_d)^2)(\lambda > 0) \] (Zhang et al. 2018).

**Definition 3** Let \( X = (x_{ij})_{m \times n} \) be a matrix. If all the \( x_{ij} \) are PFNs, then \( X \) is called a picture fuzzy matrix (PFM). Especially, if \( m = 1 \) or \( n = 1 \), then \( X \) is called a picture fuzzy vector (PFV).

The projection measure (Yue and Jia 2015, 2017) is an important tool for measuring the closeness between two evaluation matrices. A projection measure of a PFV on another was introduced by Wei et al. (2018), as shown in the following Definition 4.

**Definition 4** Let \( \tilde{\alpha} = (\alpha_0, \alpha_1, \ldots, \alpha_n) \) and \( \tilde{\beta} = (\beta_0, \beta_1, \ldots, \beta_n) \) be two PFVs, where \( \alpha_j = (\mu_{\alpha}, \eta_{\alpha}, \nu_{\alpha}) \) and \( \beta_j = (\mu_{\beta}, \eta_{\beta}, \nu_{\beta}) \) \((j = 1, 2, \ldots, n)\), then

\[
\tilde{\alpha} \cdot \tilde{\beta} = \sum_{j=1}^{n} \left( \mu_{\alpha} \cdot \mu_{\beta} + \eta_{\alpha} \cdot \eta_{\beta} + \nu_{\alpha} \cdot \nu_{\beta} + \pi_{\alpha} \cdot \pi_{\beta} \right)
\]

is called the inner product between \( \tilde{\alpha} \) and \( \tilde{\beta} \), where \( \pi_{\alpha} = 1 - \mu_{\alpha} - \eta_{\alpha} - \nu_{\alpha} \), \( \pi_{\beta} = 1 - \mu_{\beta} - \eta_{\beta} - \nu_{\beta} \) \((j = 1, 2, \ldots, n)\). And

\[
|\tilde{\beta}| = \sqrt{\sum_{j=1}^{n} (\mu_{\beta}^2 + \eta_{\beta}^2 + \nu_{\beta}^2 + \pi_{\beta}^2)}
\]

is the module of \( \tilde{\beta} \), where \( \pi_{\beta}(j = 1, 2, \ldots, n) \) is the same as in Eq. (2). And

\[
\text{Proj}_{\tilde{\beta}}(\tilde{\alpha}) = \frac{\tilde{\alpha} \cdot \tilde{\beta}}{|\tilde{\beta}|}
\]

is called the projection of \( \tilde{\alpha} \) onto \( \tilde{\beta} \).

**Criterion 1** The greater the value \( \text{Proj}_{\tilde{\beta}}(\tilde{\alpha}) \), the closer the \( \tilde{\alpha} \) is to \( \tilde{\beta} \) (Wei et al. 2018).

**4 Presented projection measure**

As mentioned in the Introduction, the projection is a useful and comprehensive measure. However, this research finds that Eq. (4) does not always conform to the Criterion 1, as shown in the following Example 2.

**Example 2** Let \( \tilde{\alpha} = ((0.900, 0.040, 0.005), (0.898, 0.070, 0.002)) \) and \( \tilde{\beta} = ((0.793, 0.162, 0.002), (0.796, 0.200, 0.030)) \) be two PFVs. It is obvious that \( \tilde{\alpha} \neq \tilde{\beta} \). Generally, it should have the result \( \text{Proj}_{\tilde{\beta}}(\tilde{\beta}) > \text{Proj}_{\tilde{\beta}}(\tilde{\alpha}) \).

However, we have \( \tilde{\alpha} \cdot \tilde{\beta} = 1.4514 \) based on Eq. (2); \( |\tilde{\alpha}| = 1.2755, |\tilde{\beta}| = 1.1535 \) based on Eq. (3). It follows that \( \text{Proj}_{\tilde{\beta}}(\tilde{\alpha}) = 1.2583, \text{Proj}_{\tilde{\beta}}(\tilde{\beta}) = 1.1535 \) based on Eq. (4). In this case, we have \( \text{Proj}_{\tilde{\beta}}(\tilde{\beta}) < \text{Proj}_{\tilde{\beta}}(\tilde{\alpha}) \). This is an obvious contradiction.

From the Example 2, some research questions (RQs) are found as follows.

- **RQ 1**: The closeness between two PFVs according to Eq. (4) does not always conform to the Criterion 1.
- **RQ 2**: Eq. (4) is not a normalization measure. That is, the projection in Eq. (4) does not always conform to the condition \( 0 \leq \text{Proj}_{\tilde{\beta}}(\tilde{\alpha}) \leq 1 \). For example, as shown in Example 2, we have that \( \text{Proj}_{\tilde{\beta}}(\tilde{\alpha}) = 1.2583 \).
- **RQ 3**: According to Eq. (4), two projection measurements \( \text{Proj}_{\tilde{\alpha}}(\tilde{\alpha}) \) and \( \text{Proj}_{\tilde{\beta}}(\tilde{\beta}) \) may be unequal. For example, we can see in Example 2 that \( \tilde{\alpha} \cdot \tilde{\beta} = 1.6269, \tilde{\beta} \cdot \tilde{\beta} = 1.3306 \) based on Eq. (2); \( |\tilde{\alpha}| = 1.2755, |\tilde{\beta}| = 1.1535 \) based on Eq. (3), \( \text{Proj}_{\tilde{\alpha}}(\tilde{\alpha}) = 1.2755 \) and \( \text{Proj}_{\tilde{\beta}}(\tilde{\beta}) = 1.1535 \) based on Eq. (4). It is obvious that \( \text{Proj}_{\tilde{\alpha}}(\tilde{\alpha}) \neq \text{Proj}_{\tilde{\beta}}(\tilde{\beta}) \).

That leads to the following research objectives (ROs). Establish a projection measure of a PFV \( \tilde{\alpha} \) on another \( \tilde{\beta} \) such that

- **RO 1**: \( 0 \leq \text{Proj}_{\tilde{\beta}}(\tilde{\alpha}) \leq 1 \) for all the PFVs \( \tilde{\alpha} \) and \( \tilde{\beta} \).
- **RO 2**: The larger the value \( \text{Proj}_{\tilde{\beta}}(\tilde{\alpha}) \), the closer the \( \tilde{\alpha} \) is to the \( \tilde{\beta} \).
- **RO 3**: \( \text{Proj}_{\tilde{\beta}}(\tilde{\alpha}) = \text{Proj}_{\tilde{\beta}}(\tilde{\beta}) = 1 \) for all the PFVs \( \tilde{\alpha} \) and \( \tilde{\beta} \).

To achieve these ROs, a new measure is developed as follows.

**Definition 5** Let \( \tilde{\alpha} = (\alpha_0, \alpha_1, \ldots, \alpha_n) \) and \( \tilde{\beta} = (\beta_0, \beta_1, \ldots, \beta_n) \) be two PFVs, where \( \alpha_j = (\mu_{\alpha}, \eta_{\alpha}, \nu_{\alpha}) \) and \( \beta_j = (\mu_{\beta}, \eta_{\beta}, \nu_{\beta}) \), for all \( j = 1, 2, \ldots, n \), then

\[
N \text{Proj}_{\tilde{\beta}}(\tilde{\alpha}) = \frac{|\tilde{\alpha}|^2 + |\tilde{\beta}|^2}{|\tilde{\alpha}|^2 + |\tilde{\beta}|^2 + |\tilde{\alpha} \cdot \tilde{\beta} - |\tilde{\beta}|^2|}.
\]

is called the normalization projection of vector \( \tilde{\alpha} \) onto \( \tilde{\beta} \), where the \( \tilde{\alpha} \cdot \tilde{\beta} = \sum_{j=1}^{n} (\mu_{\alpha} \cdot \mu_{\beta} + \eta_{\alpha} \cdot \eta_{\beta} + \nu_{\alpha} \cdot \nu_{\beta} + \pi_{\alpha} \cdot \pi_{\beta}) \) is the inner product between \( \tilde{\alpha} \) and \( \tilde{\beta} \), and \( |\tilde{\beta}|^2 = \sum_{j=1}^{n} (\mu_{\beta}^2 + \eta_{\beta}^2 + \nu_{\beta}^2 + \pi_{\beta}^2) \) is the square of module of \( \tilde{\beta} \), with

\[
\pi_{\alpha} = 1 - \mu_{\alpha} - \eta_{\alpha} - \nu_{\alpha}, \pi_{\beta} = 1 - \mu_{\beta} - \eta_{\beta} - \nu_{\beta} (j = 1, 2, \ldots, n).
\]
Criterion 2 The closer the $N\text{Proj}_\tilde{\beta}(\tilde{\alpha})$ is to 1, the closer the vector $\tilde{\alpha}$ is to the $\tilde{\beta}$.

For example, for the PFVs $\tilde{\alpha} = ((0.900, 0.040, 0.005), (0.898, 0.070, 0.002))$ and $\tilde{\beta} = ((0.793, 0.162, 0.002), (0.796, 0.200, 0.003))$ in Example 2, it is easy to calculate that $\|\tilde{\alpha} - \tilde{\beta}\| = 1.4514$, $\|\tilde{\alpha}\|^2 = 1.6269 + 1.3306 = 2.9575$, $\|\tilde{\alpha} - \tilde{\beta}\|^2 = 1.0128$. Thus the projection of vector $\tilde{\alpha}$ onto $\tilde{\beta}$ is $N\text{Proj}_\tilde{\beta}(\tilde{\alpha}) = 2.9575/(2.9575 + 1.0128) = 0.9608$ based on Eq. (5). Since $\tilde{\alpha}_0 = \|\tilde{\alpha}\|^2$ and $\tilde{\beta}_0 = \|\tilde{\beta}\|^2$, the projection of vector $\tilde{\alpha}$ onto oneself is $N\text{Proj}_\tilde{\alpha}(\tilde{\alpha}) = 2.9575/(2.9575 + 0) = 1$. Similarly, $N\text{Proj}_\tilde{\beta}(\tilde{\beta}) = 1$.

