EDAS method for decision support modeling under the Pythagorean probabilistic hesitant fuzzy aggregation information

Bushra Batool1 · Shougi Suliman Abosuliman2 · Saleem Abdullah3 · Shahzaib Ashraf4

Received: 20 July 2020 / Accepted: 25 March 2021 / Published online: 12 April 2021
© The Author(s), under exclusive licence to Springer-Verlag GmbH Germany, part of Springer Nature 2021

Abstract
The significance of emergency decision-making (EmDM) has been experienced recently due to the continuous occurrence of various emergency situations that have caused significant social and monetary misfortunes. EmDM assumes a manageable role when it is important to moderate property and live misfortunes and to reduce the negative effects on the social and natural turn of events. Genuine world EmDM issues are usually described as complex, time-consuming, lack of data, and the effect of mental practices that make it a challenging task for decision-makers. This article shows the need to manage the various types of vulnerabilities and to monitor practices to resolve these concerns. In clinical analysis, how to select an ideal drug from certain drugs with efficacy values for coronavirus disease has become a common problem these days. To address this issue, we are establishing a multi-attribute decision-making approach (MADMap) based on the EDAS method under Pythagorean probabilistic hesitant fuzzy information. In addition, an algorithm is developed to address the uncertainty in the selection of drugs in EmDM issues with regards to clinical analysis. The actual contextual analysis of the selection of the appropriate drug to treat coronavirus ailment is utilized to show the practicality of our proposed technique. Finally, with the help of a comparative analysis of the TOPSIS technique, we demonstrate the efficiency and applicability of the established methodology.

Keywords Pythagorean probabilistic hesitant fuzzy set · Decision making

1 Introduction
In the twenty-first century, with the quick monetary globalization growth and the acceleration of industrialization, ecological issues, diagnostic decision-making is definitely not a basic undertaking in drugs. The proposal and approval of diagnosis must consider the patient’s clinical boundaries, the clinical framework, and the specialist’s clinical information. Specialists utilize more than one million snippets of data in the care of their patients (Pauker et al. 1976) and just about 33% of their time is devoted recording and ordering data (Lunin and Hersh 1995). Obviously, specialists might not be able to review each thing of related data and to relate every one of these things to the consideration method (Peleg and Tu 2006). Clinical decision support systems have been established to recuperate quiet insurance and care methodology (Balas et al. 2000; Sittig et al. 2008; Ashraf and Abdullah 2020; Ashraf et al. 2020) established the emergency decision making using fuzzy decision making technique. Various examinations and investigations have uncovered these tools to be effective (Garg et al. 2005; Kawamoto et al.)
averaging operator, generalized form of quasi-hesitant fuzzy hybrid weighted geometric. Khan et al. (2017) recognized the idea of Pythagorean HFS (PyHFS). They presented assessment method and recognized operators to aggregate the data. Khan et al. (2018; 2019a, b, c) recognized Pythagorean hesitant fuzzy weighted average and hybrid aggregation operators and their application to MAGDMAp. Xu and Zhou (2017) recognized a new idea of probabilistic hesitant fuzzy sets (PHFSs). In Ayub et al. (2021), the author developed a new decision method for decision support models.

Keshavarz et al. (2015) initially proposed EDAS to resolve numerous multi-attribute decision-making approaches. The EDAS method is very fruitful specially when the incompatible criteria happen in multi-attribute decision-making approach. Analogous to VIKOR method (Mirghafoori et al. 2018) and TOPSIS method (Liang et al. 2018), some traditional distances are also derived for EDAS method. Nevertheless, EDAS method should be considered as PDAS and NDAS on the base of average solution (AS). The finest alternative should have the major value of PDAS and the minimum value of NDAS (Keshavarz et al. 2016). Kahraman et al. (2017) constructed EDAS method under IFSs. Keshavarz et al. (2017) applied the EDAS method to stochastic multi-attribute decision-making approach. Keshavarz et al. (2018) established EDAS method in dynamic multi-attribute decision-making approach. Stevic et al. (2018) considered one of the novel method based on the multicriteria analysis of fuzzy EDAS method to select the most appropriate manufacturer of PVC carpentry for the apartment refurbishing.

