Journal of Disability Research

2025 | Volume 4 | Pages: 1–15 | e-location ID: e20240107
DOI: 10.57197/JDR-2024-0107

Pythagorean Cubic Normal Fuzzy Information Aggregation Operators and Their Application in Disability Evaluation

Muneeza[1,*], Mariya Gul[1], Samah M. Alzanin[2] and Abdu H. Gumaei[2]

[1]Department of Mathematics, Government Postgraduate College for Women, Mardan, Khyber Pakhtunkhwa, Pakistan
[2]Department of Computer Science, College of Computer Engineering and Sciences, Prince Sattam bin Abdulaziz University, Al-Kharj 11942, Saudi Arabia

Correspondence to:
Muneeza*, e-mail: muneeza0315@gmail.com, Tel.: +923229425720, Fax: +966115882101

Received: June 5 2024; Revised: August 15 2024; Accepted: August 25 2024; Published Online: January 3 2025

ABSTRACT

Normal fuzzy sets and Pythagorean cubic fuzzy sets are the best means to deal with fuzziness. Combining both of these structures in our current work, we establish the idea of Pythagorean cubic normal fuzzy set. We define some basic operational laws for Pythagorean cubic normal fuzzy set and introduce a number of aggregation operators, including Pythagorean cubic normal fuzzy weighted averaging operator, Pythagorean cubic normal fuzzy weighted geometric operator, Pythagorean cubic normal fuzzy order weighted averaging operator and Pythagorean cubic normal fuzzy order weighted geometric operator. We examine several favorable properties, including monotonicity, boundedness, and idempotency for the proposed operators. We develop an algorithm for the solution of multicriteria decision-making problems. Moreover, we propose an extended form of the TODIM (Portuguese acronym for Interactive Multi-Criteria Decision Making) method. We present a multicriteria decision-making example related to assessing the educational needs of students with disabilities. The techniques and operators defined in the current work provide greater generality and accuracy and give precise results. Ultimately, a detailed illustration is provided to show the closure process of these specified procedures and functions, demonstrating their credibility and efficacy.

KEYWORDS

Education, TODIM method, Pythagorean cubic normal fuzzy numbers, aggregation operators, multicriteria decision-making

INTRODUCTION

Multiple-criteria decision-making (MCDM) is one of the topmost appropriate methods in decision-making theory which has been rapidly utilized in various human activities (Ivanov and Webster, 2024; Wang et al., 2024). The MCDM method determines the best alternative among a group of alternatives according to established criteria (Muneeza et al., 2022). Sometimes uncertainty may arise in decision-making problems from time constraints, conflicting objectives, or inadequate information (Muneeza et al., 2023, 2024). To deal with this uncertainty Zadeh et al. (1996) in the 1960s developed the fuzzy set (FS) theory as a mathematical representation of the vagueness and uncertainty inherent in human language and thought. However, labeling the more composite fuzzy information was still challenging as FS was unable to count for the degree of hesitation. Therefore, Atanassov (1999) extended the concept of FS and put forward a new concept, named intuitionistic fuzzy set (IFS), which incorporated the degree of hesitation (Deveci and Güler, 2024).

As the complexities of human social activities grow rapidly over time, managing uncertain data from various sources, such as logbooks, resources, and specialists, can make navigating through unclear situations even more challenging. The intuitionistic fuzzy sets were intensively studied and generalized into various forms; a few are interval valued fuzzy set (IVFS) (Wasim et al., 2024), triangular interval type-2 (Tian et al., 2024), interval valued hesitant fuzzy set (Xian et al., 2024), interval type-2 trapezoidal fuzzy (Mollaoglu et al., 2023), normal fuzzy set (Rickard et al., 2024), and many more. Yager and Abbasov (2013) and Yager (2013) further generalized the concept of IFSs to effectively characterize fuzzy information and established a new concept called Pythagorean fuzzy set (PyFS); similarly, Verma and Sherma outlined PyFS based on exponential entropy (Premalatha and Dhanalakshmi, 2022; Verma and Mittal, 2023). Peng et al. introduced several novel operational laws in PyFSs (Peng et al., 2022; Akram et al., 2023).

Wang defined q-rung orthopair fuzzy sets (q-ROFSs), as well as metric (Wang et al., 2019a). Wei et al., introduced Pythagorean cubic fuzzy sets (PyCFSs) bidirectional projection technique (Wei et al., 2019). Many researchers have done a lot of work on PyCFS (Khan et al., 2020; Rahim, 2023) and also on complex spherical fuzzy sets (Zahid et al., 2022; Akram et al., 2021a). In the Pythagorean fuzzy environment, various researchers proposed some basic aggregation operators (AOs) including averaging (Singh and Ganie, 2020), geometric (Riaz et al., 2020), Hamacher (Hadi et al., 2021), and Yager (Akram et al., 2021b) AOs. The aggregation operations play a key role in combining and analyzing fuzzy data (Paul et al., 2023b). Alamoodi et al. presented geometric AOs (Alamoodi et al., 2022). Hussain et al. (2022) proposed novel Aczel–Alsina AOs for PyFSs. Moreover, Hussain et al. (2023) and Cui et al. (2023) introduced a procedure for the calculation of the distance between two Pythagorean fuzzy numbers (PyFNs) which is based on Hausdorff metric space. Similarly, Kumar and Gupta have proposed the Technique for Order Preference by Similarity to Ideal Solution (TOPSIS) approach with PyFNs (Kumar and Gupta, 2023), Zhang et al. (2023) introduced the generalization of the TOPSIS method, and Chen (2020) has proposed the Pythagorean fuzzy multiple-criteria decision analysis. Gao analyzed the Pythagorean fuzzy Hamacher prioritized AOs (Paul et al., 2023a). Wang et al. proposed the idea of few mean operators based on q-ROFS (Wang et al., 2019b).

All the mentioned theories are more vastly used in various fields; however, still the above concepts do not significantly improve the articulation of the main points. So, for this purpose, Khan et al. (2019b) suggested PyCFS and defined the Pythagorean cubic fuzzy (PyCF) AOs. PyCFS is an extended form of interval-valued PyFS (Chander and Das, 2021). In PyCFS, each element consists of membership and non-membership degrees with the conditions, for all n, $p \in [0, 1]$, $n^2 + p^2 \leq 1$. Khan et al. (2019a) and Rahim et al. (2024) developed AOs for PyCFSs. In addition, other kinds of AOs for PyCFSs were introduced by different authors, for example, Wang et al. (2018) defined PyCF Muirhead logical operators. Literature review is shown in Figure 1 as well.

In the current study, we establish a new FS, called the Pythagorean cubic normal fuzzy set (PyCNFS), and its application in disability evaluation. Disability refers to a physical, intellectual, perceptual, or developmental condition that constraints a person ability to involve in certain activities or interact with their environment in a conventional manner. Disabilities can vary widely in intensity and influence, and they may be temporary or permanent. Demeter (2012) discussed disability evaluation in his article and differentiated disability and handicap. In the current study, we propose that disability evaluation involves evaluating the availability and embracing diversity of various aspects of society to ensure they meet the requirements of people with disabilities. This evaluation procedure normally include, physically accessibility, information and communication accessibility, policy and legal frameworks, employment and education opportunities, healthcare and social services, and also community and social inclusion. Therefore, the purpose of the current work is to provide the implementation of a disability evaluation system that holds significant promise in promoting fairness, efficiency, and accuracy in accessing individuals' disabilities.

The remaining part of this article is arranged as follows: the Preliminaries section contains various ideas related to our proposed work. In the AOs for Pythagorean Cubic Normal Fuzzy Numbers section, we present the concept PyCNFSs, their operational laws for PyCNFSs, Pythagorean cubic normal fuzzy AOs, and their properties. The Multicriteria Decision-making Algorithm section presents multicriteria decision-making algorithm based on the proposed AOs. The TODIM Method section presents extended TODIM approach with Pythagorean cubic normal fuzzy numbers (PyCNFNs) and gives detailed instructions. In the Illustrative Example section, a numerical example of disability evaluation is given to demonstrate the viability of our new model. The Comparative Analysis section presents the comparison of our proposed techniques with other existing techniques. The conclusion of our work is given in the Conclusions and Future Work section.

Literature review

Here, we present the review of FSs and related research motivations. The details of which are present in Figure 1 as follows.

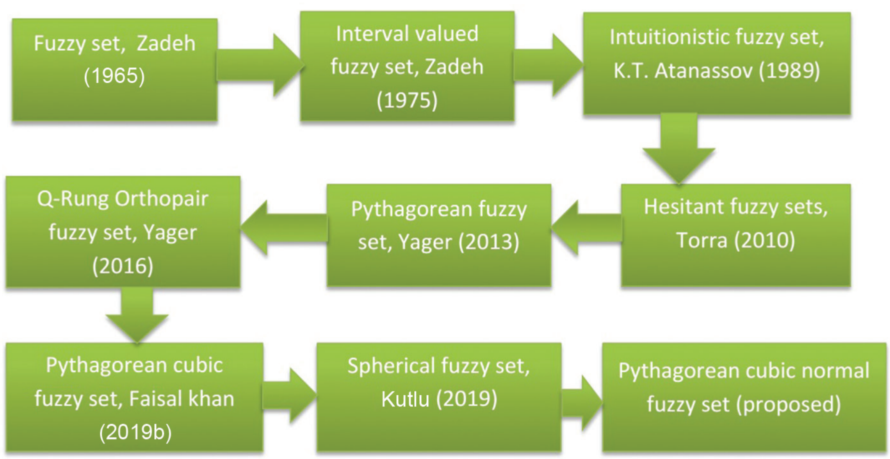

Figure 1: Literature review.

Table 1: Research on PyCFS.

Authors	Fuzzy set	Method	Research focus
Khan et al. (2019b)	PyCFS	TOPSIS	Investment corporation
Seker and Kahraman (2022)	PyCFS	TOPSIS and TODIM	Software selection program
Abdullah et al. (2022)	PyCFS	Hamacher operators	Green supplier management
Amin et al. (2022)	PyCFS	Generalized operators	Disaster risk management
Palanikumar et al. (2023)	PyCFS	Combination with soft sets	Laptop evaluation
Al-Sabri et al. (2023)	PyCFS	Einstein operators	Depression and anxiety

Abbreviation: PyCFS, Pythagorean cubic fuzzy set.

Table 2: Research on normal numbers.

Authors	Structure	Method	Research focus
Zhang et al. (2016)	Normal number	Aggregation operators	Cloud generator
Sherwani et al. (2021)	Normal number	Variance, generating functions	Extension of Smarandache
Liu and Li (2017)	Normal number	Score function operational laws	Bonferoni mean operators
Palanikumar et al. (2024)	Normal number	Interaction based decision-making	Robot technology
Rickard et al. (2024)	Normal number	Score function operational laws	Decision-making methods

In Table 1, the details of the work on PyCFS are given, while in Table 2, the details of the work on normal numbers are given.

PRELIMINARIES

We present the PyCFS and its fundamental properties on a universal set S.

Definition 1 (Yang and Ko, 1996): *Let S be the real number set, the membership function of fuzzy number is*

$$Z(x) = e^{\left(\frac{x-\beta}{\pounds}\right)^2} \quad (\pounds > 0),$$

is called as a normal fuzzy number (NFN) Z = (β + £), the normal fuzzy number set is denoted by K.

Definition 2 (Khan et al., 2019b): *Suppose that S is a fixed set, a PyCFS is defined as follows:*

$$\varrho_\kappa = \left\{ \left\langle \tau, \left(\mu_{\kappa_1}(\tau), \rho_{\kappa_1}(\tau)\right)\right\rangle \middle| \tau \in S \right\}, \tag{1}$$

where $\rho_{\kappa_1}(\tau) = \langle \tilde{F}(\tau), \mu(\tau)\rangle, \mu_{\kappa_1}(\tau) = \langle F(\tau), \lambda(\tau)\rangle$, *such that* $0 \le \mu^2(\tau) + \lambda^2(\tau) \le 1$ *and* $0 \le \sup\left(F(\tau)\right)^2 + \left(\left(\sup\left(\tilde{F}(\tau)\right)\right)\right)^2 \le 1.$

Definition 3 (Khan et al., 2019b): *Let* $\varrho_{\kappa_1} = \left(\langle t_1, \lambda_1\rangle, \langle t_1, \mu_1\rangle\right)$, $\varrho_{\kappa_2} = \left(\langle F_2, \lambda_2\rangle, \langle F_2, \mu_2\rangle\right), \varrho_\kappa = \left(\langle F, \lambda\rangle, \langle F, \mu\rangle\right)$ *are three PyCF numbers (PyCFNs) and* $\varphi > 0, \varphi_1 > 0, \varphi_2 > 0$, *where* $t_1 = [t_1, s_1]$, $t_1 = [t_1, \tilde{s}_1], F_2 = [r_2, s_2], F_2 = [\tilde{r}_2, \tilde{s}_2], F = [r, s], F = [\tilde{r}, \tilde{s}]$. *Then, the below properties hold.*

1. $\varphi\left(\varrho_{\kappa_2} \oplus \varrho_{\kappa_1}\right) = \varphi\left(\varrho_{\kappa_2}\right) \oplus \varphi\left(\varrho_{\kappa_1}\right)$
2. $\varrho_{\kappa_2} \otimes \varrho_{\kappa_1} = \varrho_{\kappa_1} \otimes \varrho_{\kappa_2}$
3. $\varrho_{\kappa_2} \oplus \varrho_{\kappa_1} = \varrho_{\kappa_1} \oplus \varrho_{\kappa_2}$
4. $(\varphi_1 + \varphi_2)\varrho_\kappa = \varphi_1\varrho_\kappa \oplus \varphi_2\varrho_\kappa$
5. $\varrho_\kappa^{\varphi_1} \otimes \varrho_\kappa^{\varphi_2} = \varrho_\kappa^{(\varphi_1+\varphi_2)}$
6. $\left(\varrho_{\kappa_2} \otimes \varrho_{\kappa_1}\right)^\varphi = \left(\varrho_{\kappa_2}\right)^\varphi \otimes \left(\varrho_{\kappa_1}\right)^\varphi.$

Definition 4 (Khan et al., 2019b): *Let* $\varrho_\kappa = \left(\langle t_1, \lambda_1\rangle, \langle t_1, \mu_1\rangle\right)$ *be a PyCFN,* $t_1 = [t_1, s_1], t_1 = [t_1, \tilde{s}_1]$. *The score function of* ϱ_κ *can be presented as follows:*

$$S\left(\varrho_\kappa\right) = \left(\frac{t_1 + s_1 - \lambda_1}{3}\right)^2 - \left(\frac{t_1 + \tilde{s}_1 - \mu_1}{3}\right)^2, \tag{2}$$

where $S(\varrho_\kappa) \in [-1, 1]$. *Let* $\left(\varrho_{\kappa_1} = \langle t_1, \lambda_1\rangle, \langle t_1, \mu_1\rangle\right)$ *and* $\varrho_{\kappa_2} = \left(\langle r_2, \lambda_2\rangle, \langle r_2, \mu_2\rangle\right)$ *be two PyCFN where* $t_1 = [t_1, s_1]$, $t_1 = [t_1, \tilde{s}_1]$, *and* $S\left(\varrho_{\kappa_1}\right)$ *and* $S\left(\varrho_{\kappa_2}\right)$ *be the score functions of* ϱ_{κ_1} *and* ϱ_{κ_2}, *respectively. Then,*

1. *If* $S\left(\varrho_{\kappa_2}\right) = S\left(\varrho_{\kappa_1}\right)$, *then* $\varrho_{\kappa_2} \sim \varrho_{\kappa_1}$
2. *If* $S\left(\varrho_{\kappa_2}\right) > S\left(\varrho_{\kappa_1}\right)$, *then* $\varrho_{\kappa_2} > \varrho_{\kappa_1}$
3. *If* $S\left(\varrho_{\kappa_2}\right) < S\left(\varrho_{\kappa_1}\right)$, *then* $\varrho_{\kappa_2} < \varrho_{\kappa_1}$,

Definition 5 (Khan et al., 2019b): *Let* $\varrho_\kappa = \left[\langle t_1, \lambda_1\rangle, \langle t_1, \mu_1\rangle\right]$ *be a PyCFN. Then, the accuracy degree of* ϱ_κ *by* $\beta(\varrho_\kappa)$ *is defined as follows:*

$$\beta\left(\varrho_\kappa\right) = \left(\frac{t_1 + s_1 - \lambda_1}{3}\right)^2 + \left(\frac{t_1 + \tilde{s}_1 - \mu_1}{3}\right)^2, \tag{3}$$

where $S_1 = [t_1, s_1], t_1 = [t_1, \tilde{s}_1]\beta\left(\varrho_\kappa\right) \in [0,1]$.