Definition 6 Let $X = (x_{kj})_{1 \times n}$ and $Y = (y_{kj})_{1 \times n}$ be two PFMs, where $x_{kj} = (\mu_{kj}, \eta_{kj}, \nu_{kj}), y_{kj} = (\tau_{kj}, \varsigma_{kj}, \upsilon_{kj})(k = 1, 2, \ldots, t; j = 1, 2, \ldots, n)$, then

$$N\text{Proj}_Y(X) = \frac{|X|^2 + |Y|^2}{|X|^2 + |Y|^2 + |XY| - |Y|^2},$$

(6)

is the normalization projection of matrix $X$ onto $Y$, where $|X|^2 = \sum_{k=1}^{t}\sum_{j=1}^{n}(\mu_{kj})^2 + (\eta_{kj})^2 + (\nu_{kj})^2, |Y|^2 = \sum_{k=1}^{t}\sum_{j=1}^{n}(\tau_{kj})^2 + (\varsigma_{kj})^2 + (\upsilon_{kj})^2, XY = \sum_{k=1}^{t}\sum_{j=1}^{n}(\mu_{kj}\tau_{kj} + \eta_{kj}\varsigma_{kj} + \nu_{kj}\upsilon_{kj} + \pi_{kj}\rho_{kj})$, and $\pi_{kj} = 1 - \mu_{kj} - \eta_{kj} - \nu_{kj}; \rho_{kj} = 1 - \tau_{kj} - \varsigma_{kj} - \upsilon_{kj}(k = 1, 2, \ldots, t; j = 1, 2, \ldots, n)$.

For two decision matrices $X$ and $Y$, their closeness is based on the following Criterion 3.

Criterion 3 The closer the $N\text{Proj}_Y(X)$ is to 1, the closer the matrix $X$ is to the $Y$.

5 Research methodology and algorithm

Based on the new projection measure between two PFMs and existing VIKOR methods, a new GDM approach is developed under picture fuzzy environment, which will be applied to the software reliability evaluation.

5.1 Research methodology

In this section, an assessment method based on VIKOR method and the normalization projection measure is elaborated in order to evaluate software reliability.

The software products concerning reliability comprise a set of alternatives, which is written as $A = \{A_i | i \in M\}$, where $M = \{1, 2, \ldots, m\}$; let $U = \{u_j | j \in N\}$ be a set of attributes with $N = \{1, 2, \ldots, n\}$; let $w = (w_1, w_2, \ldots, w_n)$ be a attributes’ weight vector with $0 \leq w_j \leq 1$ and $\sum_{j=1}^{n} w_j = 1$; let $D = \{d_{kj} | k \in T\}$ be a set of decision makers (DMs), where $T = \{1, 2, \ldots, t\}$.

The $i$th software product with respect to $n$ attributes is evaluated by $t$ DMs, which can be expressed by the following group utility matrices:

$$X_i = \begin{pmatrix} u_1 & u_2 & \cdots & u_n \\ d_1 & d_2 & \cdots & d_n \end{pmatrix}, i \in M,$$

(7)

where $X_i = (x_{kj})_{1 \times n}(i \in M)$ are PFMs and $x_{ij} = (\mu_{ij}, \eta_{ij}, \nu_{ij})(i \in M, k \in T, j \in N)$. The element $\mu_{ij}$ expresses the positive (agreeing, approving) degree, $\eta_{ij}$ expresses the neutral degree and $\nu_{ij}$ expresses the negative (disagreeing, disapproving) degree of software product $A_i$ with respect to attribute $u_j$. And $\pi_{ij} = 1 - \mu_{ij} - \eta_{ij} - \nu_{ij}$ expresses the refusal degree.

For the weight vector $w = (w_1, w_2, \ldots, w_n)$ of attributes, the weighted group utility matrices are determined by:

$$Y_i = \begin{pmatrix} u_1 & u_2 & \cdots & u_n \\ d_1 & d_2 & \cdots & d_n \end{pmatrix}, i \in M,$$

(8)

where $Y_i = (y_{ij})_{1 \times n}$ and $y_{ij} = (\tau_{ij}, \varsigma_{ij}, \upsilon_{ij}) = (1 - (1 - \mu_{ij}^w, \eta_{ij}^w, \nu_{ij}^w), (\mu_{ij}^w, \eta_{ij}^w, \nu_{ij}^w) - (\eta_{ij}^w, \eta_{ij}^w))$ by Definition 2.

In VIKOR method, the ideal decision is a reference point. The largest group utility is composed of positive ideal decision as follows:

$$Y_+ = \begin{pmatrix} y_{11}^{+} & y_{12}^{+} & \cdots & y_{1n}^{+} \\ d_1 & d_2 & \cdots & d_n \end{pmatrix},$$

(9)

where $Y_+ = (y_{ij}^{+})_{1 \times n}$ and $y_{ij}^{+} = (\tau_{ij}^{+}, \varsigma_{ij}^{+}, \upsilon_{ij}^{+}) = (\max\{\tau_{ij}\}, \min\{\varsigma_{ij}\}, \min\{\upsilon_{ij}\})$ by Definition 2.

The group regret matrix of alternative $A_i$ should be measured by $Y_+ - Y_i$. Here it is defined as:

$$G_i = \begin{pmatrix} u_1 & u_2 & \cdots & u_n \\ d_1 & d_2 & \cdots & d_n \end{pmatrix}, i \in M,$$

(10)
where \( Y_i = Y_i \otimes Y_+ = (y_{ki}^+)_{i \times n} \otimes (y_{ki}^+)_{i \times n} = (g_{ki}^+)_{i \times n} \)
and \( y_{ki}^+ = (t_{ki}^+, s_{ki}^+, v_{ki}^+), y_{ki}^+ = (r_{ki}^+, s_{ki}^+, v_{ki}^+), g_{ki}^+ = y_{ki}^+ \otimes y_{ki}^+ = (\omega_{ki}^+, \psi_{ki}^+, \chi_{ki}^+) \).

The largest group regret of \( G_i(i \in M) \) is determined by:

\[
G_\omega = \left( \begin{array}{c}
\frac{u_1}{g_{11}^-} & \frac{u_2}{g_{12}^-} & \ldots & \frac{u_n}{g_{1n}^-} \\
\frac{d_1}{g_{21}^-} & \frac{d_2}{g_{22}^-} & \ldots & \frac{d_n}{g_{2n}^-} \\
\vdots & \vdots & \ddots & \vdots \\
\frac{\omega_i}{g_{n1}^-} & \frac{\omega_i}{g_{n2}^-} & \ldots & \frac{\omega_i}{g_{nn}^-}
\end{array} \right),
\]

(11)

where \( G_\omega = (g_{ki}^+)_{i \times n} \) and \( g_{ki}^- = (\omega_{ki}^-, \psi_{ki}^-, \chi_{ki}^-) = \max_{1 \leq i \leq m} (g_{ki}^+) = (\max_{1 \leq i \leq m} \{\omega_{ki}^+, \psi_{ki}^+, \chi_{ki}^+ \}, \min_{1 \leq i \leq m} \{\psi_{ki}^+, \chi_{ki}^+ \}). \)

To calculate the closeness between \( Y_i \) and \( Y_+ \), a projection of \( Y_i \) onto \( Y_+ \) is based on Eq. (6) as follows:

\[
N_{\text{Proj}+}(Y_i) = \frac{|Y_i|^2 + |Y_+|^2}{|Y_i|^2 + |Y_+|^2 + |Y_i - Y_+|^2}, i \in M,
\]

(12)

where \( Y_+ = \sum_{k=1}^{l} \sum_{j=1}^{n} (t_{kj}^+ s_{kj}^+ + v_{kj}^+)(i \in M), |Y_i|^2 = \sum_{k=1}^{l} \sum_{j=1}^{n} (t_{kj}^+)^2 + (s_{kj}^+)^2 + (v_{kj}^+)^2, \pi_{kj}^+ = 1 - t_{kj}^+ - s_{kj}^+ - v_{kj}^+ |Y_+|^2 = \sum_{k=1}^{l} \sum_{j=1}^{n} (t_{kj}^+)^2 + (s_{kj}^+)^2 + (v_{kj}^+)^2, \pi_{kj}^+ = 1 - t_{kj}^+ - s_{kj}^+ - v_{kj}^+ (k \in T, j \in N, i \in M). \)

For the closeness between \( Y_i \) and \( Y_+ \), based on Eq. (12), similar to Criterion 3, we have the following Criterion 4.

**Criterion 4** The closer the \( N_{\text{Proj}+}(Y_i) \) is to 1, the closer the \( Y_i \) is to \( Y_+ \), the better the alternative \( A_i \) is.

For measuring the closeness between \( G_i \) onto \( G_- \), similar to (12), we have the following equation:

\[
N_{\text{Proj}G_-}(G_i) = \frac{|G_i|^2 + |G_-|^2}{|G_i|^2 + |G_-|^2 + |G_i G_- - |G_-|^2|}, i \in M,
\]

(13)

where \( |G_i|^2 = \sum_{k=1}^{l} \sum_{j=1}^{n} (\omega_{kj}^2 + (\psi_{kj}^2 + (\chi_{kj}^2)^2)) \), \( |G_-|^2 = \sum_{k=1}^{l} \sum_{j=1}^{n} (\omega_{kj}^2 + (\psi_{kj}^2 + (\chi_{kj}^2)^2)) \), \( G_i G_- = \sum_{k=1}^{l} \sum_{j=1}^{n} (\omega_{kj}^2) \).

Similar to Criterion 4, we have the following Criterion 5.

**Criterion 5** The closer the \( N_{\text{Proj}G_-}(G_i) \) is to 1, the closer the matrix \( G_i \) is to the \( G_- \), the larger the group regret with regard to alternative \( A_i \) is, the worse the alternative \( A_i \) is.

The group regret measure of alternative \( A_i \) is based on the following equation:

\[
R_i = N_{\text{Proj}G_-}(G_i), i \in M,
\]

(16)

where \( N_{\text{Proj}G_-}(G_i)(i \in M) \) is same as in Eq. (13).