However, there are many findings in which the fuzzy EDAS approach is used to address decision-making problems, this type of decision-making data used by these methodologies is too old, limited and therefore can not effectively manage current decision-making environments. In addition, no matter what aggregation information is used in the fuzzy EDAS method, it may cause distortion of decision information. Therefore, the innovations of this paper are mainly the following aspects: firstly, utilized novel concept of the Pythagorean probabilistic hesitant fuzzy set (PyPHFS) established by Batool et al. (2020) to presented the new decision making technique to tackle the uncertain information in real life decision making. The motivation of the new concept is that in Probabilistic hesitant fuzzy set (PHFS) only positive membership degree is considered with probabilistic information, but PyPHFS is characterized by both positive hesitant membership and negative hesitant membership degrees, with the constraint that the square sum of positive and negative hesitant membership degrees is less than or equal to one. The DMs are limited to a specific domain in PHFS and ignore the negative degree of membership with its possible chance of occurrence. Compared to others, every negative hesitant membership degree also has some preference. For example, if one DM gives values 0.3, 0.4, 0.6 for a
positive membership degree with their corresponding preference values 0.1 and 0.9, the other may reject the DMs may express their opinion in DM-problems in the form of several possible values. Under the proposed concept, the possibility of rejection with hesitation is considered. The information of chances will decrease in spite of HFSs and PHFSs. More details on the level of difference of opinion of the DMs are provided by the value of the probability of occurrence with positive and negative membership degrees. The key purpose of this manuscript is to establish PyPHF-EDAS model and to select an ideal drug to treat Coronavirus’s ailment.

The motivations of this paper can be established as: (1) It considers various specialists’ conclusions as the group hesitancy and breakers them into PyPHFSs. (2) In clinical analysis, how to choose an ideal drug from among certain drugs with alike efficacy values to treat ailments has become normal issues among specialists and patients. Generally, in clinical practice, it is problematic for specialists to precisely measure the exact efficacy value of a drug. That is, the efficacy values of drugs are usually imprecise. Consequently, this choice issue can be recognized as a MADM issue. This manuscript mainly deliberates an approach to select an appropriate drug from among certain drugs to treat Coronavirus’s ailment. As per the efficacy value of every drug about every indication, we can utilize a MADM algorithm to attain the positioning of all drugs and select an ideal drug. (3) This paper will consolidate the knowledge of conventional EDAS algorithm and the PyPHF in managing with uncertain data to establish another decision-making algorithm and select an appropriate drug to treat Coronavirus’s ailment.

The arrangement of the manuscript is as per the following. Section 2 gives survey of FSs, IFSs, PyFSs, HFSs and PyHFSs and aggregation operators of PyHFSs. In Sect. 3 we exhibit the PyPHF-EDAS method to handle vagueness in DMAP. Section 5 explains application of the established MCDM algorithm. In Sect. 6 Comparison of established and TOPSIS method is given. In Sect. 7 conclusion and discussion of the manuscript is given.

## 2 Preliminaries

In this section, we sorts out the essential knowledge about fuzzy sets, hesitant fuzzy sets, probabilistic hesitant fuzzy sets, intuitionistic fuzzy sets and Pythagorean fuzzy sets.

**Definition 1** (Zadeh 1965) For a fixed set \( F \) a FS \( S \) in \( F \) is described as

\[
S = \{ (\ell_g, r_\mathcal{S}(\ell_g)) | \ell_g \in F \},
\]

for each \( \ell_g \in F \), the positive membership grade \( r_\mathcal{S} : F \rightarrow \Phi \) specifies the degree to which the element \( \ell_g \in S \), where \( \Phi = [0, 1] \).

**Definition 2** (Atanassov 1986) For a fixed set \( F \). An IFS \( S \) in \( F \) is described as

\[
S = \{ (\ell_g, \tau_\mathcal{S}(\ell_g), \delta_\mathcal{S}(\ell_g)) | \ell_g \in F \},
\]

for each \( \ell_g \in F \), the positive membership grade \( \tau_\mathcal{S} : F \rightarrow \Phi \) and the negative membership grade \( \delta_\mathcal{S} : F \rightarrow \Phi \) specifies the positive and negative degrees of membership of \( \ell_g \) to the IFS \( S \), respectively, where \( \Phi = [0, 1] \). Additionally, it is required that \( 0 \leq \tau_\mathcal{S}(\ell_g) + \delta_\mathcal{S}(\ell_g) \leq 1 \).