Definition 6 (Khan et al., 2019b): *Let* $\varrho_{\kappa_1} = \left(\langle t_1, \lambda_1\rangle, \langle t_1, \mu_1\rangle\right)$ *and* $\varrho_{\kappa_2} = \left(\langle r_2, \lambda_2\rangle, \langle r_2, \mu_2\rangle\right)$ *be two PyCFNs, where* $t_1 = [t_1, s_1], t_1 = [t_1, \tilde{s}_1]$, *and* $\beta\left(\varrho_{\kappa_1}\right)$ *and* $\beta\left(\varrho_{\kappa_2}\right)$ *be the accuracy degrees of* ϱ_{κ_1} *and* ϱ_{κ_2}, *respectively. Then,*

1. *If* $\beta\left(\varrho_{\kappa_2}\right) = \beta\left(\varrho_{\kappa_1}\right)$, *then* $\varrho_{\kappa_2} \sim \varrho_{\kappa_1}$.
2. *If* $\beta\left(\varrho_{\kappa_2}\right) > \beta\left(\varrho_{\kappa_1}\right)$, *then* $\varrho_{\kappa_2} > \varrho_{\kappa_1}$.
3. *If* $\beta\left(\varrho_{\kappa_2}\right) < \beta\left(\varrho_{\kappa_1}\right)$, *then* $\varrho_{\kappa_2} < \varrho_{\kappa_1}$.

Definition 7 (Khan et al., 2019b): *Let* ϱ_{κ_1} *and* ϱ_{κ_2} *be two any PyCFNs on a set* $T = \{t_1, t_2, \ldots, t_\alpha\}$. *Then, the distance measure between* ϱ_{κ_1} *and* ϱ_{κ_2} *can be defined in the following way:*

$$d\left(\varrho_{\kappa_1},\varrho_{\kappa_2}\right)=\frac{1}{6}\Big[\left|t_1-r_2^2\right|+\left|s_1-s_2\right|+\left|t_1-r_2^2\right|+\left|\tilde{s}_1-\tilde{s}_2\right| \\ +\left|\lambda_1^2-\lambda_2^2\right|+\left|\mu_1^2-\mu_2^2\right|\Big]. \tag{4}$$

Definition 8 (Khan et al., 2019b): *Let* $\varrho_{\kappa_i}=\left(\left\langle t_i^2,\lambda_i\right\rangle,\left\langle t_i^2,\mu_i\right\rangle\right)$, *where* $(i=1,2,3,\ldots,m)$ *be a collection of all PyCFNs and* $S_1=\left[t_1,s_1\right],S_1=\left[t_1,\tilde{s}_1\right]$ *while* $w=(w_1,w_2,\ldots,w_\alpha)^T$ *be the weight vector (WeV) of* ϱ_{κ_i} *where* $(i=1,2,3,\ldots,m)$ *with* $w_i\geq 0$ *where* $w_i\in[0,1]$ *and* $\sum_{i=1}^m w_i=1$. *Then, the PyCF weighted averaging (PyCFWA) operator is defined as follows:*

$$PyCFWA\left(\varrho_{\kappa_1},\varrho_{\kappa_2},\ldots,\varrho_{\kappa_m}\right)=\left\{\begin{array}{c}\left(\begin{array}{c}\left[\sqrt{1-\Pi_{i=1}^m\left(1-t_i^2\right)^{v_i}},\sqrt{1-\Pi_{i=1}^m\left(1-s_i^2\right)^{v_i}}\right],\\ \sqrt{1-\Pi_{i=1}^m\left(1-\lambda_i^2\right)^{v_i}}\end{array}\right),\\ \left(\left[\Pi_{i=1}^m r_i^{v_i},\Pi_{i=1}^m\tilde{s}_i^{v_i}\right],\Pi_{i=1}^m\mu_i^{v_i}\right)\end{array}\right\}. \tag{5}$$

Definition 9 (Khan et al., 2019b): *Let* $\varrho_{\kappa_i}=\left(\left\langle t_i^2,\lambda_i\right\rangle,\left\langle t_i^2,\mu_i\right\rangle\right)$, *where* $(i=1,2,3,\ldots,m)$ *be a group of all PyCFNs and* $t_i^2=\left[t_i^2,s_i\right],t_i^2=\left[t_i^2,\tilde{s}_i\right]$ *and* $w=(w_1,w_2,\ldots,w_\alpha)^T$ *be the WeV of* ϱ_{κ_1} *where* $(i=1,2,3,\ldots,m)$ *with* $w_i\geq 0,\sum_{i=1}^m w_i=1$ and $w_i\in[0,1]$. *Then, utilizing the PyCF weighted geometric (PyCFWG) operator the aggregate result is also a PyCFN and*

$$PyCFWG\left(\rho_{\kappa_1},\rho_{\kappa_2},\ldots,\rho_{\kappa_m}\right)=\left\{\begin{array}{c}\left(\left[\Pi_{i=1}^m r_i^{v_i},\Pi_{i=1}^m s_i^{v_i}\right],\Pi_{i=1}^m\lambda_i^{v_i}\right)\\ \left(\begin{array}{c}\left[\sqrt{1-\Pi_{i=1}^m\left(1-t_i^2\right)^{v_i}},\sqrt{1-\Pi_{i=1}^m\left(1-s_i^2\right)^{v_i}}\right],\\ \sqrt{1-\Pi_{i=1}^m\left(1-\mu_i^2\right)^{v_i}}\end{array}\right)\end{array}\right\}. \tag{6}$$

AOs FOR PYTHAGOREAN CUBIC NORMAL FUZZY NUMBERS

Here, we present several AOs for Pythagorean cubic normal fuzzy numbers (PyCNFNs) along with their corresponding properties.

Definition 10: *Let S be an ordinary fixed non-empty set and*

$$\upsilon_{\kappa i}=\left\{\left\langle\beta_i,\pounds_i\right\rangle:\left[\left\langle\delta_{ij},\nu_{ij}\right\rangle,\left\langle\breve{\delta}_{ij},\varrho\right\rangle\right]\right\},(\beta,\pounds)\in N,$$

is a PCNFN. Where its membership function is defined as $\left[\nu_{ij},\varrho\right]=\left[\nu_{ij}e^{\left(\frac{\beta-\pounds}{\gamma}\right)^2},\varrho e^{\left(\frac{\beta-\pounds}{\gamma}\right)^2}\right]$, $s\in S$ *and non-membership function is defined as* $\left[\nu_{ij},\varrho_{ij}\right]=\left[\begin{array}{c}1-\left(1-\nu_{ij}\right)e^{-\left(\frac{x-\beta}{\gamma}\right)^2}\\ 1-\left(1-\varrho_{ij}\right)e^{-\left(\frac{x-\beta}{\gamma}\right)^2}\end{array}\right],s\in S,$

where $\left[\delta_{ij},\nu_{ij}\right]\in[0,1]$ *and* $\left[\breve{\delta}_{ij},\varrho_{ij}\right]\in[0,1]$, *and* $0\leq\left[\delta_{ij},\nu_{ij}\right]^2+\left[\breve{\delta}_{ij},\varrho_{ij}\right]^2\leq 1$.

New operational laws for PyCNFSs

Proposition 1: *Let* $\upsilon_{\kappa_1}\upsilon_{\kappa_2}$ *be PyCNFNs, then the below operational laws exist.*

1. $\upsilon_{\kappa_1}\oplus\upsilon_{\kappa_2}$

$$=\left\{\begin{array}{c}\left\langle\beta_i,\pounds_i\right\rangle\left\langle\left[\sqrt{1-\Pi_{i=1}^m\left(1-t_i^2\right)^{v_i}},\sqrt{1-\Pi_{i=1}^m\left(1-s_i^2\right)^{v_i}}\right],\right.\\ \left.\sqrt{1-\Pi_{i=1}^m\left(1-\lambda_i^2\right)^{v_i}}\right\rangle,\left\langle\left[\Pi_{i=1}^m\tilde{r}_i^{v_i},\Pi_{i=1}^m\tilde{s}_i^{v_i}\right],\Pi_{i=1}^m\mu_i^{v_i}\right\rangle\end{array}\right\};$$

2. $\upsilon_{\kappa_1}\otimes\upsilon_{\kappa_2}$

$$=\left\{\begin{array}{c}\left\langle\beta_i,\pounds_i\right\rangle,\left\langle\left[\Pi_{i=1}^m\tilde{r}_i^{v_i},\Pi_{i=1}^m\tilde{s}_i^{v_i}\right],\Pi_{i=1}^m\mu_i^{v_i}\right\rangle,\\ \left\langle\left[\sqrt{1-\Pi_{i=1}^m\left(1-t_i^2\right)^{v_i}},\sqrt{1-\Pi_{i=1}^m\left(1-s_i^2\right)^{v_i}}\right],\right.\\ \left.\sqrt{1-\Pi_{i=1}^m\left(1-\lambda_i^2\right)^{v_i}}\right\rangle\end{array}\right\};$$

3. $\varphi\upsilon_{\kappa_1}$

$$=\left\{\begin{array}{c}\left\langle\beta_i,\pounds_i\right\rangle,\left\langle\left[\sqrt{1-\Pi_{i=1}^m\left(1-t_i^2\right)^{v_i}},\sqrt{1-\Pi_{i=1}^m\left(1-s_i^2\right)^{v_i}}\right],\right.\\ \left.\sqrt{1-\Pi_{i=1}^m\left(1-\lambda_i^2\right)^{v_i}}\right\rangle,\left\langle\left(\Pi_{i=1}^m\tilde{r}_i^{v_i}\right)^{\varphi},\left(\Pi_{i=1}^m\tilde{s}_i^{v_i}\right)^{\varphi},\right.\\ \left.\left(\Pi_{i=1}^m\mu_i^{v_i}\right)^{\varphi}\right\rangle\end{array}\right\};$$

4. $\upsilon_{\kappa_1}^{\varphi}$

$$=\left\{\begin{array}{c}\left\langle\beta_i,\pounds_i\right\rangle,\left\langle\left(\Pi_{i=1}^m\tilde{r}_i^{v_i}\right)^{\varphi},\left(\Pi_{i=1}^m\tilde{s}_i^{v_i}\right)^{\varphi},\left(\Pi_{i=1}^m\mu_i^{v_i}\right)^{\varphi}\right\rangle,\\ \left\langle\left[\sqrt{1-\Pi_{i=1}^m\left(1-t_i^2\right)^{v_i}},\sqrt{1-\Pi_{i=1}^m\left(1-s_i^2\right)^{v_i}}\right],\right.\\ \left.\sqrt{1-\Pi_{i=1}^m\left(1-\lambda_i^2\right)^{v_i}}\right\rangle\end{array}\right\};$$

5. $\upsilon_{\kappa_1}^{\kappa}=\left\{\left\langle\beta_i,\pounds_i\right\rangle,\left\langle t_1\lambda_1\right\rangle,\left\langle t_1\mu_1\right\rangle\right\}$;

Definition 11: *Let* $\upsilon_{\kappa i}=\left\langle\beta_i,\pounds_i\right\rangle\left[\left\langle\delta_{ij},\nu_{ij}\right\rangle,\left\langle\breve{\delta}_{ij},\varrho_{ij}\right\rangle\right]$ *be a PyCNFN,* $t_1=\left[t_1,s_1\right],t_1=\left[t_1,\tilde{s}_1\right]$. *The score function of* $\upsilon_{\kappa i}$ *can be represented as follows:*

$$S\left(\upsilon_{\kappa i}\right)=\left(\frac{\beta_i+\pounds_i}{2}\right)\left(t_1+s_1-\lambda_1\right)^2-\left(t_1+\tilde{s}_1-\mu_1\right)^2, \tag{7}$$

where $S\left(\upsilon_{\kappa i}\right)\in[-1,1]$. *Let* $\upsilon_{\kappa_1}=\left(\left\langle t_1,\lambda_1\right\rangle,\left\langle t_1,\mu_1\right\rangle\right)$ *and* $\upsilon_{\kappa 2}=\left(\left\langle r_2,\lambda_2\right\rangle,\left\langle r_2,\mu_2\right\rangle\right)$ *be two PyCNFN where* $t_1=\left[t_1,s_1\right]$, $t_1=\left[t_1,\tilde{s}_1\right]$, *and* $S\left(\upsilon_{\kappa_1}\right)$ *and* $S\left(\upsilon_{\kappa_2}\right)$ *be the score functions of* υ_{κ_1} *and* υ_{κ_2}, *respectively. Then,*