Equation (16) satisfies the condition \( 0 \leq R_i \leq 1 \), and we have the following Criterion 8.

**Criterion 8** The closer the \( R_i \) is to 1, the larger the group regret with regard to alternative \( A_i \) is, the worse the alternative \( A_i \) is.

Similar to Eq. (15), if we let \( R_+ = \min_{1 \leq i \leq m} \{R_i\}, R_- = \max_{1 \leq i \leq m} \{R_i\}. \) The \( R_+ \) is called as the smallest group regret, the \( R_- \) is called as the largest group regret. The normalized group regret of \( A_i \) is defined as:

\[
NGR_i = \begin{cases} 
\frac{R_+ - R_i}{R_- - R_+}, & \text{if } R_+ \neq R_-; i \in M, \\
0, & \text{if } R_+ = R_-.
\end{cases}
\]

(17)
where, in order to avoid the case that denominator $R_+ - R_-$ is zero, we let $NGR_i = 0$ when $R_+ = R_-$. 

**Criterion 9** The closer the $NGR_i$ is to 1, the better the alternative $A_i$ is.

Thus, a comprehensive VIKOR measure of alternative $A_i$ can be obtained by the following relation:

$$Q_i = \lambda NGU_i + (1 - \lambda) NGR_i, i \in M,$$  \hspace{1cm} (18)

where the $\lambda$ is referred to as a compromise coefficient, and $\lambda \in [0, 1]$. The values of $\lambda$ and $1 - \lambda$ are the weight of the normalized group utility $NGU_i$ and the normalized group regret $NGR_i$, respectively. If $\lambda > 0.5$, it is indicated that DMs tend to make decisions based on the group utility; if $\lambda < 0.5$, it is indicated that DMs tend to make decisions based on the group regret; if $\lambda = 0.5$, it is indicated that DMs adopt a balanced and compromised way to make decisions. In general, the value $\nu = 0.5$ is adopted.

The grades of alternatives are affected by VIKOR indexes $Q_i, S_i$ and $R_i$ collectively. The relative importance of three VIKOR indexes is that $Q_i > S_i$ and $Q_i > R_i$, where the “>” indicates “superior to”.

Combining the Criteria 7, 8, 9, we have the following Criterion 10.

**Criterion 10** The larger the value $Q_i$, the better the alternative $A_i$ is.

Let $Q(A_i^{(h)})$ denote that the alternative $A_i$ is ranked in $h$th position by $Q$. A compromise solution should satisfy the following two conditions:

**Condition 1.** Acceptable advantage: If $Q(A_i^{(1)}) - Q(A_i^{(2)}) \geq 1/(m-1)$, where the $m$ is the number of alternatives, then the alternative $A_i$ is regarded as a compromise solution.

**Note:** The $Q$ in Eq. (18) is compromised by $NGU$ and $NGR$. Hence, the compromise solution should be consistent with $NGU$ and/or $NGR$. However, the solution generated by $Q$ may be inconsistent with the solutions generated by $S$ and/or $R$ (see an example in Table 4). So a supplementary condition is introduced as follows.

**Condition 2.** Acceptable stability in decision making: The alternative $A_i$ must also be ranked first by $S$ or/and $R$.

If the alternative $A_i$ satisfies the only one condition, then the compromise solutions comprise a set based on the following rules (Opricovic and Tzeng 2007; Ren et al. 2017):

- The compromise solutions are comprised of the alternatives $\{A_{i1}, A_{i2}\}$ if only Condition 2 is not satisfied.
- The compromise solutions are comprised of the alternatives $\{A_{i1}, A_{i2}, \ldots, A_{im}\}$ if Condition 1 is not satisfied. The $A_{im}$ is determined by the relation $Q(A_i^{(1)}) - Q(A_i^{(M)}) < 1/(m - 1)$ for maximum $M$ (the positions of these alternatives are "in closeness").

### 5.2 Assessment algorithm

The assessment algorithm of software reliability is based on above-mentioned VIKOR method and normalized projection measure with picture fuzzy information, which are involved into the following steps. The algorithmic pseudocode used to describe algorithms in this text is also attached in detail. The pseudocode can help us to understand the algorithms easily. It can serve as an intermediate step in the construction of programs implementing algorithms in one of a variety of different programming languages.

**Step 1.** Establish the group utility matrices.

The group utility matrix $X_i$ of alternative $A_i$ is established by Eq. (5), where the $X_i (i \in M)$ are PFMs. A pseudocode for Step 1 can be described as follows:

**Function:** Establish the set of group utility matrices $X_1, X_2, \ldots, X_m$;

Input: $[x_{11}^i], [x_{12}^i], \ldots, [x_{nm}^i] \ (i = 1, 2, \ldots, m)$ ($t \times n$ matrices);

Output: $X_1, X_2, \ldots, X_m$;

for $i := 1$ to $m$

for $k := 1$ to $t$

for $j := 1$ to $n$

$x_{kj}^i := (\mu_{kj}^i, \eta_{kj}^i, \upsilon_{kj}^i) \ \{x_{kj}^i \text{ are PFNs}\}$;

$X_i := \{x_{kj}^i\}$;

end

end

return: $X_i \ \{X_i = \{x_{kj}^i\} (i = 1, 2, \ldots, m) \text{ are } t \times n \text{ matrices}\}$.

**Step 2.** Construct the weighted group utility matrices. For a given weight vector $w = (w_1, w_2, \ldots, w_n)$ of attributes, the weighted group utility matrices $Y_i (i \in M)$ are constructed by Eq. (8).

The algorithmic in Step 2 can be described using an easily understood form of pseudocode as follows:

**Function:** Construct the set of weighted utility matrices $Y_1, Y_2, \ldots, Y_m$;

Input: $X_1, X_2, \ldots, X_m; w_1, w_2, \ldots, w_n$ (with $0 \leq w_j \leq 1$ and $\\sum_{j=1}^{n} w_j = 1$);

Output: $Y_1, Y_2, \ldots, Y_m$;

for $i := 1$ to $m$

for $k := 1$ to $t$

for $j := 1$ to $n$

$t_{kj}^i := 1 - (1 - \mu_{kj}^i)^{w_j}$;

$s_{kj}^i := (\eta_{kj}^i)^{w_j}$;

$\upsilon_{kj}^i := (\eta_{kj}^i + \upsilon_{kj}^i)^{w_j}$;

$y_{kj}^i := (t_{kj}^i, s_{kj}^i, \upsilon_{kj}^i)$;
The group regret matrices can be implemented by the function $G_i$.

**Step 3.** Determine the largest group utility.

The largest group utility $Y_i$ is determined by Eq. (9).

A pseudocode in Step 3 is given as follows:

```plaintext
Function: Calculate the largest group utility $Y_i$;
Input: $Y_1, Y_2, \cdots, Y_m$;
Output: $Y_i$;
for $i := 1$ to $m$
    for $k := 1$ to $t$
        for $j := 1$ to $n$
            $\tau_{kj} := \max\{\tau_{kj}\}$;
            $s_{kj} := \min\{s_{kj}\}$;
            $v_{kj} := \min\{v_{kj}\}$;
            $y_{kj} := (\tau_{kj}, s_{kj}, v_{kj})$;
        end
        $Y_i := [y_{kj}]$;
    end
end
return: $Y_i$ \{ $Y_i = [y_{kj}]$ is $t \times n$ matrix \}.
```

**Step 4.** Establish the group regret matrices.

The group regret matrices $G_i(i \in M)$ based on $Y_i$ and $Y_1(i \in M)$ are established by Eq. (10).

The group regret matrices can be implemented by the following pseudocode:

```plaintext
Function: Establish the group regret matrices $G_1, G_2, \cdots, G_m$;
Input: $Y_1, Y_2, \cdots, Y_m, Y_1$;
Output: $G_1, G_2, \cdots, G_m$;
for $i := 1$ to $m$
    for $k := 1$ to $t$
        for $j := 1$ to $n$
            $\omega_{kj} := (\tau_{kj} + s_{kj} + v_{kj}) - s_{kj} - s_{kj}$;
            $\psi_{kj} := s_{kj} + v_{kj}$;
            $\chi_{kj} := 1 - (1 - v_{kj})(1 - v_{kj})$;
            $g_{kj} := (\omega_{kj}, \psi_{kj}, \chi_{kj})$;
        end
        $G_i := [g_{kj}]$;
    end
end
return: $G_i$ \{ $G_i = [g_{kj}]$ are $t \times n$ matrices \}.
```

**Step 5.** Determine the largest group regret matrix.

The largest group regret matrix is determined by Eq. (11). A simple pseudocode example is provided as follows:

```plaintext
Function: Determine the largest group regret matrix $G_-$;
Input: $G_1, G_2, \cdots, G_m$;
Output: $G_-$;
for $k := 1$ to $t$
    for $i := 1$ to $m$
        $\omega_{kj} := \max\{\omega_{kj}\}$;
        $\psi_{kj} := \min\{\psi_{kj}\}$;
        $\chi_{kj} := \min\{\chi_{kj}\}$;
        $g_{kj} := (\omega_{kj}, \psi_{kj}, \chi_{kj})$;
    end
end
return: $G_-$ \{ $G_- = [g_{kj}]$ is $t \times n$ matrix \}.
```

**Step 6.** Calculate the closeness between $Y_i$ and $Y_1$.

The normalization projection of $Y_i(i \in M)$ onto $Y_1$ are determined by Eq. (12).