**Definition 3** (Yager 2013) For a fixed set \( F \). A PyFS \( S \) in \( F \) is described as

\[
S = \{ (\ell_g, \tau_\mathcal{S}(\ell_g), \delta_\mathcal{S}(\ell_g)) | \ell_g \in F \},
\]

for each \( \ell_g \in F \), the positive membership grade \( \tau_\mathcal{S} : F \rightarrow \Phi \) and the negative membership grade \( \delta_\mathcal{S} : F \rightarrow \Phi \) specifies the positive and negative degrees of membership of \( \ell_g \) to the PyFS \( S \), respectively, where \( \Phi = [0, 1] \). Additionally, it is required that \( 0 \leq \tau_\mathcal{S}(\ell_g) + \delta_\mathcal{S}(\ell_g) \leq 1 \). Conventionally, \( \chi_E = \sqrt{1 - \tau_\mathcal{S}^2(\ell_g) - \delta_\mathcal{S}^2(\ell_g)} \) is said to be degree of hesitancy of \( \ell_g \) to \( S \).

In what follows, we represent by \( Py\mathcal{S}(F) \) the group of all Pythagorean fuzzy sets in \( F \). For ease, we will represent the Pythagorean fuzzy number (PyFN) by the pair \( S = (\tau_\mathcal{S}, \delta_\mathcal{S}) \).

**Definition 4** (Yager 2013) Let \( S_1, S_2 \in Py\mathcal{S}(F) \). Then (1) \( S_1 \subseteq S_2 \) if and only if \( \tau_\mathcal{S}_1(\ell_g) \leq \tau_\mathcal{S}_2(\ell_g) \) and \( \delta_\mathcal{S}_1(\ell_g) \geq \delta_\mathcal{S}_2(\ell_g) \) for each \( \ell_g \in F \). Clearly \( S_1 = S_2 \) if \( S_1 \subseteq S_2 \) and \( S_2 \subseteq S_1 \), (2) \( S_1 \cap S_2 = \{ \min \{ \tau_\mathcal{S}_1(\ell_g), \tau_\mathcal{S}_2(\ell_g) \}, \max \{ \delta_\mathcal{S}_1(\ell_g), \delta_\mathcal{S}_2(\ell_g) \} | \ell_g \in F \} \), (3) \( S_1 \cup S_2 = \{ \max \{ \tau_\mathcal{S}_1(\ell_g), \tau_\mathcal{S}_2(\ell_g) \}, \min \{ \delta_\mathcal{S}_1(\ell_g), \delta_\mathcal{S}_2(\ell_g) \} | \ell_g \in F \} \), (4) \( S_1^c = \{ \delta_\mathcal{S}_1(\ell_g), \tau_\mathcal{S}_1(\ell_g) | \ell_g \in F \} \).

**Definition 5** (Torra 2010) For a fixed set \( F \). A HFS \( S \) in \( F \) is described as

\[
S = \{ (\ell_g, h_\mathcal{S}(\ell_g)) | \ell_g \in F \},
\]

where \( h_\mathcal{S}(\ell_g) \) is in the form of set, that’s contained some possible values in unit interval, i.e., \([0, 1]\) which represent the membership degree of \( \ell_g \) in \( F \) in \( S \).

**Definition 6** (Torra 2010) Let \( S_1, S_2 \in HFS(F) \). Then

(1) \( S_1^c = \bigcup_{\ell_g \in S_1(\ell_g)} \{ 1 - \ell_g \} \).
Definition 8 (Khan et al. 2017) For a fixed set $\mathcal{F}$. A PyHFS $\mathfrak{A}$ in $\mathcal{F}$ is presented as

$$\mathfrak{A} = \left\{ \left( \ell_\varnothing, \tau_{\mathcal{H}_\varnothing}(\ell_\varnothing), \vartriangle_{\mathcal{H}_\varnothing}(\ell_\varnothing) \right) \mid \ell_\varnothing \in \mathcal{F} \right\},$$

for each $\ell_\varnothing \in \mathcal{F}$, the positive membership grade $\tau_{\mathcal{H}_\varnothing}$ and the negative membership grade $\vartriangle_{\mathcal{H}_\varnothing}$ are sets in some values in $[0,1]$, specifies the possible positive and negative degrees of membership of $\ell_\varnothing$ to the Pythagorean hesitant fuzzy set $\mathfrak{A}$, respectively. Furthermore, it is required that $(\max(\tau_{\mathcal{H}_\varnothing}(\ell_\varnothing)), (\min(\vartriangle_{\mathcal{H}_\varnothing}(\ell_\varnothing)))^2 \leq 1$ and $(\max(\vartriangle_{\mathcal{H}_\varnothing}(\ell_\varnothing)), (\min(\tau_{\mathcal{H}_\varnothing}(\ell_\varnothing)))^2 \leq 1$. For ease, we will represent the Pythagorean Hesitant Fuzzy Number (PyHFN) by the pair $\mathfrak{A} = (\tau_{\mathcal{H}_\varnothing}, \vartriangle_{\mathcal{H}_\varnothing}).$