1. *If* $S\left(\upsilon_{\kappa_2}\right)=S\left(\upsilon_{\kappa_1}\right)$, *then* $\varrho_{\upsilon_2}\sim\varrho_{\upsilon_1}$
2. *If* $S\left(\upsilon_{\kappa_2}\right)>S\left(\upsilon_{\kappa_1}\right)$, *then* $\varrho_{\upsilon_2}>\varrho_{\upsilon_1}$
3. *If* $S\left(\upsilon_{\kappa_2}\right)<S\left(\upsilon_{\kappa_1}\right)$, *then* $\varrho_{\upsilon_2}<\varrho_{\upsilon_1}$,

Definition 12: *Let* $\varrho_{\kappa}=\left[\left\langle t_1,\lambda_1\right\rangle,\left\langle t_1,\mu_1\right\rangle\right]$ *be a PyCNFN. Then, the accuracy degree of* ϱ_{κ} *by* $\beta(\varrho_{\kappa})$ *is defined as follows:*

$$S(\upsilon_{\kappa i})=\left(\frac{\beta_i+£_i}{2}\right)\left(\frac{t_1+s_1-\lambda_1}{3}\right)^2-\left(\frac{t_1+\tilde{s}_1-\mu_1}{3}\right)^2, \quad (8)$$

where $S_1=[t_1,s_1], t_1=[t_1,\tilde{s}_1]\beta(\varrho_\kappa)\in[0,1].$

Definition 13: *Let* $\upsilon_{\kappa_1}=(\langle t_1,\lambda_1\rangle,\langle t_1,\mu_1\rangle)$ *and* $\upsilon_{\kappa_2}=(\langle r_2,\lambda_2\rangle,\langle r_2,\mu_2\rangle)$ *be two PyCNFNs, where* $t_1=[t_1,s_1], t_1=[t_1,\tilde{s}_1]$, *and* $\beta(\varrho_{\kappa_1})$ *and* $\beta(\varrho_{\kappa_2})$ *be the accuracy degrees of* υ_{κ_1} *and* υ_{κ_2}, *respectively. Then,*

1. *If* $\beta(\upsilon_{\kappa_2})=\beta(\upsilon_{\kappa_1})$, *then* $\upsilon_{\kappa_2}\sim\upsilon_{\kappa_1}$.
2. *If* $\beta(\upsilon_{\kappa_2})=\beta(\upsilon_{\kappa_1})$, *then* $\upsilon_{\kappa_2}>\upsilon_{\kappa_1}$.
3. *If* $\beta(\upsilon_{\kappa_2})=\beta(\upsilon_{\kappa_1})$, *then* $\upsilon_{\kappa_2}<\upsilon_{\kappa_1}$.

Pythagorean cubic normal fuzzy aggregation operators

Here, we present several AOs for PyCNFNs along with their corresponding properties.

Definition 14: *Let* $\upsilon_{\kappa i}=\langle\beta_i,£_i\rangle[\langle\delta_{ij},\nu_{ij}\rangle,\langle\breve{\delta}_{ij},\varrho_{ij}\rangle]$ $(i = 1, 2, 3, \ldots, m)$ *be the set of all PyCNFNs, where* $w = (w_1, w_2, \ldots, w_\alpha)^T$ *be the WeV of* υ_i, $w_i\geq 0$, *with* $\sum_{i=1}^m w_i=1$ *and* $w_i\in[0, 1]$ *Then, Pythagorean cubic normal fuzzy weighted averaging (PyCNFWA) operator is a mapping PyCNFWA:PyCNFWA* $\rightarrow$ *PyCNFN given by,*

$$PyCNFWA(f_{\kappa_1},f_{\kappa_2},\ldots,f_{\kappa_m})=w_1f_{\kappa_1}\oplus w_2f_{\kappa_2},\ldots,w_\alpha f_{\kappa_m}.$$

Definition 15: *Let* $\upsilon_{\kappa i}=\langle\beta_i,£_i\rangle[\langle\delta_{ij},\nu_{ij}\rangle,\langle\breve{\delta}_{ij},\varrho_{ij}\rangle]$ $(i = 1, 2, 3, \ldots, m)$ *be the set of all PyCNFNs, where* $w = (w_1, w_2, \ldots w_\alpha)^T$ *be the WeV of* $\upsilon_i, w_i\geq 0, \sum_{i=1}^m w_i=1$ *and* $w_i\in[0, 1]$. *Then, Pythagorean cubic normal fuzzy weighted geometric (PyCNFWG) operator is a mapping:PyCNFWG* $\rightarrow$ *PCNFN given by,*

$$PyCNFWG(f_{\kappa_1},f_{\kappa_2},\ldots,f_{\kappa_m})=f_{\kappa_1}^{w_1}\otimes f_{\kappa_2}^{w_2},\ldots,f_{\kappa_m}^{w_m}.$$

Theorem 1: *Assume that* $\upsilon_{\kappa i}=\langle\beta_i,£_i\rangle[\langle\delta_{ij},\nu_{ij}\rangle,\langle\breve{\delta}_{ij},\varrho_{ij}\rangle]$ $(i = 1, 2, \ldots, m)$ *be the set of all PyCNFNs, while* $W = (w_1, w_2, \ldots, w_n)^T$ *be the WeV of* $\upsilon_i w_i\geq 0, \sum_{i=1}^n w_i=1$ *and* $w_i\in[0, 1]$. *Then, the aggregation result utilizing the PyCNFWA/PyCNFWG operator is also a PyCNFN,*

$$PyCNFWA(f_{\kappa_1},f_{\kappa_2},\ldots,f_{\kappa_m})=\left\{\begin{array}{l}\left(\sum_{i=1}^m w_i\beta_i\sum_{i=1}^m w_i£_i\right)\left(\left[\sqrt{1-\Pi_{i=1}^m(1-t_i^2)^{v_i}},\sqrt{1-\Pi_{i=1}^m(1-s_i^2)^{v_i}}\right],\right.\\ \left.\sqrt{1-\Pi_{i=1}^m(1-\lambda_i^2)^{v_i}}\right),\left(\left[\Pi_{i=1}^m\tilde{r}_i^{v_i},\Pi_{i=1}^m\tilde{s}_i^{v_i}\right],\Pi_{i=1}^m\mu_i^{v_i}\right)\end{array}\right\}$$

and

$$PyCNFWG(f_{\kappa_1},f_{\kappa_2},\ldots,f_{\kappa_m})=\left\{\begin{array}{l}\left(\prod_{i=1}^m\beta_i^{w_i},\prod_{i=1}^m£_i^{w_i}\right)\left(\left[\Pi_{i=1}^m r_i^{v_i},\Pi_{i=1}^m s_i^{v_i}\right],\Pi_{i=1}^m\lambda_i^{v_i}\right)\\ \left(\left[\sqrt{1-\Pi_{i=1}^m(1-t_i^2)^{v_i}},\sqrt{1-\Pi_{i=1}^m(1-s_i^2)^{v_i}}\right],\sqrt{1-\Pi_{i=1}^m(1-\mu_i^2)^{v_i}}\right)\end{array}\right\}.$$

Theorem 2: *Assume that* $\upsilon_{\kappa i}=\langle\beta_i,£_i\rangle[\langle\delta_{ij},\nu_{ij}\rangle,\langle\breve{\delta}_{ij},\varrho_{ij}\rangle]$ $(i = 1, 2, \ldots, m)$ *be the collection of all PyCNFNs and* $w=(w_1,w_2,\ldots,w_m)^T$ *be the WeV of* $\upsilon_{\kappa i}$ *where* $w_i\in[0, 1]$ *and* $\sum_{i=1}^m w_i=1$. *Then, we have*

1. Idempotency: If all $\upsilon_{\kappa_i}\langle\beta_i,£_i\rangle[\langle\delta_{ij},\nu_{ij}\rangle,\langle\breve{\delta}_{ij},\varrho_{ij}\rangle]$ $(i = 1, 2, \ldots, m)$ are equal with $\upsilon_{\kappa i}=\delta_{ij}$. Then

$$PyCNFWA/PyCNFWG(f_{\kappa_1},f_{\kappa_2},\ldots,f_{\kappa_m})=\upsilon_\kappa.$$

2. Boundary: $\upsilon_{\kappa\min}\leq PyCNFWA/PyCNFWG(f_{\kappa_1},f_{\kappa_2},\ldots,f_{\kappa_m})$ $\leq\upsilon_{\kappa\max}$ for all w, where $\upsilon_{\kappa\min}=(\min(\nu_{ij})\cdot\max(\varrho_{ij})),\upsilon_{\kappa\max}=(\max(\varrho_{ij})\cdot\min(\nu_{ij}))$.
3. Monotonicity: $\upsilon_{\kappa_i}=\langle\beta_i,£_i\rangle[\langle\delta_{ij},\nu_{ij}\rangle,\langle\breve{\delta}_{ij},\varrho_{ij}\rangle]$ and $\upsilon_{\kappa_i}=\langle\beta_i,£_i\rangle[\langle\delta_{ij},\nu_{ij}\rangle,\langle\breve{\delta}_{ij},\varrho_{ij}\rangle]$ $(i = 1, 2, \ldots, n)$ be the group of Pythagorean cubic normal fuzzy values. If $\upsilon_{\kappa i}\leq\upsilon_{\kappa i}$ for all i.

Then,

$$PyCNFWA/PyCNFWG(f_{\kappa_1},f_{\kappa_2},\ldots,f_{\kappa_m})\leq PyCNFWA/PyCNFWG(\upsilon_{\kappa_1},\upsilon_{\kappa_2},\ldots,\upsilon_{\kappa_m})$$

for every w. Similarly for PyCNFWG operators.

Lemma 1: *Let* $\upsilon_{\kappa i}>0$, $w_i>0$ $(i = 1, 2, \ldots, m)$ *and* $\sum_{i=1}^m w_i=1$ *Then*

$$\prod_{i=1}^m(\upsilon_{\kappa i})^{w_i}\leq\sum_{i=1}^m w_i\upsilon_{\kappa i}$$

where the equality holds if and only if

$$(f_{\kappa_1},=f_{\kappa_2},=f_3,\ldots,=f_{nm}).$$

Theorem 3: *Let* $\upsilon_{\kappa i}=\langle\beta_i,£_i\rangle[\langle\delta_{ij},\nu_{ij}\rangle,\langle\breve{\delta}_{ij},\varrho_{ij}\rangle]$ $(i = 1, 2, \ldots, m)$ *be a collection of all PyCNFNs, then*

$$PyCNFWG(\upsilon_{\kappa_1},\upsilon_{\kappa_2},\ldots,\upsilon_{\kappa_m})\leq PyCNFWA(f_{\kappa_1},f_{\kappa_2},\ldots,f_{\kappa_m}),$$

where $w = (w_1, w_2, w_3, \ldots w_n)$ is the weighted vector of $\upsilon_{\kappa i}(i=1,2,\ldots,m)$ $w_i\in[0,1], \sum_{i=1}^n w_i=1$.

Proof. Its proof based on the above lemma. □

Definition 16: *Assume that* $\upsilon_{\kappa i}=\langle\beta_i,£_i\rangle[\langle\delta_{ij},\nu_{ij}\rangle,\langle\breve{\delta}_{ij},\varrho_{ij}\rangle]$ $(i = 1, 2, \ldots, m)$ *be the collection of all PyCNFNs and* $w = (w_1, w_2, \ldots, w_n)^T$ *be the WeV of* $\upsilon_{\kappa i}$ $(i = 1, 2, \ldots, m)$ *with* $w_i\geq 0$ $(i = 1, 2, \ldots, m)$ *wherever* $w_i\in[0, 1]$ *and* $\sum_{i=1}^n w_i=1$. *Pythagorean cubic fuzzy normal ordered weighted averaging (PyCNFOWA) operator is a mapping PyCNFOWA:PyCNFN*^{m} $\rightarrow$ *PyCNFN, defined by*

$$PyCNFOWA(f_{\kappa_1},f_{\kappa_2},\ldots,f_{\kappa_m})=(w_1f_{\kappa_1}\oplus w_2f_{\kappa_2},\ldots,w_mf_{\kappa_m}).$$

Definition 17: *Let* $\upsilon_{\kappa i}=\langle\beta_i,£_i\rangle[\langle\delta_{ij},\nu_{ij}\rangle,\langle\breve{\delta}_{ij},\varrho_{ij}\rangle]$ $(i = 1, 2, \ldots, m)$ *be the group of all PyCNFNs and* $w = (w_1, w_2, \ldots w_n)^T$ *be the WeV of* $\upsilon_{\kappa i}$ $(i = 1, 2, \ldots, m)$ *with* $w_i\geq 0$ $(i = 1, 2, \ldots, m)$ *where* $w_i\in[0, 1]$ *and* $\sum_{i=1}^n w_i=1$. *Then, Pythagorean cubic normal fuzzy order weighted geometric (PyCNFOWG) operator is a mapping*

$$PyCNFOWG\left(f_{\kappa_1}, f_{\kappa_2}, \ldots, f_{\kappa_m}\right) = \left(f_{\kappa_1}^{w_1} \otimes f_{\kappa_2}^{w_2}, \ldots, f_{\kappa_m}^{w_n}\right).$$

and the PyCNFOWG operator is said to be Pythagorean cubic normal fuzzy ordered weighted geometric operator.