A programming pseudocode is shown as follows:

```plaintext
Function: Calculate the normalized projections $N Proj_{Y_1}(Y_i)$, for $i = 1, 2, \cdots, m$;
Input: $Y_i, Y_1$, for $i = 1, 2, \cdots, m$;
Output: $N Proj_{Y_1}(Y_i)$, for $i = 1, 2, \cdots, m$;
for $i := 1$ to $m$
    for $k := 1$ to $t$
        for $j := 1$ to $n$
            $\pi_{kj} := 1 - \tau_{kj} - s_{kj} - v_{kj}$;
            $\pi_{ij} := 1 - \tau_{ij} - s_{ij} - v_{ij}$;
            $Y_{ij} := \sum_{k=1}^{n} \sum_{j=1}^{n} ((\tau_{kj})^2 + (s_{kj})^2 + (v_{kj})^2) + (\pi_{kj})^2$;
            $Y_{ij} := \sum_{k=1}^{n} \sum_{j=1}^{n} ((\tau_{ij})^2 + (s_{ij})^2 + (v_{ij})^2) + (\pi_{ij})^2$;
        end
        $N Proj_{Y_1}(Y_i) := ((Y_{ij})^2 + (Y_{ij})^2)/(|Y_{ij}|^2 + |Y_{ij}|^2 + |Y_{ij}|^2)$;
    end
end
return: $N Proj_{Y_1}(Y_i)$ \{ $N Proj_{Y_1}(Y_i)$ satisfies $0 \leq N Proj_{Y_1}(Y_i) \leq 1$ \}.
```

**Step 7.** Measure the closeness between each group regret matrix and the largest group regret matrix based on the normalized projection.
The normalized projection of each group regret matrix
\( G_i (i \in M) \) onto the largest group regret matrix \( G_- \) are
determined by Eq. (13).

A programming pseudocode is shown as follows:

Function: Calculate the normalized projections
\( NP_{ro jG_+} (G_i) \), for \( i = 1, 2, \ldots, m; \)
Input: \( G_i, G_- \), for \( i = 1, 2, \ldots, m; \)
Output: \( NP_{ro jG_+}(G_i) \), for \( i = 1, 2, \ldots, m; \)

for \( i := 1 \) to \( m \)
for \( k := 1 \) to \( t \)
\[
\sigma_{ij}^k := 1 - \omega_{ik}^o - \psi_{ij}^k \chi^k_{ij}; \\
\sigma_{ij}^- := 1 - \omega_{ik}^- - \psi_{ij}^- \chi^k_{ij}; \\
G_i G_- := \sum_{i=1}^n \sum_{j=1}^m (\omega_{ik}^o \omega_{kj}^- + \psi_{ij}^k \psi_{kj}^- + \\
\chi^k_{ij} \chi^k_{kj} + \sigma_{ij}^k \sigma_{ij}^-); \\
|G_i|_2 := \sum_{i=1}^n (\omega_{ik}^o)^2 + (\psi_{ij}^k)^2 + \\
(\chi^k_{ij})^2 + (\sigma_{ij}^k)^2; \\
|G_-|_2 := \sum_{i=1}^n (\omega_{ik}^-)^2 + (\psi_{kj}^-)^2 + \\
(\chi^k_{kj})^2 + (\sigma_{ij}^-)^2; \\
\]
end

\( NP_{ro jG_+}(G_i) := |G_i|_2 + |G_-|_2 / (|G_i|_2 + |G_-|_2 + \\
|G_i G_-|_2); \)
end
return: \( NP_{ro jG_+}(G_i) \) \{ \( NP_{ro jG_+}(G_i) \) satisfies \( 0 \leq NP_{ro jG_+}(G_i) \leq 1 \).\}

Step 8. Construct the group utility measurement.
The group utility measure \( S_i \) of alternative \( A_i \) is determined by Eq. (14).
The VIKOR index \( S_i \) can be implemented by the following pseudocode:

Function: Calculate the VIKOR index \( S_i \), for \( i = 1, 2, \ldots, m; \)
Input: \( NP_{ro jY_i}(Y_i) \), for \( i = 1, 2, \ldots, m; \)
Output: \( S_i \), for \( i = 1, 2, \ldots, m; \)

for \( i := 1 \) to \( m \)
\( S_i := NP_{ro jY_i}(Y_i); \)
end
return: \( S_i \) \{ \( S_i \) \( (i = 1, 2, \ldots, m) \) satisfies \( 0 \leq S_i \leq 1 \).\}

Step 9. Determine the normalized group utilities.
The normalized group utilities are determined by Eq. (15).
The normalized group utility can be implemented by the following pseudocode:

Function: Determine the normalized group utility \( NGU_i \),
for \( i = 1, 2, \ldots, m; \)

for \( i := 1 \) to \( m \)
\( R_+ := \min \{ R_i \}; \)
\( R_- := \max \{ R_i \}; \)
if \( R_+ \neq R_- \), then \( NGR_i := (R_- - R_i) / (R_- - \) \\
\( R_+); \) \else \( NGR_i := 0; \)
end
return: \( NGR_i \) \{ \( NGR_i \) \( (i = 1, 2, \ldots, m) \) satisfies \( 0 \leq NGR_i \leq 1 \).\}

Step 10. Construct the group regret measurement.
The group regret measurement of alternative \( A_i \) is established by Eq. (16).
The group regret measurement of alternative \( A_i \) can be expressed by the following pseudocode:

Function: Construct the group regret measurement \( R_i \),
for \( i = 1, 2, \ldots, m; \)
Input: \( NP_{ro jY_\cdot}(G_i) \), for \( i = 1, 2, \ldots, m; \)
Output: \( R_i \), for \( i = 1, 2, \ldots, m; \)

for \( i := 1 \) to \( m \)
\( R_i := NP_{ro jY_\cdot}(G_i); \)
end
return: \( R_i \) \{ \( R_i \) \( (i = 1, 2, \ldots, m) \) satisfies \( 0 \leq R_i \leq 1 \).\}

Step 11. Construct the normalized group regret measurement.
The normalized group regret measurements are constructed by Eq. (17).
The algorithms used in Step 11 can be implemented by the following pseudocode:

Function: Determine the normalized group regret \( NGR_i \),
for \( i = 1, 2, \ldots, m; \)
Input: \( R_i \), for \( i = 1, 2, \ldots, m; \)
Output: \( NGR_i \), for \( i = 1, 2, \ldots, m; \)

for \( i := 1 \) to \( m \)
\( R_+ := \min \{ R_i \}; \)
\( R_- := \max \{ R_i \}; \)
if \( R_+ \neq R_- \), then \( NGR_i := (R_- - R_i) / (R_- - \) \\
\( R_+); \) \else \( NGR_i := 0; \)
end
return: \( NGR_i \) \{ \( NGR_i \) \( (i = 1, 2, \ldots, m) \) satisfies \( 0 \leq NGR_i \leq 1 \).\}

Step 12. Construct the comprehensive VIKOR measurement.
The comprehensive VIKOR measure \( Q_i \) of alternative \( A_i \) is calculated by Eq. (18). The comprehensive VIKOR measure \( Q_i \) can be expressed in an easily understood form of structured pseudocode:

**Function:** Construct the comprehensive VIKOR measure \( Q_i \), for \( i = 1, 2, \ldots, m \);

**Input:** \( NGU_i, NGR_i \), for \( i = 1, 2, \ldots, m \);

**Output:** \( Q_i \), for \( i = 1, 2, \ldots, m \);

for \( i := 1 \) to \( m \)

\[
Q_i := \lambda NGU_i + (1 - \lambda) NGR_i \quad \{ \lambda \text{ satisfies } 0 \leq \lambda \leq 1 \};
\]

end

return: \( Q_i \) (\( i = 1, 2, \ldots, m \)) satisfies \( 0 \leq Q_i \leq 1 \).

**Step 13.** Rank the preference order of alternatives. The alternatives are ranked in descending order in accordance with the VIKOR measures \( S_i \), \( R_i \) and \( Q_i \). The compromise solutions are based on above Condition 1 and Condition 2 with two rules.

The preference order of alternative \( A_i \) can be based on the following pseudocode:

**Function:** Rank the preference order of alternative \( A_i \), for \( i = 1, 2, \ldots, m \);

**Input:** \( Q(A_i^{(1)}), Q(A_i^{(2)}), \ldots, Q(A_i^{(m)}) \); \( S(A_i^{(1)}), R(A_i^{(1)}) \) \( Q(A_h^{(b)}) \) denotes the software \( A_h \) is ranked in \( h \)th position in the ranking list by \( Q \); \( S(A_i^{(1)}) \) denotes the software \( A_i \) is ranked in the first position in the ranking list by \( S \); \( R(A_i^{(1)}) \) denotes the software \( A_i \) is ranked in the first position in the ranking list by \( R \);

**Output:** The preference order of alternative \( A_i \), for \( i = 1, 2, \ldots, m \);

for \( h := 1 \) to \( m - 1 \)

while \( Q(A_h^{(b)}) - Q(A_h^{(b+1)}) > 1/(m - 1) \) (\( m \) is the number of software products), and \( S(A_i^{(1)}) \) or \( R(A_i^{(1)}) \), write \( A_i > A_j \) if \( Q(A_i^{(b)}) - Q(A_i^{(b+1)}) < 1/(m - 1) \), and \( Q(A_i^{(b)}) - Q(A_i^{(b+1)}) \geq 1/(m - 1) \), then \( \{ A_{h1}, A_{h+1}, \ldots, A_{hk} \} \) is tied for the \( h \)th position in the ranking list;

end

return: \( \{ A_{i1}, A_{i2}, \ldots, A_{ik} \} > \{ A_{h1}, A_{h+1}, A_{h+2}, \ldots, A_{ij} \} > \ldots > \{ A_{ig}, A_{g+1}, \ldots, A_{im} \} (g > i) \) is in decreasing order of software products \( A_1, A_2, \ldots, A_m \) with classification.

### 6 Experimental analysis

A software reliability assessment is shown in the following illustrative example section. The superiorities developed methodology in this paper are shown through some experimental analyses.

#### 6.1 Illustrative example

This section provides a real evaluation of software reliability. Four software products are evaluated here, which are used in a university, Guangdong, China. For convenience, the evaluated software as alternatives comprise a set denoted by \( A = \{ A_1, A_2, A_3, A_4 \} \).