Definition 9 (Xu and Zhou 2017) For a fixed set $\mathcal{F}$. A PHFS $\mathfrak{S}$ in $\mathcal{F}$ is described as

$$\mathfrak{S} = \left\{ \left( \ell_\varnothing, h_x(\ell_\varnothing), \varnothing_{\mathcal{H}_\varnothing}(\ell_\varnothing) \right) \mid \ell_\varnothing \in \mathcal{F} \right\},$$

where $h_x(\ell_\varnothing)$ is subset of $[0,1]$ and $h_x(\ell_\varnothing)$ represent the membership degree of $\ell_\varnothing \in \mathcal{F}$ in $\mathfrak{S}$. And $\varnothing_{\mathcal{H}_\varnothing}$ represents the possibilities of $h_x(\ell_\varnothing)$, with constraint that $\sum_{\varnothing} \varnothing_{\mathcal{H}_\varnothing} = 1.$

### 3 Pythagorean probabilistic hesitant fuzzy set

Definition 10 (Batool et al. 2020) For a fixed set $\mathcal{F}$. A PyPHFS $\mathfrak{S}$ in $\mathcal{F}$ is described as

$$\mathfrak{S} = \left\{ \left( \ell_\varnothing, \tau_{\mathcal{H}_\varnothing}(\ell_\varnothing)/\varnothing_{\mathcal{H}_\varnothing}(\ell_\varnothing) \right) \mid \ell_\varnothing \in \mathcal{F} \right\},$$

for all $\ell_\varnothing \in \mathcal{F}$, $\tau_{\mathcal{H}_\varnothing}(\ell_\varnothing)$ and $\varnothing_{\mathcal{H}_\varnothing}(\ell_\varnothing)$ are sets of some values in $[0,1]$. Where $\tau_{\mathcal{H}_\varnothing}(\ell_\varnothing)$ and $\varnothing_{\mathcal{H}_\varnothing}(\ell_\varnothing)$ specifies the possible positive and negative degrees of membership of $\ell_\varnothing$ to the Pythagorean probabilistic hesitant fuzzy set $\mathfrak{S}$, respectively. $\varnothing_{\mathcal{H}_\varnothing}$ and $\varnothing_{\mathcal{H}_\varnothing}$ represent the possibilities of membership grades. Also, there is $0 \leq \ell_{ij} \leq 1$ and $0 \leq \ell_{ij}, \ell_{ij} \leq 1$ with $\sum_{i=1}^{I} \varnothing_{ij} \leq 1$, $\sum_{j=1}^{J} \varnothing_{ij} \leq 1$ (L is a positive integer to describe the number of elements contained in PyPHFS), where $\ell_{ij} \in \tau_{\mathcal{H}_\varnothing}(\ell_\varnothing), \ell_{ij} \in \varnothing_{\mathcal{H}_\varnothing}(\ell_\varnothing), \ell_{ij} \in \varnothing_{\mathcal{H}_\varnothing}(\ell_\varnothing), \ell_{ij} \in \varnothing_{\mathcal{H}_\varnothing}(\ell_\varnothing), \ell_{ij} \in \varnothing_{\mathcal{H}_\varnothing}(\ell_\varnothing), \ell_{ij} \in \varnothing_{\mathcal{H}_\varnothing}(\ell_\varnothing).$ Additionally, it is required that $(\max(\tau_{\mathcal{H}_\varnothing}(\ell_\varnothing)))^2 + (\min(\varnothing_{\mathcal{H}_\varnothing}(\ell_\varnothing)))^2 \leq 1$ and $(\max(\varnothing_{\mathcal{H}_\varnothing}(\ell_\varnothing)))^2 + (\min(\tau_{\mathcal{H}_\varnothing}(\ell_\varnothing)))^2 \leq 1.$