Theorem 4: *Let* $\upsilon_{\kappa} = \langle \beta_i, £_i \rangle \left[\langle \delta_{ij}, \nu_{ij} \rangle, \langle \breve{\delta}_{ij}, \varrho_{ij} \rangle\right]$ (*i* = 1, 2, …, *m*) *be the group of all PyCNFNs and* $w = (w_1, w_2, \ldots w_n)^T$ *be the WeV of* $\upsilon_{\kappa i}$ (*i* = 1, 2, …, *m*) *with* $w_i \geq 0$ (*i* = 1, 2, …, *m*) (*i* = 1, 2, …, *m*) *where* $w_i \in [0, 1]$ *and* $\sum_{i=1}^{n} w_i = 1$. *Then, the aggregation consequence using PyCNFOWG function is also a PyCNFN and*

$$PyCNFOWA\left(f_{\kappa_1}, f_{\kappa_2}, \ldots, f_{\kappa_m}\right) = \left\{ \begin{array}{l} \left(\sum_{i=1}^{m} w_i \beta_i, \sum_{i=1}^{m} w_i £_i\right)\left(\left[\sqrt{1-\Pi_{i=1}^{m}\left(1-t_i^2\right)^{v_i}},\right.\right. \\ \left.\sqrt{1-\Pi_{i=1}^{m}\left(1-s_i^2\right)^{v_i}}\right], \left.\sqrt{1-\Pi_{i=1}^{m}\left(1-\lambda_i^2\right)^{v_i}}\right), \\ \left(\left[\Pi_{i=1}^{m} \tilde{r}_i^{v_i}, \Pi_{i=1}^{\alpha} \tilde{s}_i^{v_i}\right], \Pi_{i=1}^{m} \mu_i^{v_i}\right) \end{array} \right\},$$

And the aggregation consequence using the PyCNFOWG operator is also a PyCNFN and

$$PyCNFOWG\left(f_{\kappa_1}, f_{\kappa_2}, \ldots, f_{\kappa_m}\right) = \left\{ \begin{array}{l} \left(\prod_{i=1}^{m} \beta_i^{w_i}, \prod_{i=1}^{m} £_i^{w_i}\right)\left(\left[\Pi_{i=1}^{m} r_i^{v_i}, \Pi_{i=1}^{m} s_i^{v_i}\right], \Pi_{i=1}^{m} \lambda_i^{v_i}\right) \\ \left(\left[\sqrt{1-\Pi_{i=1}^{m}\left(1-t_i^2\right)^{v_i}}, \sqrt{1-\Pi_{i=1}^{m}\left(1-s_i^2\right)^{v_i}}\right],\right. \\ \left.\sqrt{1-\Pi_{i=1}^{m}\left(1-\mu_i^2\right)^{v_i}}\right) \end{array} \right\}.$$

Theorem 5: *Let* $\upsilon_{\kappa} = \langle \beta_i, £_i \rangle \left[\langle \delta_{ij}, \nu_{ij} \rangle, \langle \breve{\delta}_{ij}, \varrho_{ij} \rangle\right]$ (*i* = 1, 2, …, *m*) *be collection of all PyCNFNs and* $w = (w_1, w_2, \ldots w_m)^T$ *be the WeV of* $\upsilon_{\kappa i}$ (*i* = 1, 2, …, *n*) *along* $w_i \geq 0$ (*i* = 1, 2, …, *m*) *wherever* $w_i \in [0, 1]$ *and* $\sum_{i=1}^{m} w_i = 1$ *then we have*

1. Idempotency: If all $\upsilon_{\kappa i} = \langle \langle \beta_i, £_i \rangle, \langle \delta_i, \nu_i \rangle, \langle \breve{\delta}_i, \varrho_i \rangle \rangle$, (*i* = 1, 2, …, *m*) are equal with $\upsilon_{\kappa i} = \delta_i$ then *PCNFWA*/*PyCNFWG* $\left(f_{\kappa_1}, f_{\kappa_2}, \ldots, f_{\kappa_m}\right) = \upsilon_{\kappa}$ for all *i*. Then,

$$PyCNFOWA/PyCNFOWG\left(f_{1\kappa_1}, f_{1\kappa_2}, \ldots, f_{1\kappa_m}\right) = \upsilon_{\kappa_i}$$

2. Boundedness: $\upsilon_{\kappa\min} \leq PyCNFOWA/PyCNFOWG\left(f_{\kappa_1}, f_{\kappa_2}, \ldots, f_{\kappa_m}\right) \leq$, $\upsilon_{\kappa\max}$ for all *w* where $\upsilon_{\kappa\min} = \left(\min_j\left(\nu_i\right), \max_j\left(\varrho_i\right)\right), \upsilon_{\kappa\max} = \left(\max_j\left(\varrho_i i\right), \min_j\left(\nu_i\right)\right)$.
3. Monotonicity: $\upsilon_{\kappa_i} = \langle \beta_i, £_i \rangle \left[\langle \delta_i, \nu_i \rangle, \langle \breve{\delta}_i, \varrho_i \rangle\right]$ and $\upsilon_{\kappa i} = \langle \beta_i, £_i \rangle \left[\langle \delta_i, \nu_i \rangle, \langle \breve{\delta}_i ; \varrho_i \rangle\right]$ (*i* = 1, 2, …, *n*) be the group of Pythagorean cubic normal fuzzy values, if $\upsilon_{\kappa i} \leq \upsilon_{\kappa i}$ for all *i*.

Then

$$PyCNFOWA/PyCNFOWG\left(f_{\kappa_1}, f_{\kappa_2}, \ldots, f_{\kappa_n}\right) \leq PyCNFWA/PyCNFWG\left(\upsilon_{\kappa_1}, \upsilon_{\kappa_2}, \ldots, \upsilon_{\kappa_n}\right)$$

for every *w*.

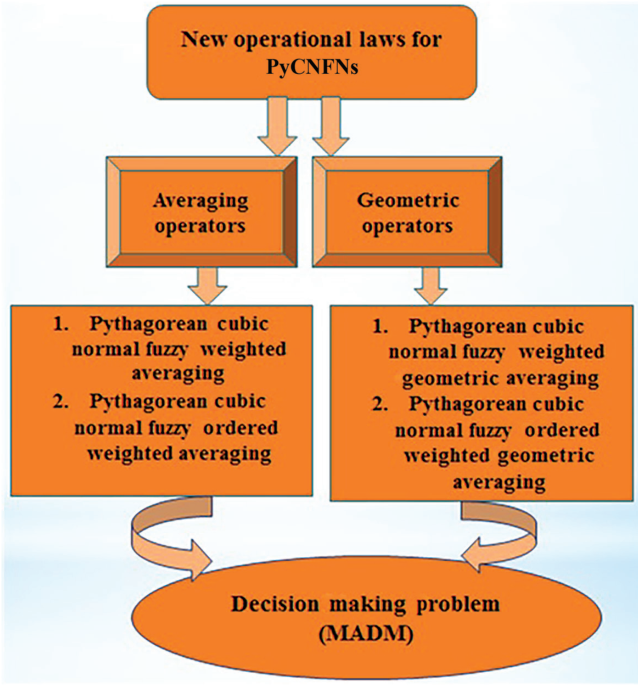

Figure 2: Operational laws. Abbreviations: MCDM, multiple-criteria decision-making; PyCNFNs, Pythagorean cubic normal fuzzy numbers.

Lemma 2: *Let* $\upsilon_{\kappa i} > 0$, $w_i > 0$ (*i* = 1, 2, …, *n*) *and* $\sum_{i=1}^{n} w_i = 1$. *Then* $\prod_{i=1}^{n}\left(\upsilon_{\kappa i}\right)^{w_i} \leq \sum_{i=1}^{n} w_i \upsilon_{\kappa i}$ *where the quantity hold if and only if* $\upsilon_{\kappa i} = \left(f_{\kappa_1}, f_{\kappa_2}, \ldots, f_{\kappa_n}\right)$.

Theorem 6: *Let* $\upsilon_{\kappa i} = \langle \beta_i, £_i \rangle \left[\langle \delta_i, \nu_i \rangle, \langle \breve{\delta}_i, \varrho_i \rangle\right]$ (*i* = 1, 2, …, *n*) *be the collection of all PyCNFN. Then,*

$$PyCNFOWG\left(f_{\kappa_1}, f_{\kappa_2}, \ldots, f_{\kappa_n}\right) \leq PyCNFOWA\left(f_{\kappa_1}, f_{\kappa_2}, \ldots, f_{\kappa_n}\right).$$

Wherever $w = (w_1, w_2, \ldots w_n)^T$ is the weighted vector of $\upsilon_{\kappa i}$ (*i* = 1, 2, …, *n*) ∃ $w_i \in [0, 1]$ (*i* = 1, 2, …, *n*) and $\sum_{i=1}^{n} w_i = 1$.. The flowcharts of all these operational laws and AOs is given in figure 2.

MULTICRITERIA DECISION-MAKING ALGORITHM

Here, we present the multicriteria decision-making algorithm based on the developed Pythagorean cubic normal fuzzy AOs.

Suppose there are ϱ_{κ} alternatives $\varrho = \left(\varrho_{\kappa_1}, \varrho_{\kappa_2}, \ldots, \varrho_{\kappa_i}\right)$ to be assessed on the basis of *κ* criteria $Y = (Y_1, Y_2, \ldots, Y_j)$, having WeV $w = (w_1, w_2, \ldots w)^T$ ∃ $w_i \in [0, 1]$ and $\sum_{i=1}^{n} w_j = 1$. To assess the accomplishment of alternatives m_i with respect to the criteria Y_j, the decision-makers are required to give information about the alternative ϱ_i which satisfies the criteria Y_j also about those alternatives which do not satisfy the criteria Y_j.

These two components can be expressed by $\langle \delta_{ij}, \nu_{ij} \rangle$ and $\langle \breve{\delta}_{ij}, \varrho_{ij} \rangle$ which represent the degree that the alternatives ϱ_j

satisfy the criteria Y_j and degree that the alternatives m_i do not satisfy the criteria Y_j as it may be presented by a PCNFN. Then the decision matrix $\upsilon_{\kappa ij} = \left[\langle \beta_{ij}, £_{ij}\rangle, \langle\langle \delta_{ij}, \nu_{ij}\rangle, \langle \breve{\delta}_{ij}, \varrho_{ij}\rangle\rangle\right]$, $(i = 1, 2, \ldots, m)$, $(j = 1, 2, \ldots, n)$ presents the whole data presented by the decision-maker. Further steps of the algorithm are given below.

Step 1: Compose the Pythagorean cubic normal fuzzy decision matrix $\upsilon_{\kappa i} = \left[\langle \beta_i, £_i\rangle, \langle\langle \delta_{ij}, \nu_{ij}\rangle, \langle \breve{\delta}_{ij}, \varrho_{ij}\rangle\rangle\right]_{m\times n}$. Pythagorean cubic normal decision matrix can be transformed into the organized Pythagorean normal fuzzy decision matrix as follows:

$$D_z = \left[\langle \beta_{ij}, £_{ij}\rangle, \langle\langle \delta_{ij}, \nu_{ij}\rangle, \langle \breve{\delta}_{ij}, \varrho_{ij}\rangle\rangle\right]_{m\times n}.$$

where

$$\beta_{\kappa ij} = \begin{Bmatrix} Y_{\kappa i} \text{ If the attribute is of benefit type} \\ Y_{\kappa i}^{\kappa} \text{ If the attribute is of cost type} \end{Bmatrix},$$

and

$$Y_{\kappa i}^{\kappa} = \langle \beta_{ij}, £_{ij}\rangle, \langle\langle \breve{\delta}_{ij}, \varrho_{ij}\rangle, \langle \delta_{ij}, \nu_{ij}\rangle\rangle.$$

If all criteria are uniform in type, then normalization of the decision matrix is unnecessary.

Step 2: Utilize the proposed aggregation functions to get the aggregated values for the alternatives $\varrho_{\kappa i}$, i.e. the developed operator to stem the whole preference values $\varrho_{\kappa i}$ $(i = 1, 2, \ldots, m)$ of the alternative $\varrho_{\kappa i}$ wherever $w = (w_1, w, \ldots, w_\alpha)^T$ is the weighting vector of the criteria.

Step 3: We assess the scores $(\varrho_{\kappa i})$ $(i = 1, 2, \ldots, m)$ and the divergence degree $\beta(\varrho_{\kappa i})$ of the whole values $\varrho_{\kappa i}$.

Step 4: Order all the alternatives according to the score values and choose the best one.

THE TODIM METHOD

Gomes and Lema initially proposed the TODIM method in the 1990s (Lourenzutti et al., 2017). The steps of TODIM method are given below.

Step 1: In the TODIM method, first define the decision matrix $\upsilon = [f_{i\kappa}]_{m\times n}$ $(i = 1, 2, 3, \ldots, m)$, $(j = 1, \ldots, n)$, where evaluations of alternatives $\varrho_{i\kappa}$ corresponding to criteria Y_j are represented by the decision-makers in the form of PyCNFNs, as follows:

Decision Matrix

$$\upsilon = \begin{array}{lccc} \text{alternatives} & Y_1 & \cdot & Y_n \\ \varrho_{\kappa_1} & f_{11} & \cdot & f_{1n} \\ \varrho_{\kappa_2} & \cdot & \cdot & \cdot \\ \varrho_{\kappa_3} & \cdot & \cdot & \cdot \\ \varrho_{\kappa_4} & \cdot & \cdot & \cdot \\ \varrho_{\kappa_5} & f_{m1} & \cdot & f_{mn} \end{array}$$

Step 2: Normalizing the above decision matrix $\upsilon_{ij} = [f_{\kappa ij}]_{m\times n}$.

Step 3: Assume that $w = (w_1, w_2, \ldots w_n)$ be the WeV of the criteria $(\varrho_{\kappa_1}, \varrho_{\kappa_2}, \ldots, \varrho_{\kappa_n})$, where $0 \le w_i \le 1$ and $\sum_{i=1}^{n} w_i = 1$. It is essential that the decision maker (DM) defines a reference criterion ϱ_κ, $1 \le r \le m$, ordinarily the reference criterion w_i is the upmost weight. Compute the relative weights,

$$w_\kappa = \frac{w_\kappa}{w_m}$$

where ϱ_κ is a generate criterion.

Step 4: Find the dominance degree of ϱ_κ corresponding to Y_l. The term $\varpi_\kappa(\varrho_m, Y_n)$ describes the partial dominance θ is the attenuation part of the losses, and the option of θ has an influence on the form of the possibility value function.

$$\tau(\varrho_m, Y_n) = \sum_{i=1}^{m} \varpi_\kappa(\varrho_m, Y_n)\,(m, n = 1, 2, \ldots, k).$$

where

$$\varpi_\kappa(\varrho_m, Y_n) = \begin{cases} \sqrt{w_{r\kappa}(\kappa_{m\kappa} - \kappa_{n\kappa}) / \Sigma_{\kappa=1}^{m} w_{r\kappa}} \\ 0 \\ -\frac{1}{\theta}\sqrt{(\kappa_{m\kappa} - \kappa_{m\kappa})(\Sigma_{\kappa=1}^{m} w_{r\kappa}) / w_{r\kappa}} \end{cases},$$

$$(\kappa_{m_\kappa}\ \kappa_{n_\kappa} = 0), (\kappa_{m_\kappa} - \kappa_{m_\kappa} < 0).$$

Step 5: Also normalize the dominance values.

$$\in_i = \frac{\Sigma_{m=1}^{m} \varpi\kappa(\varrho_m, Y_m) - \min \Sigma_{m=1}^{m} \varpi_\kappa(\varrho_m, Y_n)}{\max \Sigma_{m=1}^{m} \varpi_\kappa(\varrho_m, Y_m) - \min \Sigma_{m=1}^{m} \varpi_\kappa(\varrho_m, Y_n)}.$$

Step 6: In the final step, we categorize the alternatives corresponding to the values of ε_i. The flow chart of TODIM Method is given in figure 3.