The DMs are users, who are from three different colleges in this university. They are written as \( D = \{ d_1, d_2, d_3 \} \), where each \( d_k (k = 1, 2, 3) \) is a group of users from a college. More specifically, \( d_1 \) is the users from college of mathematics and computer science; \( d_2 \) is the users from college of mechanical engineering; \( d_3 \) is the users from college of ocean and engineering.

The assessment criteria/attributes are determined by DMs collectively. Four assessment attributes comprise a set denoted by \( Y = \{ u_{11}, u_{12}, u_{13}, u_{4} \} = \{ \text{program complexity, program categories, programming language, design methodologies} \} \), where the program size is used as a measure of program complexity; the program categories indicate system complexity; different programming languages have different complexity and structure, therefore the possibility for different languages to introduce errors are different; different design methodologies for the same software may have different impact on the quality of the final software products (Zhu and Pham 2017). Each software is evaluated by three DMs, whose evaluation values are integrally arranged in an evaluation matrix. By Step 1, four evaluation matrices are written as \( \{ X_1, X_2, X_3, X_4 \} \), which are shown in Table 1.

In Table 1, the assessment values \( x_{kj}^i \) in \( X_i \) are characterized by \( (\mu_{ij}^k, \eta_{ij}^k, \nu_{ij}^k)(i, j = 1, 2, 3, 4; k = 1, 2, 3) \), where the \( \mu_{ij}^k \) is the ratio that the voters vote for \( i \)th software with respect to \( j \)th attribute in \( k \)th college; \( \eta_{ij}^k \) is the ratio that the voters abstain on \( i \)th software with respect to \( j \)th attribute in \( k \)th college; \( \nu_{ij}^k \) is the ratio that the voters are against on \( i \)th software with respect to \( j \)th attribute in \( k \)th college.

Through negotiation, the weights of attributes are given by DMs, the weight vector is \( w = (w_1, w_2, w_3, w_4) = (0.3, 0.3, 0.2, 0.2) \). By Step 2, the weighted group decisions \( Y_1, Y_2, Y_3, Y_4 \) are shown in Table 2.

The largest group utility \( Y_+ \) can be obtained by Step 3, which is also shown in Table 2.

Then the group regret matrices \( G_i (i \in M) \) based on \( Y_i (i \in M) \) and \( Y_+ \) are obtained by Step 4; the largest group regret
matrix $G_-$ is obtained by Step 5. The group regret matrices and the largest group regret matrix are shown in Table 3.

To measure the closeness of each $Y_i$ to the largest group utility $Y_+$, the normalized projections $NProj_{Y_+}(Y_i)$ ($i \in M$) are calculated by Step 6.

To measure the closeness between each $G_i$ and the largest group regret $G_-$, the normalized projections $NProj_{G_-}(G_i)$ ($i \in M$) are calculated by Step 7, which are shown in Table 4. The group utility measures $S_i$ ($i \in M$) are obtained by Step 8, which are also shown in Table 4. The group regret index $R_i$ ($i \in M$) are obtained from $NProj_{G_-}(G_i)$, which are also shown in Table 4.

From Table 4 we can see that the largest group utility $S_+$ is 0.9965; the smallest group utility $S_-$ is 0.9717; the largest group regret $R_+$ is 0.7749; the smallest group regret $R_-$ is 0.7544. Therefore we have comprehensive VIKOR measurements $Q_i$ ($i \in M$) based on Step 12, which are also shown in Table 4. And the rankings of four products based on $S_i$, $R_i$, $Q_i$, respectively, are summarized in Table 4.

Table 1 Four assessment matrices

| Matrix | DM | $u_1$ | $u_2$ | $u_3$ | $u_4$ |
|--------|----|-------|-------|-------|-------|
| $X_1$  | $d_1$ | (0.49, 0.21, 0.20) | (0.53, 0.11, 0.29) | (0.45, 0.21, 0.31) | (0.52, 0.30, 0.13) |
| $d_2$  | (0.47, 0.16, 0.23) | (0.43, 0.22, 0.20) | (0.35, 0.23, 0.35) | (0.50, 0.27, 0.19) |
| $d_3$  | (0.43, 0.19, 0.19) | (0.58, 0.08, 0.17) | (0.46, 0.17, 0.26) | (0.55, 0.22, 0.11) |
| $X_2$  | $d_1$ | (0.53, 0.23, 0.12) | (0.49, 0.18, 0.20) | (0.51, 0.22, 0.22) | (0.49, 0.16, 0.11) |
| $d_2$  | (0.36, 0.15, 0.23) | (0.45, 0.11, 0.18) | (0.37, 0.02, 0.36) | (0.52, 0.10, 0.09) |
| $d_3$  | (0.49, 0.18, 0.20) | (0.38, 0.07, 0.16) | (0.56, 0.14, 0.25) | (0.48, 0.11, 0.13) |
| $X_3$  | $d_1$ | (0.55, 0.22, 0.11) | (0.51, 0.17, 0.19) | (0.51, 0.21, 0.21) | (0.50, 0.15, 0.10) |
| $d_2$  | (0.37, 0.14, 0.22) | (0.46, 0.11, 0.17) | (0.38, 0.12, 0.35) | (0.53, 0.11, 0.08) |
| $d_3$  | (0.50, 0.17, 0.19) | (0.39, 0.06, 0.15) | (0.57, 0.13, 0.23) | (0.49, 0.10, 0.12) |
| $X_4$  | $d_1$ | (0.50, 0.20, 0.20) | (0.55, 0.10, 0.30) | (0.46, 0.20, 0.32) | (0.49, 0.30, 0.12) |
| $d_2$  | (0.48, 0.15, 0.23) | (0.43, 0.21, 0.19) | (0.35, 0.22, 0.36) | (0.51, 0.26, 0.19) |
| $d_3$  | (0.44, 0.18, 0.20) | (0.59, 0.07, 0.16) | (0.48, 0.16, 0.25) | (0.56, 0.21, 0.10) |

Table 2 Four weighted assessment matrices and the largest group utility matrix

| Matrix | DM | $u_1$ | $u_2$ | $u_3$ | $u_4$ |
|--------|----|-------|-------|-------|-------|
| $Y_1$  | $d_1$ | (0.18, 0.63, 0.14) | (0.20, 0.52, 0.24) | (0.11, 0.73, 0.15) | (0.14, 0.79, 0.06) |
| $d_2$  | (0.17, 0.58, 0.18) | (0.16, 0.63, 0.14) | (0.08, 0.75, 0.15) | (0.13, 0.77, 0.09) |
| $d_3$  | (0.16, 0.61, 0.14) | (0.23, 0.47, 0.19) | (0.12, 0.70, 0.14) | (0.15, 0.74, 0.06) |
| $Y_2$  | $d_1$ | (0.20, 0.64, 0.09) | (0.18, 0.60, 0.15) | (0.13, 0.74, 0.11) | (0.13, 0.69, 0.08) |
| $d_2$  | (0.13, 0.57, 0.18) | (0.16, 0.52, 0.17) | (0.09, 0.46, 0.37) | (0.14, 0.63, 0.09) |
| $d_3$  | (0.18, 0.60, 0.15) | (0.13, 0.45, 0.19) | (0.15, 0.67, 0.15) | (0.12, 0.64, 0.11) |
| $Y_3$  | $d_1$ | (0.21, 0.63, 0.08) | (0.19, 0.59, 0.15) | (0.13, 0.73, 0.11) | (0.13, 0.68, 0.07) |
| $d_2$  | (0.13, 0.55, 0.18) | (0.17, 0.52, 0.17) | (0.09, 0.65, 0.21) | (0.14, 0.64, 0.07) |
| $d_3$  | (0.19, 0.59, 0.15) | (0.14, 0.43, 0.20) | (0.16, 0.66, 0.15) | (0.13, 0.63, 0.11) |
| $Y_4$  | $d_1$ | (0.19, 0.62, 0.14) | (0.21, 0.50, 0.26) | (0.12, 0.72, 0.15) | (0.14, 0.79, 0.05) |
| $d_2$  | (0.18, 0.57, 0.18) | (0.16, 0.63, 0.13) | (0.08, 0.74, 0.16) | (0.13, 0.76, 0.09) |
| $d_3$  | (0.16, 0.60, 0.15) | (0.23, 0.45, 0.19) | (0.12, 0.69, 0.14) | (0.15, 0.73, 0.06) |
| $Y_+$  | $d_1$ | (0.21, 0.62, 0.08) | (0.21, 0.50, 0.15) | (0.13, 0.72, 0.11) | (0.14, 0.68, 0.05) |
| $d_2$  | (0.18, 0.55, 0.18) | (0.17, 0.52, 0.13) | (0.09, 0.46, 0.15) | (0.14, 0.63, 0.07) |
| $d_3$  | (0.19, 0.59, 0.14) | (0.23, 0.43, 0.19) | (0.16, 0.66, 0.14) | (0.15, 0.63, 0.06) |
6.2 Comparison with a VIKOR-based GDM method

This section provides a comparison with a VIKOR-based GDM method (Yue 2020b). The data are based on the same illustrative example in Sect. 6.1.

According to the idea in (Yue 2020b), a negative ideal decision from $Y_i$ in (8) is shown as follows:

$$Y_- = \left( y_{11}, y_{12}, \ldots, y_{1n} \right),$$  \hspace{1cm} (19)

where $Y_- = (y_{ij})_{1 \times n}$ and $y_{ij}^{-} = (\tau_{kj}^{-}, \varsigma_{kj}^{-}, \nu_{kj}^{-}) = (\min \{v_{ij}^{t} \}, \max \{\varsigma_{ij}^{t} \}, \max \{\nu_{ij}^{t} \})(k \in T, j \in N)$ depended on Definition 2.