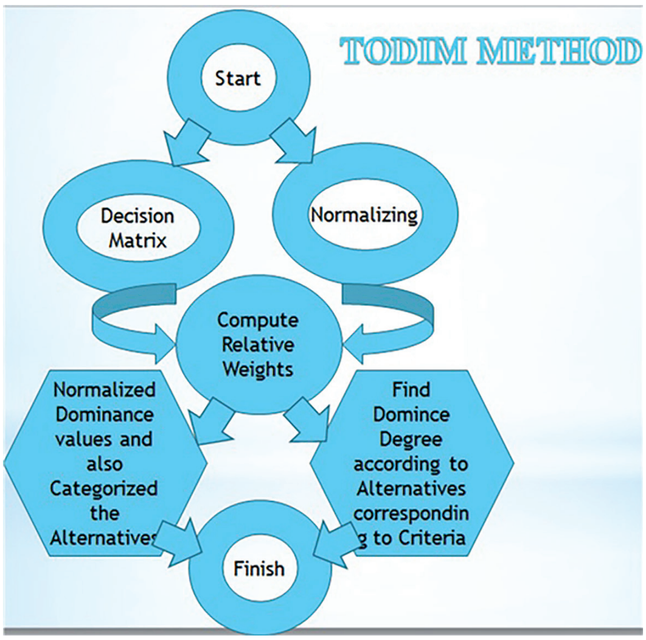

Figure 3: TODIM method algorithm.

ILLUSTRATIVE EXAMPLE

The special education department of a school is responsible for assessing the educational needs of students with disabilities. They will evaluate each assessment method based on the four criteria, i.e. validity, individualization, reliability, and feasibility.

The assessment method should accurately measure the students' abilities and needs in alignment with their individual disabilities. Valid assessments ensure that appropriate educational interventions and supports are provided. It is essential that the assessment method allows for individualization to address the unique needs and strengths of each student with a disability. Individualized assessments ensure personalized educational plans that maximize student learning and development. Reliable assessments produce consistent results when administered multiple times or by different assessors. Reliability ensures that decisions regarding educational programming for students with disabilities are based on dependable data. The assessment method should be practical and feasible to implement within the resources and constraints of the school district. Feasibility considerations include factors such as time, cost, personal, and ease of administration. Five alternatives are selected for evaluation and also we can denote the alternative set is $\varrho_{\kappa} = \left\{\varrho_{\kappa_1}, \varrho_{\kappa_2}, \varrho_{\kappa_3}, \varrho_{\kappa_4}, \varrho_{\kappa_5}\right\}$ and four criteria are considered the details of which are as follows,

Standardized testing: validity, individualization, reliability, feasibility
Provides standardized measures of academic achievement but may not fully capture students' abilities and needs related to their disabilities. Limited individualization as it follows a standardized format. Generally, reliable in terms of consistency but may not account for fluctuations in students' performance. Relatively straightforward to administer but may not accommodate students with severe disabilities or diverse needs.

Observational assessments: validity, individualization, reliability, feasibility
Offers insight into students' behavior, interactions, and learning styles, providing a holistic view of their abilities and needs. Allows for individualized observation and documentation of students' strengths and challenges. Reliability depends on the consistency and objectivity of observations but can be enhanced through structured protocols. Requires trained observers and dedicated observation time but can be implemented in various educational settings.

Portfolio assessment: validity, individualization, reliability, feasibility
Provides authentic evidence of students' progress and achievements over time, including work samples, projects, and performance assessments. Highly individualized, allowing students to showcase their unique abilities and accomplishments. Reliability may vary depending on the consistency and standardization of portfolio assessment procedures. Requires ongoing documentation and organization of student work but offers flexibility and authenticity in assessment.

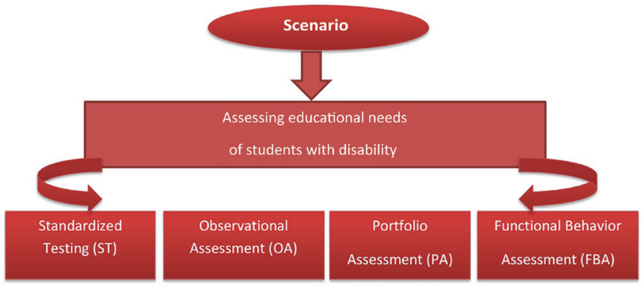

Figure 4: Illustrative example.

Functional behavior assessment: validity, individualization, reliability, feasibility
Focuses on understanding the function of students' behavior and identifying interventions to support their learning and social-emotional needs. Tailored to each student's behavior patterns and environmental factors, facilitating personalized behavior support plans. Reliability depends on the rigor of data collection methods and the expertise of behavior analysts. Requires specialized training and expertise but is essential for addressing challenging behaviors and promoting positive outcomes. Consisting the criteria set $Y = \{Y_1, Y_2, Y_3, Y_4\}$, all are benefits and their corresponding weights are $W = \{0.3, 0.25, 0.25, 0.2\}$. After carefully evaluating each assessment method based on these criteria, the special education department will select the most suitable methods or combination of methods to effectively assess the educational needs of students with disabilities, ensuring that appropriate interventions and supports are provided to promote their success in school. This example is also explained by Figure 4.

In this example, the school district uses four distinct criteria to evaluate each assessment method, emphasizing the importance of validity, individualization, reliability, and feasibility in assessing the educational needs of students with disabilities. Suppose that $w = (0.3, 0.35, 0.25)$ be the experts WeV. The experts informations in the form of PyCNFNs are given in the accompanying Tables 3-5.

PyCNFWA operator

Step 1: Tables 3-5 show the informations given by decision-makers.
Step 2: By using the PyCNFWA operator to assess the overall consequences of each alternative, considering the expert weight, $w = (0.3, 0.35, 0.35)$. The result is shown in Table 6.
Step 3: Since all four alternatives are benefit type, normalization is unnecessary.
Step 4: To obtain the other aggregated results, we used the attribute weights.
Step 5: Based on $\omega = (0.3, 0.25, 0.25, 0.2)$ and using the same PyCNFWA operator on the data present in 6, the whole benefits of each alternative is obtained,

Table 3: First decision-maker's information.

	Y_1	Y_2	Y_3	Y_4
ϱ_{κ_1}	$\langle(0.8,9)\rangle$, $([0.5,0.7];0.5)$ $([0.9,0.4];0.6)$	$\langle(0.5,4)\rangle$, $([0.5,0.7];0.4)$ $([0.6,0.3];0.7)$	$\langle(7,80)\rangle$, $([0.5,0.3];0.7)$ $([0.7,0.8];0.4)$	$\langle(4,50)\rangle$, $([0.8,0.4];0.6)$ $([0.3,0.7];0.5)$
ϱ_{κ_1}	$\langle(0.7,8)\rangle$, $([0.4,0.7];0.4)$ $([0.5,0.5];0.6)$	$\langle(0.6,7)\rangle$, $([0.8,0.7];0.7)$ $([0.3,0.4];0.4)$	$\langle(8,90)\rangle$, $([0.4,0.4];0.5)$ $([0.5,0.7];0.6)$	$\langle(5,45)\rangle$, $([0.5,0.3];0.5)$ $([0.6,0.7];0.6)$
ϱ_{κ_1}	$\langle(0.6,6)\rangle$, $([0.4,0.4];0.4)$ $([0.5,0.7];0.7)$	$\langle(0.8,9)\rangle$, $([0.5,0.7];0.5)$ $([0.6,0.4];0.6)$	$\langle(6.5,75)\rangle$, $([0.5,0.6];0.5)$ $([0.6,0.5];0.6)$	$\langle(6,60)\rangle$, $([0.5,0.6];0.4)$ $([0.6,0.5];0.6)$
ϱ_{κ_1}	$\langle(0.6,7)\rangle$, $([0.5,0.4];0.5)$ $([0.6,0.7];0.6)$	$\langle(0.7,8)\rangle$, $([0.5,0.4];0.4)$ $([0.6,0.7];0.7)$	$\langle(7.5,85)\rangle$, $([0.4,0.5];0.4)$ $([0.5,0.6];0.7)$	$\langle(7,70)\rangle$, $([0.7,0.7];0.4)$ $([0.4,0.4];0.7)$
ϱ_{κ_1}	$\langle(0.5,6)\rangle$, $([0.3,0.4];0.6)$ $([0.8,0.7];0.5)$	$\langle(0.6,7)\rangle$, $([0.5,0.5];0.5)$ $([0.5,0.6];0.6)$	$\langle(6,95)\rangle$, $([0.5,0.5];0.4)$ $([0.6,0.7];0.7)$	$\langle(6.5,65)\rangle$, $([0.5,0.4];0.4)$ $([0.6,0.7];0.7)$

Table 4: Information given by second decision-maker.

	Y_1	Y_2	Y_3	Y_4
ϱ_{κ_1}	$\langle(.7,.6)\rangle$, $([.6,.8];.6)$ $([.8,.6];.7)$	$\langle(.4,.2)\rangle$, $([.6,.8];.5)$ $([.8,.6];.8)$	$\langle(.25,.15)\rangle$, $([.6,.4];.6)$ $([.8,.9];.5)$	$\langle(.75,.6)\rangle$, $([.9,.5];.7)$ $([.4,.8];.6)$
ϱ_{κ_1}	$\langle(.9,.7)\rangle$, $([.5,.8];.5)$ $([.7,.6];.8)$	$\langle(.45,.4)\rangle$, $([.9,.9];.8)$ $([.4,.5];.5)$	$\langle(.55,.4)\rangle$, $([.6,.5];.6)$ $([.7,.8];.7)$	$\langle(5,45)\rangle$, $([.5,.3];.5)$ $([.6,.7];.6)$
ϱ_{κ_1}	$\langle(.55,.5)\rangle$, $([.7,.5];.5)$ $([.6,.8];.8)$	$\langle(.75,.7)\rangle$, $([.6,.8];.6)$ $([.7,.5];.7)$	$\langle(.65,.6)\rangle$, $([.6,.7];.6)$ $([.8,.8];.7)$	$\langle(.5,.3)\rangle$, $([.6,.7];.6)$ $([.8,.6];.7)$
ϱ_{κ_1}	$\langle(.8,.7)\rangle$, $([.6,.5];.5)$ $([.8,.8];.7)$	$\langle(.6,.5)\rangle$, $([.6,.5];.5)$ $([.7,.8];.8)$	$\langle(.6,.5)\rangle$, $([.6,.7];.5)$ $([.7,.8];.8)$	$\langle(.8,.7)\rangle$, $([.8,.8];.5)$ $([.5,.5];.8)$
ϱ_{κ_1}	$\langle(.4,.3)\rangle$, $([.4,.5];.7)$ $([.9,.8];.6)$	$\langle(.55,.5)\rangle$, $([.6,.6];.6)$ $([.8,.7];.7)$	$\langle(.7,.6)\rangle$, $([.6,.7];.5)$ $([.8,.5];.8)$	$\langle(.65,.6)\rangle$, $([.6,.5];.6)$ $([.7,.8];.8)$

$$\varrho_{\kappa_1}=\left[\langle(1.257,10.39)\rangle,\langle[0.701,0.65];0.563\rangle,\langle[0.614,0.545];0.659\rangle\right]$$

$$\varrho_{\kappa_2}=\left[\langle(1.428,11.03)\rangle,\langle[0.659,0.646];522\rangle,\langle[0.579,0.593];0.721\rangle\right]$$

$$\varrho_{\kappa_3}=\left[\langle(1.350,10.77)\rangle,\langle[0.592,0.625];0.632\rangle,\langle[0.61,0.574];0.0656\rangle\right]$$

$$\varrho_{\kappa_4}=\left[\langle(1.534,12.19)\rangle,\langle[0.634,0.554];0.549\rangle,\langle[0.585,0.625];0.661\rangle\right]$$

$$\varrho_{\kappa_5}=\left[\langle(1.294,0.396)\rangle,\langle[0.661,0.606];0.616\rangle,\langle[0.615,0.648];0.6\rangle\right]$$

Step 6: By applying explanation, we can obtained each alternative's score

$$\varrho_{\kappa_1}=0.00494,\varrho_{\kappa_2}=0.006451,\varrho_{\kappa_3}=0.000148,$$

$$\varrho_{\kappa_4}=0.000497,\varrho_{\kappa_5}=0.0001,$$

Step 7: According to $\varrho_{\kappa_2}>\varrho_{\kappa_4}>\varrho_{\kappa_1}>\varrho_{\kappa_3}>\varrho_{\kappa_5}$, the ranking order is $n_2>n_4>n_1>n_3>n_5$. Hence, n_2 is the best alternative.

Table 5: Information given by third decision-maker.

	Y_1	Y_2	Y_3	Y_4
ϱ_{κ_1}	$\langle(.4,.3)\rangle$, $([.8,.7];.4)$ $([.6,.5];.7)$	$\langle(.65,.45)\rangle$, $([.8,.7];.7)$ $([.6,.6];.6)$	$\langle(.8,.65)\rangle$, $([.7,.7];.7)$ $([.6,.5];.5)$	$\langle(.45,.3)\rangle$, $([.7,.5];.5)$ $([.6,.6];.8)$
ϱ_{κ_2}	$\langle(.35,.2)\rangle$, $([.6,.5];.9)$ $([.5,.8];.2)$	$\langle(.6,.5)\rangle$, $([.6,.5];.7)$ $([.8,.8];.4)$	$\langle(.7,.6)\rangle$, $([.6,.7];.3)$ $([.8,.4];.8)$	$\langle(.55,.5)\rangle$, $([.5,.5];.4)$ $([.8,.6];.8)$
ϱ_{κ_3}	$\langle(.6,.5)\rangle$, $([.6,.7];.7)$ $([.8,.4];.4)$	$\langle(.35,.3)\rangle$, $([.6,.7];.7)$ $([.8,.4];.4)$	$\langle(.4,.3)\rangle$, $([.9,.4];.4)$ $([.2,.7];.8)$	$\langle(.65,.6)\rangle$, $([.6,.6];.4)$ $([.7,.4];.7)$
ϱ_{κ_4}	$\langle(.4,.35)\rangle$, $([.7,.5];.6)$ $([.6,.7];.7)$	$\langle(.7,.6)\rangle$, $([.7,.5];.6)$ $([.6,.7];.7)$	$\langle(.75,.7)\rangle$, $([.6,.5];.3)$ $([.7,.7];.7)$	$\langle(.45,.4)\rangle$, $([.6,.5];.5)$ $([.8,.7];.8)$
ϱ_{κ_5}	$\langle(.7,.55)\rangle$, $([.9,.5];.5)$ $([.3,.6];.6)$	$\langle(.35,.3)\rangle$, $([.9,.5];.5)$ $([.3,.6];.6)$	$\langle(0.45,.3)\rangle$, $([.7,.6];.7)$ $([.5,.7];.4)$	$\langle(0.35,0.3)\rangle$, $([.8,.6];0.7)$ $([.6,.5];.5)$

The PyCNFWG operator

Step 1: Tables 3-5 show the informations given by decision-makers.

Step 2: By utilizing the PyCNFWG operator to get the aggregated value of each alternative, considering the expert weight w = (0.3, 0.35, 0.35). The result is given in Table 7.

Step 3: Since all four alternatives are benefits, normalization is unnecessary.

Step 4: To obtain the other aggregated result, we used the attribute weights.

Step 5: With criteria weight ω = (0.3, 0.25, 0.25, 0.2), using the same PyCNFWG operator with the data present in Table 7, we get the below aggregated values.

Table 6: Aggregated data by the PyCNFWA operator

	Y_1	Y_2	Y_3	Y_4
ϱ_{κ_1}	$\langle(1.68,17.8)\rangle$, $\langle[0.672,0.741];0.512\rangle$, $\langle[0.664,0.499];0.669\rangle$	$\langle(0.55,0.502)\rangle$, $\langle[0.672,0.741];0.568\rangle$, $\langle[0.695,0.717];0.695\rangle$	$\langle(0.477,0.322)\rangle$, $\langle[0.617,0.527];0.670\rangle$, $\langle[0.695,0.717];0.468\rangle$	$\langle(0.54,0.40)\rangle$, $\langle[0.822,0.474];0.613\rangle$, $\langle[0.423,0.695];0.629\rangle$
ϱ_{κ_2}	$\langle(1.83,20.01)\rangle$, $\langle[0.516,0.696];0.722\rangle$ $\langle[0.563,0.629];0.452\rangle$	$\langle(0.68,0.62)\rangle$, $\langle[0.811,0.767];0.741\rangle$, $\langle[0.468,0.552];0.433\rangle$	$\langle(0.636,0.513)\rangle$, $\langle[0.553,0.568];0.491\rangle$, $\langle[0.664,613];0.711\rangle$	$\langle(0.521,0.397)\rangle$, $\langle[0.539,0.499];0.512\rangle$, $\langle[0.711,0.695];0.711\rangle$
ϱ_{κ_3}	$\langle(1.85,19.78)\rangle$, $\langle[0.599,0.568];0.756\rangle$ $\langle[0.629,0.614];0.614\rangle$	$\langle(0.96,0.975)\rangle$, $\langle[0.617,0.696];0.617\rangle$, $\langle[0.634,0.527];0.550\rangle$	$\langle(0.60,0.513)\rangle$, $\langle[0.750,0.592];0.512\rangle$, $\langle[0.452,.664];0.711\rangle$	$\langle(0.481,0.401)\rangle$, $\langle[0.574,0.640];0.487\rangle$, $\langle[0.711,0.493];0.667\rangle$
ϱ_{κ_4}	$\langle(1.91,21.4)\rangle$, $\langle[0.617,0.474];0.574\rangle$ $\langle[0.664,0.134];0.669\rangle$	$\langle(1.178,1.39)\rangle$, $\langle[0.512,0.516];0.568\rangle$, $\langle[0.730,0.653];0.734\rangle$	$\langle(0.69,0.591)\rangle$, $\langle[0.553,0.565];0.474\rangle$, $\langle[0.633,0.711];0.734\rangle$	$\langle(0.551,0.489)\rangle$, $\langle[0.715,0.696];0.412\rangle$, $\langle[0.552,0.527];0.769\rangle$
ϱ_{κ_5}	$\langle(1.62,20.03)\rangle$, $\langle[0.699,0.474];0.613\rangle$ $\langle[0.592,0.695];0.569\rangle$	$\langle(0.77,0.79)\rangle$, $\langle[0.491,0.574];0.672\rangle$, $\langle[0.811,0.669];0.595\rangle$	$\langle(0.54,0.46)\rangle$, $\langle[0.617,0.617];0.568\rangle$, $\langle[0.623,0.623];0.613\rangle$	$\langle(0.458,0.359)\rangle$, $\langle[0.672,0.516];0.599\rangle$, $\langle[0.634,0653];0.653\rangle$

Table 7: Aggregated data by the PyCNFWG operator

	Y_1	Y_2	Y_3	Y_4
ϱ_{κ_1}	$\langle(1.68,17.8)\rangle$, $\langle[0.691,0.516];0.674\rangle$, $\langle[0.629,0.734];0.493\rangle$	$\langle(0.55,0.502)\rangle$, $\langle[0.691,0.538];0.715\rangle$, $\langle[0.629,0.734];0.527\rangle$	$\langle(0.477,0.322)\rangle$, $\langle[0.715,0.793];0.474\rangle$, $\langle[0.601,0.447];0.669\rangle$	$\langle(0.54,0.40)\rangle$, $\langle[0.467,0.715];0.672\rangle$, $\langle[0.796,0.468];0.595\rangle$
ϱ_{κ_2}	$\langle(1.83,20.01)\rangle$, $\langle[0.588,0.672];0.631\rangle$ $\langle[0.499,0.653];0.575\rangle$	$\langle(1.68,0.62)\rangle$, $\langle[0.611,0.633];0.439\rangle$, $\langle[0.754,0.690];0.734\rangle$	$\langle(0.636,0.513)\rangle$, $\langle[0.711,0.680];0.719\rangle$, $\langle[0.532,0.527];0.446\rangle$	$\langle(0.521,0.397)\rangle$, $\langle[0.719,0.715];0.719\rangle$, $\langle[0.533,0.458];0.493\rangle$
ϱ_{κ_3}	$\langle(1.85,19.78)\rangle$, $\langle[0.672,0.680];0.980\rangle$ $\langle[0.561,0.527];0.527\rangle$	$\langle(0.96,0.975)\rangle$, $\langle[0.640,0.568];0.592\rangle$, $\langle[0.601,0.653];0.601\rangle$	$\langle(0.60,0.513)\rangle$, $\langle[0.631,0.712];0.719\rangle$, $\langle[0.655,0.550];0.493\rangle$	$\langle(0.481,0.401)\rangle$, $\langle[0.719,0.512];0.674\rangle$, $\langle[0.569,0.634];0.461\rangle$
ϱ_{κ_4}	$\langle(1.91,21.4)\rangle$ $\langle[0.691,0.741];0.674\rangle$ $\langle[0.611,0.468];0.568\rangle$	$\langle(1.178,1.39)\rangle$, $\langle[0.784,0.696];0.741\rangle$, $\langle[0.493,0.499];0.522\rangle$	$\langle(1.69,0.591)\rangle$, $\langle[0.654,0.719];0.741\rangle$, $\langle[0.532,0.521];0.468\rangle$	$\langle(0.551,0.489)\rangle$, $\langle[0.633,0.568];0.775\rangle$, $\langle[0.695,0.653];0.392\rangle$
X_5	$\langle(1.62,20.03)\rangle$, $\langle[0.776,0.715];0.574\rangle$ $\langle[0.488,0.468];0.595\rangle$	$\langle(0.77,0.79)\rangle$, $\langle[0.825,0.674];0.613\rangle$, $\langle[0.446,0.569];0.629\rangle$	$\langle(0.54,0.46)\rangle$, $\langle[0.669,0.646];0.680\rangle$, $\langle[0.601,0.601];0.527\rangle$	$\langle(0.458,0.359)\rangle$, $\langle[0.640,0.696];0.696\rangle$, $\langle[0.629,0.499];0.561\rangle$

$$(\varrho_{\kappa_1}) = [\langle(1.257,10.39)\rangle, \langle[0.651,0.556];0.522\rangle, \langle[0.666,0.643];0.697\rangle]$$

$$(\varrho_{\kappa_2}) = [\langle(1.428,11.03)\rangle, \langle[0.593,0.595];493\rangle, \langle[0.659,0.665];0.731\rangle]$$

$$(\varrho_{\kappa_3}) = [\langle(1.350,10.77)\rangle, \langle[0.561,0.583];0.529\rangle, \langle[0.648,0.634];0.747\rangle]$$

$$(\varrho_{\kappa_4}) = [\langle(1.534,12.19)\rangle, \langle[0.57,0.521];0.53\rangle, \langle[0.653,0.671];0.685\rangle]$$

$$(\varrho_{\kappa_5}) = [\langle(1.294,0.396)\rangle, \langle[0.582,0.571];0.578\rangle, \langle[0.675,0.701];0.639\rangle]$$

Step 6: By applying explanation, we can obtained each alternative's score.

$$(\varrho_{\kappa_1}) = 0.00033, (\varrho_{\kappa_2}) = 0.00065, (\varrho_{\kappa_3}) = 0.00032,$$

$$(\varrho_{\kappa_4}) = 0.00036, (\varrho_{\kappa_5}) = 0.00024.$$

Step 7: According to $\varrho_{\kappa_2} > \varrho_{\kappa_4} > \varrho_{\kappa_1} > \varrho_{\kappa_3} > \varrho_{\kappa_5}$, the ranking order is $n_2 > n_4 > n_1 > n_3 > n_5$. Hence, n_2 is the best alternative.

The PyCNFOWA Operator

Step 1: Tables 3-5 show the informations given by decision-makers.

Step 2: By using the PyCNFOWA operator to obtain the whole consequence of each alternative, the expert weight, w = (0.3, 0.35, 0.35). The result is shown in Table 8.

Step 3: As we see that all four alternatives are benefit types, so normalization is not required.

Step 4: To obtain the other aggregated result, we used the attribute weights.

Step 5: With criteria weight ω = (0.3, 0.25, 0.25, 0.2), using the same PyCNFOWA operator with the data present in Table 8, we get the below aggregated values

$$\varrho_{\kappa_1} = [\langle(1.257,10.39)\rangle, \langle[0.712,0.652];0.573\rangle, \langle[0.614,0.545];0.659\rangle]$$

$$\varrho_{\kappa_2} = [\langle(1.428,11.03)\rangle, \langle[0.659,0.646];0.522\rangle, \langle[0.577,0.606];0.726\rangle]$$

Table 8: Aggregated information by the PyCNFOWA operator.

	Y_1	Y_2	Y_3	Y_4
ϱ_{κ_1}	$\langle(1.68,17.8)\rangle$, $\langle[0.672,0.741];0.516\rangle$, $\langle[0.664,0.452];0.734\rangle$	$\langle(0.55,0.502)\rangle$, $\langle[0.672,0.722];0.474\rangle$, $\langle[0.664,0.433];0.653\rangle$	$\langle(0.477,0.322)\rangle$, $\langle[0.822,0.512];0.696\rangle$, $\langle[0.695,0.711];0.711\rangle$	$\langle(0.54,0.40)\rangle$, $\langle[0.617,0.491];0.565\rangle$, $\langle[0.423,0.711];0.527\rangle$
ϱ_{κ_2}	$\langle(1.83,20.01)\rangle$, $\langle[0.741,0.599];0.574\rangle$ $\langle[0.488,0.634];0.734\rangle$	$\langle(0.68,0.62)\rangle$, $\langle[0.527,0.750];0.474\rangle$, $\langle[0.499,0.629];0.669\rangle$	$\langle(1.636,0.513)\rangle$, $\langle[0.741,0.617];0.568\rangle$, $\langle[0.695,0.711];0.791\rangle$	$\langle(0.521,0.397)\rangle$, $\langle[0.474,0.574];0.412\rangle$, $\langle[0.711,0.452];0.734\rangle$
ϱ_{κ_3}	$\langle(1.85,19.78)\rangle$, $\langle[0.670,0.592];0.617\rangle$ $\langle[0.695,0.527];0.811\rangle$	$\langle(0.96,0.975)\rangle$, $\langle[0.568,0.696];0.491\rangle$, $\langle[0.669,0.611];0.592\rangle$	$\langle(0.60,0.513)\rangle$, $\langle[0.512,0.568];0.699\rangle$, $\langle[0.629,0.493];0.634\rangle$	$\langle(0.481,0.401)\rangle$, $\langle[0.613,0.640];0.672\rangle$, $\langle[0.468,0.664];0.623\rangle$
ϱ_{κ_4}	$\langle(1.91,21.4)\rangle$, $\langle[0.553,0.512];0.617\rangle$ $\langle[0.711,0.669];0.653\rangle$	$\langle(1.178,1.39)\rangle$, $\langle[0.811,0.617];0.574\rangle$, $\langle[0.664,0.711];0.623\rangle$	$\langle(0.69,0.591)\rangle$, $\langle[0.516,0.568];0.474\rangle$, $\langle[0.563,0.611];0.695\rangle$	$\langle(0.551,0.489)\rangle$, $\langle[0.539,0.487];0.516\rangle$, $\langle[0.468,0.550];0.669\rangle$
ϱ_{κ_5}	$\langle(1.62,20.03)\rangle$, $\langle[0.499,0.715];0.599\rangle$ $\langle[0.611,0.633];0.611\rangle$	$\langle(0.77,0.79)\rangle$, $\langle[0.696,0.617];0.613\rangle$, $\langle[0.552,0.730];0.595\rangle$	$\langle(0.6,0.5)\rangle$, $\langle[0.568,0.553];0.568\rangle$, $\langle[0.695,0.552];0.653\rangle$	$\langle(0.6,0.5)\rangle$, $\langle[0.661,0.512];0.672\rangle$, $\langle[0.629,0.664];0.569\rangle$

$$\varrho_{\kappa_3}=\left[\langle(1.350,10.77)\rangle,\langle[0.6,0.626];0.628\rangle,\langle[0.62,0.561];0.666\rangle\right]$$

$$\varrho_{\kappa_4}=\left[\langle(1.534,12.19)\rangle,\langle[0.636,0.551];0.556\rangle,\langle[0.604,0.634];0.659\rangle\right]$$

$$\varrho_{\kappa_5}=\left[\langle(1.294,0.396)\rangle,\langle[0.639,0.621];0.612\rangle,\langle[0.616,0.64];0.606\rangle\right]$$

Step 6: By utilizing the definition, we can obtained each alternative's score

$$\varrho_{\kappa_1}=0.00051,\left(\varrho_{\kappa_2}\right)=0.0063,\left(\varrho_{\kappa_3}\right)=0.0004,$$
$$\varrho_{\kappa_4}=0.0002,\left(\varrho_{\kappa_5}\right)=0.0001,$$

Step 7: According to $\varrho_{\kappa_2}>\varrho_{\kappa_1}>\varrho_{\kappa_3}>\varrho_{\kappa_4}>\varrho_{\kappa_5}$, the ranking order is $n_2>n_1>n_3>n_4>n_5$. Hence, n_2 is the best alternative.

The PyCNFOWG operator

Step 1: Tables 3-5 show the informations given by decision-makers.

Table 9: Aggregated data by the PyCNFOWG operator.