Similar to (12), the normalization projection of $Y_i$ onto $Y_-$, based on Eq. (6), are shown in the following equation:

$$N \text{ Proj}_Y(Y_i) \quad = \quad \frac{\left| Y_i \right|^2 + \left| Y_- \right|^2}{\left| Y_i \right|^2 + \left| Y_- \right|^2 + \left| Y_i Y_- - |Y_-|^2 \right|}, \ i \in M,$$ \hspace{1cm} (20)

where $|Y_i|^2 = \sum_{i=1}^{n} \sum_{j=1}^{n} (\tau_{ij}^{*} + (\varsigma_{ij}^{*})^2 + (\nu_{ij}^{*})^2), |Y_-|^2 = \sum_{i=1}^{n} \sum_{j=1}^{n} (\tau_{ij}^{-} + (\varsigma_{ij}^{-})^2 + (\nu_{ij}^{-})^2 + (\nu_{ij}^{-})^2), Y_i Y_- = \sum_{k=1}^{n} \sum_{l=1}^{n} (\tau_{ij}^{t} \tau_{kl}^{-} + \varsigma_{ij}^{t} \varsigma_{kl}^{-} + \nu_{ij}^{t} \nu_{kl}^{-} + \nu_{ij}^{t} \nu_{kl}^{-} + \nu_{ij}^{t} \nu_{kl}^{-})(k \in M, i \in M), \ \pi_{kj} = 1 - \tau_{kj}^{-} - \varsigma_{kj}^{-} - \nu_{kj}^{-} (k \in T, j \in N, i \in M).

$N \text{ Proj}_Y(Y_i)$ is also measure for ranking alternatives, which is shown in the following criterion.

**Criterion 11** The smaller value $N \text{ Proj}_Y(Y_i)$ means the better alternative $A_i$ (Yue 2020).

A group utility measure of alternative $A_i$ is provided by Yue (2020b) as follows:

$$S_i = \frac{N \text{ Proj}_Y(Y_i)}{N \text{ Proj}_Y(Y_i) + N \text{ Proj}_Y(Y_i)} \quad i \in M. \quad (21)$$

**Criterion 12** The larger $S_i$ means the better alternative $A_i$ (Yue 2020b).

A group regret measure of alternative $A_i$ is provided by Yue (2020b) as follows:

$$R_i = 1 - S_i, \ i \in M. \quad (22)$$

where $S_i = N \text{ Proj}_Y(Y_i)(i \in M)$ are same as in Eq. (14).
If the group utility measure is replaced by Eq. (21), the group regret measure is replaced by Eq. (22), then the comprehensive VIKOR measure of software $A_i$ is shown as follows:

$$Q^2_i = \lambda NGU^2_i + (1 - \lambda) NGR^2_i, \quad i \in M,$$

\(23\)

where the $NGU^2_i = (S^2_i - S^2_{i+1})/(S^2_i - S^2_j), S^2_i = \max \{S^2_i\}, S^2_j = \min \{S^2_i\}, NGR^2_i = (R^2_i - R^2_{i+1})/(R^2_i - R^2_j), R^2_i = \min \{R^2_i\}, R^2_j = \max \{R^2_i\}$; The $S^2_i$ is the same as in Eq. (21); $R^2_i$ is the same as in Eq. (22). $\lambda$ is a compromise coefficient, and $\lambda \in [0, 1]$.

The VIKOR indexes $S^2_i, NGU^2_i, NGR^2_i$ and rankings of four software products $A_i$ are shown in Table 5. Let $\lambda = 0.5$. The VIKOR comprehensive index $Q^2_i$ and the ranking are also shown in Table 5.

Table 5 shows that $Q^2(A^{(1)}_1) = 1$, $Q^2(A^{(2)}_1) = 0.9145$, $Q^2(A^{(3)}_1) = 0.1238$, $Q^2(A^{(4)}_1) = 0$. We verify the difference $Q^2(A^{(2)}_1) - Q^2(A^{(3)}_1) = 1 - 0.9145 = 0.0855 < 0.3333$, and $S^2(A^{(2)}_1) = 1$, so the $A_2$ and $A_3$ should be in the same classification based on the measure $Q^2$. We further verify the difference $Q^2(A^{(3)}_1) - Q^2(A^{(4)}_1) = 0.1238 < 0.3333$. So we have $\{A_2, A_3\} \succ A_4$ based on the measure $Q^2$. Continuing to compare the $A_1$, $A_4$, from the difference $Q^2(A^{(4)}_1) - Q^2(A^{(4)}_1) = 0.1238 < 0.3333$, we know that $A_1$ and $A_4$ should be in the same classification. So we have $\{A_2, A_3\} \succ \{A_1, A_4\}$ based on the measure $Q^2$. This ranking is different than the ranking $\{A_1, A_2, A_3\} \succ A_4$ with classification based on the measure $Q$ in Table 4. The primary causes of difference are that there is not the specific regret information in the model (Yue 2020b). The group regret of software $A_i$ is measured directly by the complement of group utility $1 - S_i$. This is an imperfection.

### 6.3 Rankings based on different measures

In above experimental analysis, the measures $S, NGU, R, NGR$ are used in current model. If each measure is used separately to rank the software $A_i$, similar to the ranking of comprehensive VIKOR measure $Q$, we provide the following criterion.

**Criterion 13** Let $Mea(A^{(h)}_{ih})$ denote that the alternative $A_{ih}$ is ranked in $h$th position by measure $Mea \in \{S, NGU, R, NGR\}$. Beginning by $A_{i1}$, the software products $\{A_{i1}, A_{i2}, \cdots, A_{im}\}$ are ranked with classification in phases.

**Phase 1.** If $Mea(A^{(1)}_i) - Mea(A^{(h)}_{ih}) < 1/(m - 1)$ and $Mea(A^{(2)}_i) - Mea(A^{(h+1)}_{ih+1}) \geq 1/(m - 1)$, where $h \geq 2$, then $\{A_{i1}, A_{i2}, \cdots, A_{ih}\}$ are tied for the first place in the ranking list.

**Phase 2.** Beginning by $A_{ih+1}$, if $Mea(A^{(h+1)}_{ih+1}) - Mea(A^{(i)}_{ih}) < 1/(m - 1)$ and $Mea(A^{(h+1)}_{ih+1}) - Mea(A^{(i)}_{ih}) \geq 1/(m - 1)$, where $l \geq h + 1$, then $\{A_{ih+1}, A_{ih+2}, \cdots, A_{il}\}$ are tied for the second place in the ranking list. In this case, we have that $\{A_{i1}, A_{i2}, \cdots, A_{ih}\} > \{A_{ih+1}, A_{ih+2}, \cdots, A_{il}\}$.

**Phase end.** According to the thought in Phase 2, we can extend the procedure until $A_{im}$ is graded in the ranking list.

According to Criterion 4, the group utility $S_i$ in (14) is also a measure. We search the ranking based on $S_i$ in Table 4. According to Criterion 13, We can see that $S(A^{(1)}_1) = 0.9965, S(A^{(2)}_4) = 0.9947, S(A^{(3)}_4) = 0.9750, S(A^{(4)}_4) = 0.9717, S(A^{(1)}_1) - S(A^{(4)}_4) = 0.9965 - 0.9717 = 0.0248 < 1/3, so the four software products can only be classed to one grade $\{A_1, A_2, A_3, A_4\}$ according to measure $S$.

From Table 4, we can see that $NGU(A^{(1)}_2) = 1, NGU(A^{(2)}_1) = 0.9273, NGU(A^{(3)}_3) = 0.1331, NGU(A^{(4)}_4) = 0$. From the difference $NGU(A^{(2)}_1) - NGU(A^{(3)}_3) = 1 - 0.2923 = 0.0727 < 1/3, and NGU(A^{(2)}_1) - NGU(A^{(4)}_4) = 1 - 0.1331 = 0.8669 > 1/3, NGU(A^{(3)}_3) - NGU(A^{(4)}_4) = 0.1331 < 1/3, so we have $\{A_2, A_3\} \succ \{A_1, A_4\}$ according to measure $NGU$.

Table 4 shows that $R(A^{(1)}_1) = 0.7544, R(A^{(2)}_3) = 0.7596, R(A^{(3)}_4) = 0.7643, R(A^{(4)}_4) = 0.7749$. We determine the ranking in ascending order because the $R$ is a cost index. From the difference $R(A^{(4)}_4) - R(A^{(3)}_4) = 0.0205 < 1/3$, we know that the four software products $\{A_1, A_2, A_3, A_4\}$ is also one grade according to measure $R$.

If the regret measure $R$ is normalized, then the $NGR$ in Table 4 shows that $NGR(A^{(1)}_1) = 1, NGR(A^{(2)}_1) = 0.7467, NGR(A^{(3)}_3) = 0.5184, NGR(A^{(4)}_4) = 0$. From the difference $NGR(A^{(1)}_1) - NGR(A^{(3)}_3) = 1 - 0.7596 = 0.2404 < 1/3, and NGR(A^{(1)}_1) - NGR(A^{(4)}_4) = 1 - 0.5184 = 0.4816 > 1/3, and NGR(A^{(3)}_3) - NGR(A^{(4)}_4) = 0.5184 > 1/3, we know that $\{A_1, A_3\} \succ A_4 \succ A_2$ according to measure $NGR$.

The rankings with classification based on the measures $S, NGU, R, NGR, Q$ in Table 4 are summarized in Table 6.

According to Criterion 13, the VIKOR indexes $S^2, NGU^2, NGR^2, Q^2$ are also some measures to rank the order of four software products. Their results have been shown in Table 5. Similar to above analysis, we note that $S^2(A^{(1)}_1) = 0.5130, S^2(A^{(2)}_1) = 0.5115, S^2(A^{(3)}_3) = 0.4989, S^2(A^{(4)}_4) = 0.4970$. From the difference between maximum and minimum $S^2(A^{(1)}_1) - S^2(A^{(4)}_4) = 0.5130 - 0.4970 = 0.0160 < 1/3$, we can conclude that the four software products $\{A_1, A_2, A_3, A_4\}$ is one grade according to the measure $S^2$.

For the VIKOR index $NGU^2$, Table 5 shows that $NGU^2(A^{(1)}_1) = 1, NGU^2(A^{(2)}_1) = 0.9016, NGU^2(A^{(3)}_3)$
VIKOR indexes

Table 5

| Software | $S_i^2$ | Ranking | $NGU_i^2$ | Ranking | $NGR_i^2$ | Ranking | $Q_i^2$ | Ranking |
|----------|---------|---------|-----------|---------|-----------|---------|---------|---------|
| A_1      | 0.4970  | 4       | 0.0000    | 4       | 0.0000    | 4       | 0.0000  | 4       |
| A_2      | 0.5130  | 1       | 1.0000    | 1       | 1.0000    | 1       | 1.0000  | 1       |
| A_3      | 0.5115  | 2       | 0.9016    | 2       | 0.9273    | 2       | 0.9145  | 2       |
| A_4      | 0.4989  | 3       | 0.1144    | 3       | 0.1331    | 3       | 0.1238  | 3       |

Table 6

| Measure   | A_1 | A_2 | A_3 | A_4 | Ranking with classification |
|-----------|-----|-----|-----|-----|-----------------------------|
| $S$       | ✓   | ✓   | ✓   | ✓   | {A_1, A_2, A_3, A_4}        |
| $NGU_i$   | ✓   | ✓   | ✓   | ✓   | {A_1, A_2, A_3, A_4}        |
| $R$       | ✓   | ✓   | ✓   | ✓   | {A_1, A_2, A_3, A_4}        |
| $NGR$     | ✓   | ✓   | ✓   | ✓   | {A_1, A_3} > A_4 > A_2     |
| $Q$       | ✓   | ✓   | ✓   | ✓   | {A_1, A_2, A_3} > A_4      |
| $S^2$     | ✓   | ✓   | ✓   | ✓   | {A_1, A_2, A_3, A_4}        |
| $NGU^2$   | ✓   | ✓   | ✓   | ✓   | {A_2, A_3} > {A_1, A_4}     |
| $NGR^2$   | ✓   | ✓   | ✓   | ✓   | {A_2, A_3} > {A_1, A_4}     |
| $Q^2$     | ✓   | ✓   | ✓   | ✓   | {A_2, A_3} > {A_1, A_4}     |

VIKOR indexes $S_i^2$, $NGU_i^2$, $NGR_i^2$, $Q_i^2$ and rankings of four software products.

Equation (4) provided a classical projection, which has been used for years. As mentioned in Question 3 in the Introduction, this projection measure is not always reasonable in picture fuzzy setting. This section shows an experimental comparison with it. The data are based on the same illustrative example in Sect. 6.1.

In this subsection, the basic procedure is the same as the algorithm in subsection 5.2, but the projections are based on the classical projection measure. Specifically, (12) is replaced by

$$ Proj_{Y_+}(Y_i) = \frac{Y_i Y_+}{|Y_+|^2}, i \in M, \quad (24) $$

where $|Y_+|^2 = |Y_+|^2, |Y_+|^2$ and $Y_i Y_+$ are the same as in (12).

From the Table 4, which is caused by the different projection measures. Specifically, (12) is replaced by

$$ Proj_{G_+}(Y_i) = \frac{G_i Y_+}{|G_-|^2}, i \in M, \quad (25) $$

where $|G_-|^2 = \sqrt{|G_-|^2}, G_i G_-$ and $|G_-|^2$ are same as in (13).

Equ. (14) is replaced by

$$ S_i = Proj_{Y_+}(Y_i), i \in M, \quad (26) $$

where $Proj_{Y_+}(Y_i)$ is the same as in Eq. (24).

Equ. (16) is replaced by

$$ R_i = Proj_{G_+}(G_i), i \in M, \quad (27) $$

where $Proj_{G_+}(G_i)$ is the same as in Eq. (25).

The comprehensive VIKOR index $Q_i$ is the same as in Eq. (15) and let $\lambda = 0.5$. Three VIKOR indexes and rankings of four products based on the classical projection are summarized in Table 7.

Table 7 shows that $Q(A_1(1)) = Q(A_4(2)) = 1 - 0.8807 = 0.1193 < 1/3$. And we have that $Q(A_1(1)) - Q(A_4(3)) = 1 - 0.3825 = 0.6175 > 1/3$. We further consider that $Q(A_1(1)) - Q(A_4(3)) = 0.3825 > 1/3$. Such Condition 1 is not satisfied. Only Condition 2 is satisfied. So the ranking with classification is $\{A_1, A_4\} > A_3 > A_2$. This ranking with classification is different than the order based on $Q$ from the Table 4, which is caused by the different projection measures.

6.5 Dynamic experiments

This section provides some experiments under dynamic environment. The experimental data are based on the data in Sect. 6.1.

First, let the compromise coefficient $\lambda$ of $Q_i$ in (18) as a dynamic parameter. In order to show a clear figure, let $\lambda = \alpha/100$, where $\alpha \in [0, 100]$. Let $\alpha$ increases from 0 to 100. Then the rankings of four software products $A_1, A_2, A_3, A_4$ based on $Q_i$ in (18) are shown in Fig. 1.

From Fig. 1 we can see that the ranking is $A_1 > A_3 > A_4 > A_2$ when $\alpha$ is about in $[0, 22]$; the ranking is $A_3 > A_1 > A_4 > A_2$ when $\alpha$ is about in $[22, 37]$; the ranking is $A_3 > A_1 > A_2 > A_4$ when $\alpha$ is about in $[37, 50]$; the
ranking is $A_3 > A_2 > A_1 > A_4$ when $\alpha$ is about in [50, 78];
the ranking is $A_3 > A_2 > A_4 > A_1$ when $\alpha$ is about in
[78, 90]; the ranking is $A_2 > A_3 > A_4 > A_1$ when $\alpha$ is
about in [90, 100]. How to define the order of four software
products in whole process? We define the criterion of ranking
with intersection cases as follows.

**Criterion 14** If the curves of rankings of software products
$A_i$ and $A_j$ are intersecting, then the $A_i$ and $A_j$ are graded
into the same classification.

According to Criterion 14, the ranking with classification
of four software products based on $Q_i$ in (18) with parameter
$\lambda$ is $\{A_1, A_2, A_3, A_4\}$.

Second, the experimental data are tested in another
dynamic environment. Now let us observe the $x_{11}^1 = (\mu_{11}^1, \eta_{11}^1, \nu_{11}^1) = (0.49, 0.21, 0.20)$ in $X_1$ in Table 1. Let
$\mu_{11}^1 = \delta/100$, $\eta_{11}^1 = 0.8 - \delta/100$, where the $\delta \in [0, 79]$ is a
dynamic parameter. Other values are the same as in Table 1.
Let $\lambda = 0.5$ in (18), and let $\delta$ increases from 0 to 79. Then
the rankings of four products $A_1, A_2, A_3, A_4$ based on $Q_i$
are shown in Fig. 2.

Figure 2 shows that $A_3$ is ranked in the first as $\delta$
increases/overlaps from 0 to 79; the curves $A_2$ and $A_4$
are intersected by the curve of $A_1$. So we can say that
$A_3 \succ \{A_1, A_2, A_4\}$ according to Criterion 14.

In order to examine the influence of the compromise coef-
cient $\lambda$, we let $\lambda$ be 0.3, 0.4, 0.6 respectively, the rankings of
four products $\{A_1, A_2, A_3, A_4\}$ based on $Q_i$ under the same
dynamic environment as in Fig. 2 are shown in Figs. 3, 4, 5
respectively.

Figure 3 shows that the curves of four products $\{A_1, A_2,$
$A_3, A_4\}$ are intersecting. So we can say that four software
products $\{A_1, A_2, A_3, A_4\}$ are divided into one classification
according to Criterion 14.

Figure 4 shows that the curve of $A_3$ is located above the
$A_1, A_2, A_4$, and the curves of three products $A_1, A_2, A_4$
are intersecting. So we can conclude that $A_3 \succ \{A_1, A_2, A_4\}$
according to Criterion 14.

| Software | $S_i$ | Ranking | $R_i$ | Ranking | $Q_i$ | Ranking |
|----------|-------|---------|-------|---------|-------|---------|
| $A_1$    | 2.3745| 1       | 1.5292| 1       | 1.0000| 1       |
| $A_2$    | 2.2482| 4       | 1.6000| 4       | 0.0000| 4       |
| $A_3$    | 2.2566| 3       | 1.5505| 3       | 0.3825| 3       |
| $A_4$    | 2.3565| 2       | 1.5359| 2       | 0.8807| 2       |
Figure 5 shows that the curve of $A_2$, $A_3$ are located above the $A_1$ and $A_4$, and the curves of $A_1$ and $A_4$ are intersecting. So we can conclude that $A_3 \succ A_2 \succ \{A_1, A_4\}$ according to Criterion 14.

If let compromise coefficient $\lambda$ be 1, then the $Q_i$ in (18) is degenerated into the largest group utility $NGU_i$. If let compromise coefficient $\lambda$ be 0, then the $Q_i$ in (18) is degenerated into the largest group regret $NGR_i$. For the dynamic parameter $\delta$ in Fig. 2, the rankings of four products $A_1$, $A_2$, $A_3$, $A_4$ based on $NGU_i$ in Eq. (15) are shown in Fig. 6.

Figure 6 shows that four curves are disjoint. In this case, we can say that the ranking of four software products is $A_2 \succ A_3 \succ A_4 \succ A_1$ based on $NGU_i$ in (15) under this dynamic sense.

For the dynamic parameter $\delta$ in Fig. 2, the rankings of four products $A_1$, $A_2$, $A_3$, $A_4$ based on $NGR_i$ in Eq. (17) are shown in Fig. 7.

Figure 7 shows that $A_1$ is intersecting to $A_2$, $A_3$, $A_4$. Therefore, the four software products $\{A_1, A_2, A_3, A_4\}$ are divided into one classification based on $NGR_i$ in Eq. (17) with parameter $\delta$.

Now we examine the changes based on the group utility $S_i$ in the dynamic sense. Let parameter $\delta$ be same as in Fig. 2, the rankings of four products based on $S_i$ in (14) are shown in Fig. 8.

Figure 8 shows that four curves are disjoint. Therefore, the ranking based on $S_i$ in (14) with parameter $\delta$ is $A_2 \succ A_3 \succ A_4 \succ A_1$ according to Criterion 14.