	Y_1	Y_2	Y_3	Y_4
ϱ_{κ_1}	$\langle(1.68,17.8)\rangle$, $\langle[0.611,0.447];0.664\rangle$, $\langle[0.715,0.794];0.474\rangle$	$\langle(0.55,0.502)\rangle$, $\langle[0.629,0.734];0.493\rangle$, $\langle[0.691,0.516];0.674\rangle$	$\langle(0.477,0.322)\rangle$, $\langle[0629.,0.734];0.527\rangle$, $\langle[0.691,0.538];0.723\rangle$	$\langle(0.54,0.40)\rangle$, $\langle[0.81,0.468];0.595\rangle$, $\langle[0.467,0.715];0.672\rangle$
ϱ_{κ_2}	$\langle(1.83,20.01)\rangle$, $\langle[0.533,0.458];0.493\rangle$ $\langle[0.719,0.715];0.719\rangle$	$\langle(0.68,0.62)\rangle$, $\langle[0.754,0.680];0.734\rangle$, $\langle[0.611,0.633];0.439\rangle$	$\langle(0.636,0.513)\rangle$, $\langle[0.532,0.527];0.446\rangle$, $\langle[0.711,0.680];0.719\rangle$	$\langle(0.521,0.397)\rangle$, $\langle[0.501,0.662];0.575\rangle$, $\langle[0.588,0.672];0.631\rangle$
ϱ_{κ_3}	$\langle(1.85,19.78)\rangle$, $\langle[0.569,0.634];0.461\rangle$ $\langle[0.719,0.512];0.674\rangle$	$\langle(0.96,0.975)\rangle$, $\langle[0.561,0.527];0.527\rangle$, $\langle[0.672,0.680];0.680\rangle$	$\langle(0.60,0.513)\rangle$, $\langle[0.656,0.550];0.550\rangle$, $\langle[0.631,0.711];0.719\rangle$	$\langle(0.481,0.401)\rangle$, $\langle[0.601,0.653];0.691\rangle$, $\langle[0.601,0.568];0.601\rangle$
ϱ_{κ_4}	$\langle(1.91,21.4)\rangle$, $\langle[0.695,0.653];0.392\rangle$ $\langle[0.633,0.568];0.775\rangle$	$\langle(1.178,1.39)\rangle$, $\langle[0.493,0.499];0.527\rangle$, $\langle[0.784,0.696];0.741\rangle$	$\langle(0.69,0.591)\rangle$, $\langle[0.601,0.468];0.569\rangle$, $\langle[0.691,0.741];0.674\rangle$	$\langle(0.551,0.489)\rangle$, $\langle[0.532,0.521];0.468\rangle$, $\langle[0.654,0.719];0.741\rangle$
ϱ_{κ_5}	$\langle(1.62,20.03)\rangle$, $\langle[0.488,0.468];0.595\rangle$ $\langle[0.776,0.715];0.574\rangle$	$\langle(0.77,0.79)\rangle$, $\langle[0.446,0.569];0.629\rangle$, $\langle[0.825,0.674];0.613\rangle$	$\langle(0.54,0.46)\rangle$, $\langle[0.629,0.499];0.561\rangle$, $\langle[0.640,0.696];0.696\rangle$	$\langle(0.458,0.359)\rangle$, $\langle[0.601,0.601];0.527\rangle$, $\langle[0.669,0.646];0.680\rangle$

Step 2: By using the PyCNFOWG operator to get the aggregated value of each alternative, considering the expert weight, w = (0.3, 0.35, 0.35). The consequences are mentioned in Table 9.

Step 3: As we see that all alternatives are benefit types, so normalization is not required.

Step 4: To obtain the other aggregated result, we used the attribute weights.

Step 5: With criteria weight ω = (0.3, 0.25, 0.25, 0.2), using the same PyCNFOWG operator with the data present in Table 9, we get the below aggregated values

$$\varrho_{\kappa_1}=\left[\langle(1.257,10.39)\rangle,\langle[0.811,0.801];0.785\rangle,\langle[0.650,0.578];0.569\rangle\right]$$

$$\varrho_{\kappa_2}=\left[\langle(1.428,11.03)\rangle,\langle[0.801,0.823];0.787\rangle,\langle[0.574,0.562];0.548\rangle\right]$$

$$\varrho_{\kappa_3}=\left[\langle(1.350,10.77)\rangle,\langle[0.818,0.780];0.818\rangle,\langle[0.594,0.587];0.511\rangle\right]$$

$$\varrho_{\kappa_4}=\left[\langle(1.534,12.19)\rangle,\langle[0.829,0.818];0.857\rangle,\langle[0.583,0.537];0.480\rangle\right]$$

$$\varrho_{\kappa_5}=\left[\langle(1.294,0.396)\rangle,\langle[0.854,0.828];0.796\rangle,\langle[0.530,0.535];0.580\rangle\right]$$

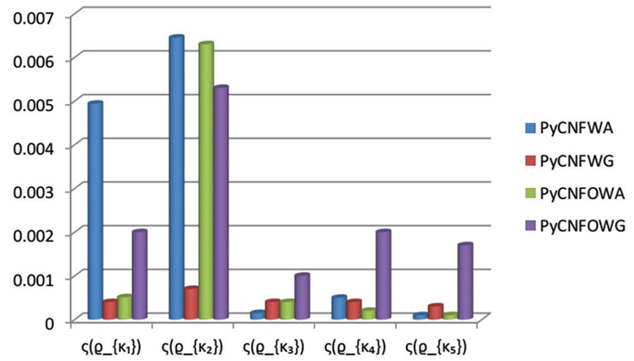

Figure 5: Ranking of the alternatives.

Table 10: Ranking.

Operators	ϱ_{κ_1}	ϱ_{κ_2}	ϱ_{κ_3}	ϱ_{κ_4}	ϱ_{κ_5}
PyCNFWA	0.00494	0.006451	0.000148	0.000497	0.0001
PyCNFWG	0.0004	0.0007	0.0004	0.0004	0.0003
PyCNFOWA	0.00051	0.0063	0.0004	0.0002	0.0001
PyCNFOWG	0.002	0.0053	0.001	0.002	0.0017
Ranking of the Alternatives					
$\varrho_{\kappa_2} > \varrho_{\kappa_4} > \varrho_{\kappa_1} > \varrho_{\kappa_3} > \varrho_{\kappa_5}$					
$\varrho_{\kappa_2} > \varrho_{\kappa_4} > \varrho_{\kappa_1} > \varrho_{\kappa_3} > \varrho_{\kappa_5}$					
$\varrho_{\kappa_2} > \varrho_{\kappa_1} > \varrho_{\kappa_3} > \varrho_{\kappa_4} > \varrho_{\kappa_5}$					
$\varrho_{\kappa_2} > \varrho_{\kappa_3} > \varrho_{\kappa_1} > \varrho_{\kappa_4} > \varrho_{\kappa_5}$					

Step 6: By utilizing the definition, we can obtained each alternative's score.

$$\varrho_{\kappa_1} = 0.002993, \varrho_{\kappa_2} = 0.03367, \varrho_{\kappa_3} = 0.00959,$$

$$\varrho_{\kappa_4} = -0.00182, \varrho_{\kappa_5} = -0.00851,$$

Step 7: According to $\varrho_{\kappa_2} > \varrho_{\kappa_3} > \varrho_{\kappa_1} > \varrho_{\kappa_4} > \varrho_{\kappa_5}$, the ranking order is $n_2 > n_3 > n_1 > n_4 > n_5$. Hence, n_2 is the best alternative. The comparative ranking of all the alternatives is shown in Figure 5.

In Table 10, the score values of the alternatives is given based on the proposed operators.

Thus alternative ϱ_{κ_2} is the best choice, because of the calculations discussed above. The results are also shown in Figure 5. In the figure, the y-axis shows the score values and on the x-axis the AOs are shown. It is clearly shown that ϱ_{κ_2} has the highest score values and it is the best alternative to be selected.

Solution by the TODIM method

Before normalizing the matrix, it is necessary to compute the decision matrix. We use the data given in Table 6 to calculate the dominance matrix which is given in Table 11. Then we calculated the alternative's utility which is shown in Table 12. From Table 12 it is clear that second alternative is the best choice. This ranking result is also shown by Figure 6.

Table 11: Final dominance matrix ($\theta = 1$).

	Y_1	Y_2	Y_3	Y_4
ϱ_{κ_1}	⟨(0.702)⟩	⟨(0.616)⟩	⟨(−1.771)⟩	⟨(−2.058)⟩
ϱ_{κ_2}	⟨(0.330)⟩	⟨(−0.140)⟩	⟨(1.286)⟩	⟨(1.1195)⟩
ϱ_{κ_3}	⟨(−0.671)⟩	⟨(−1.453)⟩	⟨(−0.990)⟩	⟨(−1.348)⟩
ϱ_{κ_4}	⟨(−1.632)⟩	⟨(0.307)⟩	⟨(−0.213)⟩	⟨(−0.321)⟩
ϱ_{κ_5}	⟨(0.698)⟩	⟨(−2.448)⟩	⟨(−1.999)⟩	⟨(−3.082)⟩

Table 12: Ranking series and alternative's utility mentioned.

Ranking	Alternatives	Gross	Normalized
1	ϱ_{κ_1}	−0.5114721	0.669692
2	ϱ_{κ_2}	2.6043008	1
3	ϱ_{κ_3}	−4.4698081	0.2500604
4	ϱ_{κ_4}	−1.85843154	0.5268980
5	ϱ_{κ_5}	−6.828598	0

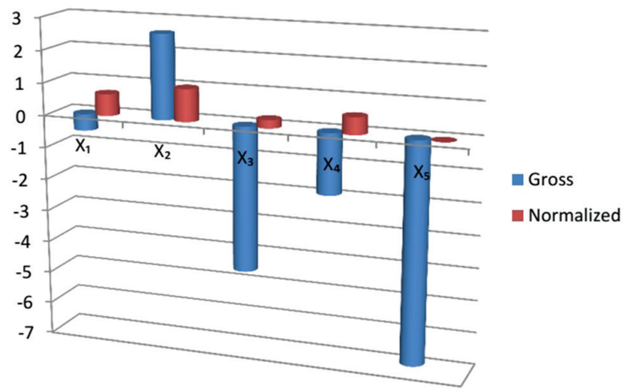

Figure 6: Ranking of the alternatives.

COMPARATIVE ANALYSIS

In this section, we compare our proposed AOs with the existing AOs in Khan et al. (2019b). Table 13 presents the results of these comparisons. Since the optimal choice remains unchanged after we solved the equivalent detailed example shown in the previous section utilizing the existing AOs in Khan et al. (2019b), the new proposed techniques prove to be much better than the old ones.

As a result of the comparative analysis shown in Table 13, we can say that alternative ϱ_{κ_2} is the finest alternative among

Table 13: Comparison analysis.

Operators	ϱ_{κ_1}	ϱ_{κ_2}	ϱ_{κ_3}	ϱ_{κ_4}	ϱ_{κ_5}
PyCNFWA Rahim et al. (2024)	0.0412	0.0456	0.007	0.0121	−0.002
PyCNFWG Rahim et al. (2024)	0.0107	0.0145	0.0102	−0.0103	−0.0235
PyCNFOWA Rahim et al. (2024)	0.0418	0.0448	0.0103	0.007	−0.0004
PyCNFOWG Rahim et al. (2024)	0.03	0.0337	0.0096	−0.0019	−0.0086
Ranking of the Alternatives					
$\varrho_{\kappa_2} > \varrho_{\kappa_1} > \varrho_{\kappa_4} > \varrho_{\kappa_3} > \varrho_{\kappa_5}$					
$\varrho_{\kappa_2} > \varrho_{\kappa_1} > \varrho_{\kappa_3} > \varrho_{\kappa_4} > \varrho_{\kappa_5}$					
$\varrho_{\kappa_2} > \varrho_{\kappa_1} > \varrho_{\kappa_3} > \varrho_{\kappa_4} > \varrho_{\kappa_5}$					
$\varrho_{\kappa_2} > \varrho_{\kappa_1} > \varrho_{\kappa_3} > \varrho_{\kappa_4} > \varrho_{\kappa_5}$					

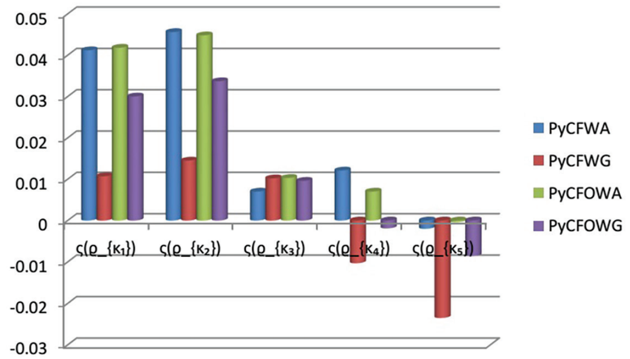

Figure 7: Ranking of the alternatives.

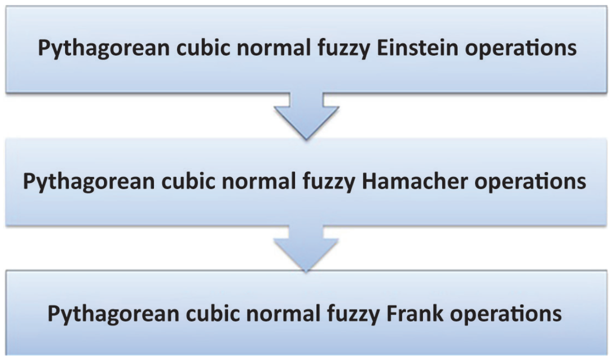

Figure 8: Future work.

all the alternatives. In Figure 7, we have presented these ranking results graphically.

CONCLUSIONS AND FUTURE WORK

The concept of AOs for PyCNFNs has been presented in this work, also many operational laws for PyCNFNs have been introduced. We have presented the Pythagorean cubic normal fuzzy AOs, including PyCNFWA, PyCNFWG, PyCNFOWA, and PyCNFOWG AOs that combine data into PyCNFNs. We introduced their fundamental axioms, i.e. idempotency, monotonicity and boundedness, and established the interrelationship among these proposed operators. To emphasize their effectiveness and address decision-making challenges more intensively, we also suggested a TODIM method considering the suggested operators. We proposed an illustrative example related to disability evaluation which showed that our suggested operators are appropriate approaches to realistically address MCDM problems. To show the effectiveness, practicality, reliability, and dependability of our proposed techniques, we presented comparisons with the existing methods. The results showed that our proposed methods are more feasible and accurate.