Now we examine the ranking based on the VIKOR index $R_i$. From Criterion 8, we known that the smaller the $R_i$, the better the product $A_i$ is. In order to logical consistency, the $R_i$ is transformed as:

$$\overline{R_i} = 1 - R_i, \ i \in M,$$

\(28\)
where it is obvious that the larger the value $\overline{R}_i$, the better the software $A_i$ is.

The rankings of four products $A_1, A_2, A_3, A_4$ based on (28) are shown in Fig. 9, where the dynamic environment is same as in Fig. 2.

Figure 9 shows that $A_1$ is intersecting to $A_2, A_3, A_4$. Therefore, the four software products $\{A_1, A_2, A_3, A_4\}$ are divided into one kind based on $\overline{R}_i$ in (28) with parameter $\delta$.

From above experimental analysis, we can see that (1) the different model may lead to different result; (2) the different experimental condition may lead to different result. We know that there are no the best model for decision science. A good model should be relative to other model. To show the most preferred ranking, a natural idea is that the ranking that occurs more frequently should be considered as a higher acceptance solution. The rankings based on above experimental results, including the results in Table 6, are shown in Table 8.

Table 8 shows that the software $A_3$ is the most preferred product, which appears 16 times in 18 experiments. So this solution should be believable. In this sense, $A_3$ is ideal, followed by $A_2, A_1$ and $A_4$.

7 Conclusion

Regarding the Questions 1 and 4 in Introduction section, this work has provided a VIKOR-based GDM method, in which the users of software are experts. The users’ satisfaction and expectations are reflected in assessment criteria, which are focused on the concerns of the users of software. Regarding the Question 2 in Introduction section, an evaluation method of software reliability has been developed by the aid of a novel VIKOR-based GDM method with picture fuzzy information. Regarding the Question 3 in Introduction section, a new normalization projection measure has been developed in picture fuzzy setting.

These are some main innovations and differences in the new VIKOR-based GDM method, which are listed as follows:

1. The group utility of each alternative $A_i$ is shown by a decision matrix $X_i$ in the new method; whereas the group utility of each alternative $A_i$ is shown by a decision vector in the traditional VIKOR method.
2. The largest group utility is composed by a decision matrix \( Y_+ \) (see (9)) in the new method; whereas the largest group utility (or the best/positive criterion (Opricovic and Tzeng 2007), or ideal solution (Büyüközkan et al. 2019)) is composed by a vector \( f_+ \) in the traditional method. These is no the worst (or negative) criterion \( f_- \) in the new method.

3. A group regret matrix \( G_i \) is provided in the new method.

   The largest group regret matrix is composed by the maximum matrix \( G_- \) of \( G_i(i \in M) \) in the new method; whereas there is no group regret in the traditional method.

4. The group utility measurement is based on the normalization projection of \( Y_i \) on \( Y_+ \) in the new method; whereas the group utility measurement is based on the positive and negative ideal solutions shown by two vectors \( f_+ \) and \( f_- \) in the traditional method.

5. The group regret measurement is based on the normalization projection of \( G_i \) on \( G_- \) in the new method; whereas the group regret measurement is based on the positive and negative ideal solutions shown by two vectors \( f_+ \) and \( f_- \) in the traditional method.

6. A new ranking method under a static environment and a new ranking method in a dynamic environment are provided in this work; whereas only a compromise solution can be obtained in the traditional method.

7. The closeness between two decision matrices is based on a normalization projection in the new method; whereas the closeness between two decision matrices is based on the Euclidean distance or the Hamming distance in the traditional method (Çali and Balaman 2019).

8. There is no the collective decision aggregated by all the individual decisions in decision process in the new method; There is a collective decision aggregated by all the individual decisions in decision process in the traditional method.

The above-mentioned differences and innovations have led to a technical promotion of VIKOR-based GDM method. The technical promotion in this work can greatly improve the evaluation effect of software reliability.

We know that any model has its limitations. The new model provided in this paper is no exception. First, the evaluation information is expressed by PFN. This is a limitation. The future research should extend to other measures. For example, the Euclidean distance or the Hamming distance. Fourth, the proposed model is only implemented in software reliability evaluation. This is a limitation. The future research may be implemented in a real time datasets, like UCI Machine Learning Repository data to evaluate the performance of some software qualities. And the time and space complexity will also be considered in future researches, where the algorithm will be specified with more details for the software reliability evaluation. Fifth, the effectiveness of the suggested method is verified in some static experiments and some dynamic experiments. This is a limitation. The effectiveness of the suggested model should also be demonstrated by more methods in future researches, like normal simulation, real-time simulation, and so on.

Acknowledgements The author would like to thank the editors and the anonymous reviewers for their insightful and constructive comments and suggestions that have led to this improved version of the paper.

Funding This work was supported partially by the Science and Technology Planning Project from Zhanjiang Science and Technology Bureau (No.2020B01415) and Project of Teaching Innovation in 2022 from Guangdong Ocean University (No. PX-131223610).

Data availability Enquiries about data availability should be directed to the authors.

Declarations

Conflict of interest No potential conflict of interest was reported by author.

Ethical approval This article does not contain any studies with human participants performed by author.

References

Abuta E, Tian J (2018) Reliability over consecutive releases of a semiconductor optical Endpoint Detection software system developed in a small company. J Syst Softw 137:355–365
Atanassov K (1986) Intuitionistic fuzzy sets. Fuzzy Sets Syst 20(1):87–96
Bertsatos A, Christaki A, Chovalopoulou M-E (2019) Testing the reliability of 3D-ID software in sex and ancestry estimation with a modern Greek sample. Forensic Sci Int 297:132–137
Büyüközkan G, Göçer F, Karabulut Y (2019) A new group decision making approach with IF AHP and IF VIKOR for selecting hazardous waste carriers. Measurement 134:66–82
Büyüközkan G, Güler M (2020) Smart watch evaluation with integrated hesitant fuzzy linguistic SAW-ARAS technique. Measurement 153:107353
Çali S, Balaman SY (2019) A novel outranking based multi criteria group decision making methodology integrating ELECTRE and VIKOR under intuitionistic fuzzy environment. Exp Syst Appl 119:36–50
Wu H, Xu Z, Ren P, Liao H (2018) Hesitant fuzzy linguistic projection model to multi-criteria decision making for hospital decision support systems. Comput Ind Eng 115:449–458

Wu Q, Lin W, Zhou L, Chen Y, Chen H (2019) Enhancing multiple attribute group decision making flexibility based on information fusion technique and hesitant Pythagorean fuzzy sets. Comput Ind Eng 127:954–970

Wu Q, Zhou L, Chen Y, Chen H (2019) An integrated approach to green supplier selection based on the interval type-2 fuzzy best-worst and extended VIKOR methods. Inform Sci 502:394–417

Wu Z, Ahmad J, Xu J (2016) A group decision making framework based on fuzzy VIKOR approach for machine tool selection with linguistic information. Appl Soft Comput 42:314–324

Wu Z, Xu J, Jiang X, Zhong L (2019) Two MAGDM models based on hesitant fuzzy linguistic term sets with possibility distributions: VIKOR and TOPSIS. Inform Sci 473:101–120

Xu G, Liu F (2013) An approach to group decision making based on interval multiplicative and fuzzy preference relations by using projection. Appl Math Modell 37(6):3929–3943

Ye J (2017) Bidirectional projection method for multiple attribute group decision making with neutrosophic numbers. Neural Comput Appl 28(5):1021–1029

You X-Y, You J-X, Liu H-C, Zhen L (2015) Group multi-criteria supplier selection using an extended VIKOR method with interval 2-tuple linguistic information. Exp Syst with Appl 42(4):1906–1916

Yue C (2016) A geometric approach for ranking interval-valued intuitionistic fuzzy numbers with an application to group decision-making. Comput Ind Eng 102:233–245

Yue C (2017) Entropy-based weights on decision makers in group decision-making setting with hybrid preference representations. Appl Soft Comput 60:737–749

Yue C (2017) Two normalized projection models and application to group decision-making. J of Int Fuzzy Syst 32(6):4389–4402

Yue C (2019) An interval-valued intuitionistic fuzzy projection-based approach and application to evaluating knowledge transfer effectiveness. Neural Comput Appl 31(11):7685–7706

Yue C (2019) A normalized projection-based group decision-making method with heterogeneous decision information and application to software development effort assessment. Appl Int 49(10):3587–3605

Yue C (2019) Projection-based approach to group decision-making with hybrid information representations and application to software quality evaluation. Comput Ind Eng 132:98–113

Yue C (2020) An intuitionistic fuzzy projection-based approach and application to software quality evaluation. Soft Comput 24:429–433

Yue C (2020) Picture fuzzy normalized projection and extended VIKOR approach to software reliability assessment. Applied Soft Computing, p 106056

Yue C, Yue Z (2019) Measuring the satisfaction and loyalty of Chinese smartphone users: A simple symbol-based decision-making method. Sci Iran 26(1):589–604

Zadeh L (1965) Fuzzy sets Inform control 8(3):338–353

Zeng S, Chen S-M, Kuo L-W (2019) Multiattribute decision making based on novel score function of intuitionistic fuzzy values and modified VIKOR method. Inform Sci 488:76–92

Zhang X-Y, Wang J-Q, Hu J-H (2018) On novel operational laws and aggregation operators of picture 2-tuple linguistic information for MCDM problems. Int J Fuzzy Syst 20(3):938–969

Zhao X, Band SS, ElNaflir S, Sookhak M, Mosavi A, Salwana E (2021) The implementation of border gateway protocol using software-defined networks: A systematic literature review. IEEE Access

Zhu M, Pham H (2017) Environmental factors analysis and comparison affecting software reliability in development of multi-release software. J Syst Softw 132:72–84

Zou B, Yang M, Zhang J (2018) Reliability analysis and allocation: Development of a hierarchical structure modeling platform in I&C system Software Life Cycle. Nucl Eng Des 328:345–352

Publisher’s Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.