In the future, we will utilize frameworks established on recent multi-attribute estimate models to address ambiguity and fuzziness in decision-making parameters. These include techniques like PyCNF Frank operations, PyCNF Einstein operations, PyCNF Hamacher operations, and Dombi AO. We will develop the generalized structures of fuzzy set theory to tackle complex problems in decision support system for environmental and economic issues in supply chain management, emergency decision-making, hydropower plants evaluation, waste disposal plant site selection, evaluation of solar energy cells, manufacturing technologies, smart mines, crude oil refineries, robotics, cybercrime, artificial intelligence, photovoltaic technologies, slope design scheme renewable energy sources, robotics and many other fields. The future work is also shown in Figure 8.

FUNDING

The authors extend their appreciation to the King Salman Center for Disability Research for funding this work through Research Group No. KSRG-2023-544.

AUTHOR CONTRIBUTIONS

All authors actively contributed to every phase of the research, and they have reviewed and endorsed the final manuscript.

COMPETING INTERESTS

The authors have no conflict of interest.

ACKNOWLEDGMENTS

The authors extend their appreciation to the King Salman Center for Disability Research for funding this work through Research Group No. KSRG-2023-544.

ETHICAL APPROVAL

The authors of this article did not conduct any research involving human participants or animals.

DATA AVAILABILITY

To support this study, no data were used related to humans or animals.

REFERENCES

Abdullah S., Qiyas M., Naeem M. and Liu Y. (2022). Pythagorean cubic fuzzy Hamacher aggregation operators and their application in green supply selection problem. *AIMS Mathemat.*, 7(3), 4735-4766.

Akram M., Kahraman C. and Zahid K. (2021a). Group decision-making based on complex spherical fuzzy VIKOR approach. *Knowl. Based Syst.*, 216, 106793.

Akram M., Peng X. and Sattar A. (2021b). Multi-criteria decision-making model using complex Pythagorean fuzzy Yager aggregation operators. *Arab. J. Sci. Eng.*, 46, 1691-1717.

Akram M., Ramzan N. and Deveci M. (2023). Linguistic Pythagorean fuzzy CRITIC-EDAS method for multiple-attribute group decision analysis. *Eng. Appl. Artif. Intell.*, 119, 105777.

Alamoodi A.H., Albahri O.S., Zaidan A.A., AlSattar H.A., Ahmed M.A., Pamucar D., et al. (2022). New extension of fuzzy-weighted zero-inconsistency and fuzzy decision by opinion score method based on cubic Pythagorean fuzzy environment: a benchmarking case study of sign language recognition systems. *Int. J. Fuzzy Syst.*, 24(4), 1909-1926.

Al-Sabri E.H.A., Rahim M., Amin F., Ismail R., Khan S., Alanzi A.M., et al. (2023). Multi-criteria decision-making based on Pythagorean cubic fuzzy Einstein aggregation operators for investment management. *AIMS Mathemat.*, 8(7), 16961-16988.

Amin F., Rahim M., Ali A. and Ameer E. (2022). Generalized cubic Pythagorean fuzzy aggregation operators and their application to multi-attribute decision-making problems. *Int J. Comput. Intell. Syst.*, 15(1), 92.

Atanassov K.T. (1989). More on intuitionistic fuzzy sets. *Fuzzy Sets Syst.*, 33(1), 37-45.

Atanassov K.T. (1999). Intuitionistic fuzzy sets (pp. 1-137). Physica-Verlag HD.

Chander G. and Das S. (2021). Decision making using interval-valued Pythagorean fuzzy set-based similarity measure. In: *Intelligent Computing and Communication Systems. Algorithms for Intelligent Systems* (Singh B., Coello Coello C.A., Jindal P., Verma P. eds.), Springer, Singapore. 10.1007/978-981-16-1295-4_28.

Chen T.Y. (2020). New Chebyshev distance measures for Pythagorean fuzzy sets with applications to multiple criteria decision analysis using an extended ELECTRE approach. *Expert Syst. Appl.*, 147, 113164.

Cui C., Wei M., Che L. and Yang P. (2023). Movie recommendation algorithms based on an improved Pythagorean hesitant fuzzy distance measure and VIKOR method. *Int. J. Fuzzy Syst.*, 26, 513-526.

Demeter S.L. (2012). Disability evaluation. In: *Spine Secrets Plus*; pp. 59-64.

Deveci K. and Güler Ö. (2024). Ranking intuitionistic fuzzy sets with hypervolume-based approach: an application for multi-criteria assessment of energy alternatives. *Appl. Soft Comput.*, 150, 111038.

Hadi A., Khan W. and Khan A. (2021). A novel approach to MADM problems using Fermatean fuzzy Hamacher aggregation operators. *Int. J. Intell. Syst.*, 36(7), 3464-3499.

Hussain A., Ullah K., Alshahrani M.N., Yang M.S. and Pamucar D. (2022). Novel Aczel–Alsina operators for Pythagorean fuzzy sets with application in multi-attribute decision making. *Symmetry*, 14(5), 940.

Hussain Z., Abbas S., Hussain R. and Sharif R. (2023). Belief and plausibility measures on Pythagorean fuzzy sets and its applications with BPI-VIKOR. *J. Intell. Fuzzy Syst.*, 44(1), 729-743.

Ivanov S. and Webster C. (2024). Automated decision-making: Hoteliers' perceptions. *Technol. Soc.*, 76, 102430.

Khan F., Abdullah S., Mahmood T., Shakeel M. and Rahim M. (2019a). Pythagorean cubic fuzzy aggregation information based on confidence levels and its application to multi-criteria decision making process. *J. Intell. Fuzzy Syst.*, 36(6), 5669-5683.

Khan F., Khan M.S.A., Shahzad M. and Abdullah S. (2019b). Pythagorean cubic fuzzy aggregation operators and their application to multi-criteria decision making problems. *J. Intell. Fuzzy Syst.*, 36(1), 595-607.

Khan M.S.A., Khan F., Lemley J., Abdullah S. and Hussain F. (2020). Extended topsis method based on Pythagorean cubic fuzzy multi-criteria decision making with incomplete weight information. *J. Intell. Fuzzy Syst.*, 38(2), 2285-2296.

Kumar S. and Gupta P. (2023). Prioritizing the key actors of an organization for business excellence using the efficient interpretive ranking process. *Strojniški vestnik-J. Mech. Eng.*, 69(5-6), 248-260.

Kutlu G.F. and Kahraman C. (2019). Extension of WASPAS with spherical fuzzy sets. *Informatica*, 30(2), 269-292.

Liu P. and Li H. (2017). Multiple attribute decision-making method based on some normal neutrosophic Bonferroni mean operators. *Neural Comput. Appl.*, 28, 179-194.

Lourenzutti R., Krohling R.A. and Reformat M.Z. (2017). Choquet based TOPSIS and TODIM for dynamic and heterogeneous decision making with criteria interaction. *Inform. Sci.*, 408, 41-69.

Mollaoglu M., Gurturk M., Celik E. and Gul M. (2023). Interval type-2 trapezoidal fuzzy AHP: evaluation of sustainable port service quality factors. In: *Analytic Hierarchy Process with Fuzzy Sets Extensions: Applications and Discussions* (Cengiz Kahraman and Selcuk Cebi, eds.), pp. 27-45, Springer International Publishing, Cham.

Muneeza, Abdullah S., Qiyas M. and Khan M.A. (2022). Multi-criteria decision making based on intuitionistic cubic fuzzy numbers. *Granul. Comput.*, 7, 1-11.

Muneeza, Ihsan A. and Abdullah S. (2023). Multicriteria group decision making for COVID-19 testing facility based on picture cubic fuzzy aggregation information. *Granul. Comput.*, 8(4), 771-792.

Muneeza, Alzanin S.M. and Gumaei A.H. (2024). A multicriteria decision-making approach to create accessible environments to empower mobility-impaired individuals. *J. Disabil. Res.*, 3(6), 20240072.

Palanikumar M., Arulmozhi K., Iampan A. and Manavalan L.J. (2023). Novel possibility Pythagorean cubic fuzzy soft sets and their applications. *Int. J. Innovat. Comput.*, 19(2), 325-337.

Palanikumar M., Jana C., Mohamadghasemi A., Pal M. and Pamucar D. (2024). Selection of robot technology using q-rung normal fuzzy interaction based decision-making model. *Eng. Appl. Artif. Intell.*, 133, 108464.

Paul T.K., Jana C. and Pal M. (2023a). Enhancing multi-attribute decision making with Pythagorean fuzzy Hamacher aggregation operators. *J. Ind. Intell.*, 8, 372-383.

Paul T.K., Jana C. and Pal M. (2023b). Multi-criteria group decision-making method in disposal of municipal solid waste based on cubic Pythagorean fuzzy EDAS approach with incomplete weight information. *Appl. Soft Comput.*, 144, 110515.

Peng X., Huang H. and Luo Z. (2022). When CCN meets MCGDM: optimal cache replacement policy achieved by PRSRV with Pythagorean fuzzy set pair analysis. *Artif. Intell. Rev.*, 55(7), 5621-5671.

Premalatha R. and Dhanalakshmi P. (2022). Enhancement and segmentation of medical images through Pythagorean fuzzy sets: an innovative approach. *Neural Comput. Appl.*, 34(14), 11553-11569.

Rahim M. (2023). Multi-criteria group decision-making based on frank aggregation operators under Pythagorean cubic fuzzy sets. *Granul. Comput.*, 8, 1429-1449.

Rahim M., Amin F., Shah K., Abdeljawad T. and Ahmad S. (2024). Some distance measures for Pythagorean cubic fuzzy sets: application selection in optimal treatment for depression and anxiety. *MethodsX*, 12, 102678.

Riaz M., Naeem K. and Afzal D. (2020). A similarity measure under Pythagorean fuzzy soft environment with applications. *Comput. Appl. Mathemat.*, 39(4), 269.

Rickard J.T., Aisbett J. and Rickard J.T. (2024). On a class of general type-n normal fuzzy sets synthesized from subject matter expert inputs. *IEEE Trans. Fuzzy Syst.*, 32, 3718-3188.

Saade J.J. (1996). Mapping convex and normal fuzzy sets. *Fuzzy Syst.*, 81(2), 251-256.

Seker S. and Kahraman C. (2022). A Pythagorean cubic fuzzy methodology based on TOPSIS and TODIM methods and its application to software selection problem. *Soft Comput.*, 26(5), 2437-2450.

Sherwani R.A.K., Aslam M., Raza M.A., Farooq M., Abid M. and Tahir M. (2021). Neutrosophic normal probability distribution—a spine of parametric neutrosophic statistical tests: properties and applications. In: *Neutrosophic Operational Research* (Florentin Smarandache & Mohamed Abdel-Basset, eds.) pp. 153-169, Springer, Berlin.

Singh S. and Ganie A.H. (2020). On some correlation coefficients in Pythagorean fuzzy environment with applications. *Int. J. Intell. Syst.*, 35(4), 682-717.

Tian Y., Song S., Bao S., Zhou D. and Wei C. (2024). Canonical triangular interval type-2 fuzzy linguistic distribution assessment EDAS approach with its application to production supplier evaluation and ranking. *Appl. Soft Comput.*, 154, 111309.

Torra, V. (2010). Hesitant fuzzy sets. *Int. J. Intell. Syst.*, 25(6), 529-539.

Verma R. and Mittal A. (2023). Multiple attribute group decision-making based on novel probabilistic ordered weighted cosine similarity operators with Pythagorean fuzzy information. *Granul. Comput.*, 8(1), 111-129.

Wang L., Zhang H.Y., Wang J.Q. and Li L. (2018). Picture fuzzy normalized projection-based VIKOR method for the risk evaluation of construction project. *Appl. Soft Comput.*, 64, 216-226.

Wang P., Wang J., Wei G. and Wei C. (2019a). Similarity measures of q-rung orthopair fuzzy sets based on cosine function and their applications. *Mathematics*, 7(4), 340.

Wang R., Wang J., Gao H. and Wei G. (2019b). Methods for MADM with picture fuzzy muirhead mean operators and their application for evaluating the financial investment risk. *Symmetry*, 11(1), 6.

Wang Y., Han Z., Xing Y., Xu S. and Wang J. (2024). A survey on datasets for the decision making of autonomous vehicles. *IEEE Intell. Trans. Syst. Mag.*, 16, 23-40.

Wasim M., Yousaf A., Alolaiyan H., Akbar M.A., Alburaikan A. and El-Wahed Khalifa H.A. (2024). Optimizing decision-making with aggregation operators for generalized intuitionistic fuzzy sets and their applications in the tech industry. *Sci. Rep.*, 14(1), 16538.

Wei G., Wei C., Wang J., Gao H. and Wei Y. (2019). Some q-rung orthopair fuzzy maclaurin symmetric mean operators and their applications to potential evaluation of emerging technology commercialization. *Int. J. Intell. Syst.*, 34(1), 50-81.

Xian S., Ma D. and Feng X. (2024). Z hesitant fuzzy linguistic term set and their applications to multi-criteria decision making problems. *Exp. Syst. Appl.*, 238, 121786.

Yager R.R. (2013). Pythagorean fuzzy subsets. In: *Proceedings of the 2013 Joint IFSA World Congress and NAFIPS Annual Meeting (IFSA/NAFIPS)*; pp. 57-61, IEEE, New York, NY.

Yager R.R. (2016). Generalized orthopair fuzzy sets. *IEEE Trans. Fuzzy Syst.*, 25(5), 1222-1230.

Yager R.R. and Abbasov A.M. (2013). Pythagorean membership grades, complex numbers, and decision making. *Int. J. Intell. Syst.*, 28(5), 436-452.

Yang M.S. and Ko C.H. (1996). On a class of fuzzy c-numbers clustering procedures for fuzzy data. *Fuzzy Sets Syst.*, 84(1), 49-60.

Zadeh L.A. (1965). Fuzzy sets. *Information and Control*, 8(3), 338-353.

Zadeh L.A., Klir G.J. and Yuan B. (1996). *Fuzzy sets, Fuzzy Logic, and Fuzzy Systems: Selected Papers*, Vol. 6, World scientific, Singapore.

Zadeh L.A. (1975). The concept of a linguistic variable and its application to approximate reasoning—I. *Inf. Sci.*, 8(3), 199-249.

Zahid K., Akram M. and Kahraman C. (2022). A new ELECTRE-based method for group decision-making with complex spherical fuzzy information. *Knowl. Based Syst.*, 243, 108525.

Zhang H.Y., Ji P., Wang J.Q. and Chen X.H. (2016). A neutrosophic normal cloud and its application in decision-making. *Cognit. Comput.*, 8, 649-669.

Zhang S., Liu S., Fang Z., Zhang Q. and Zhang J. (2023). Generalized grey information entropy weight TOPSIS model for financial performance evaluation considering differentiation. *Kybernetes*, 52(11), 5412-5426.