For ease, we will represent the Pythagorean Probabilistic Hesitant Fuzzy Number (PyPHFN) by the pair $\mathfrak{S} = (\tau_{\mathcal{H}_\varnothing}/\varnothing_{\mathcal{H}_\varnothing}, \varnothing_{\mathcal{H}_\varnothing}/\varnothing_{\mathcal{H}_\varnothing}).$ The group of all Pythagorean probabilistic hesitant fuzzy sets in $\mathcal{F}$ is represented by $PyPHF(\mathfrak{S}).$

Definition 11 (Batool et al. 2020) Let $\mathfrak{S}_1 = (\tau_{\mathcal{H}_1}/\varnothing_{\mathcal{H}_1}, \varnothing_{\mathcal{H}_1}/\varnothing_{\mathcal{H}_1})$ and $\mathfrak{S}_2 = (\tau_{\mathcal{H}_2}/\varnothing_{\mathcal{H}_2}, \varnothing_{\mathcal{H}_2}/\varnothing_{\mathcal{H}_2})$ be two PHPHFS. The basic operational laws defined as

$$\mathfrak{S}_1 \cup \mathfrak{S}_2 = \left\{ \left( \ell_\varnothing, \tau_{\mathcal{H}_\varnothing}(\ell_\varnothing), \varnothing_{\mathcal{H}_\varnothing}(\ell_\varnothing) \right) \mid \ell_\varnothing \in \mathcal{F} \right\},$$

$$\mathfrak{S}_1 \cap \mathfrak{S}_2 = \left\{ \left( \ell_\varnothing, \tau_{\mathcal{H}_\varnothing}(\ell_\varnothing), \varnothing_{\mathcal{H}_\varnothing}(\ell_\varnothing) \right) \mid \ell_\varnothing \in \mathcal{F} \right\},$$

$$\mathfrak{S}_1^c = \left\{ \left( \ell_\varnothing, \tau_{\mathcal{H}_\varnothing}(\ell_\varnothing), \varnothing_{\mathcal{H}_\varnothing}(\ell_\varnothing) \right) \mid \ell_\varnothing \in \mathcal{F} \right\}.$$
Definition 13

For any PyPHFN $\mathfrak{S} = (\tau_h / \varphi_h, \mathfrak{d}_h / T_h)$, a score function be defined as

$$s(\mathfrak{S}) = \left( \frac{1}{M_3} \sum_{\ell \in \varphi_h} (\ell_i \cdot \varphi_i)^2 \right)^2 - \left( \frac{1}{N_3} \sum_{\ell \in \varphi_h} (\ell_i \cdot T_i)^2 \right)^2,$$

where $M_3$ represents the number of elements in $\tau_h$, and $N_3$ represents the number of elements in $\mathfrak{d}_h$.

Definition 14

For any PyPHFN $\mathfrak{S} = (\tau_h / \varphi_h, \mathfrak{d}_h / T_h)$, an accuracy function is defined as

$$\mathfrak{A}(\mathfrak{S}) = \frac{1}{M_3} \sum_{\ell \in \varphi_h} (\ell_i \cdot \varphi_i)^2.$$

where $M_3$ represents the number of elements in $\tau_h$, and $N_3$ represents the number of elements in $\mathfrak{d}_h$.
\[ h(\mathcal{S}) = \left( \frac{1}{M_{\mathcal{S}}} \sum_{\mathcal{F} \in \mathcal{F}_{\mathcal{S}}} (\ell_i \cdot \varphi_i) \right)^2 + \left( \frac{1}{N_{\mathcal{S}}} \sum_{\varnothing \in \mathcal{D}, \mathcal{P} \in \mathcal{P}_{\mathcal{S}}} (\alpha_i \cdot \beta_i) \right)^2, \]

where \( M_{\mathcal{S}} \) represents the number of elements in \( r_{h_{\mathcal{S}}} \) and \( N_{\mathcal{S}} \) represents the number of elements in \( d_{h_{\mathcal{S}}} \).

**Definition 15** Let \( \mathcal{S}_1 = (r_{h_{1}}, \mathcal{D}_{h_{1}}) \) and \( \mathcal{S}_2 = (r_{h_{2}}, \mathcal{D}_{h_{2}}) \) be two PyPHFNs. Then by using above definition, comparison of PyPHFNs can be described as (1) If \( s(\mathcal{S}_1) > s(\mathcal{S}_2) \), then \( \mathcal{S}_1 > \mathcal{S}_2 \). (2) If \( s(\mathcal{S}_1) = s(\mathcal{S}_2) \), and \( h(\mathcal{S}_1) > h(\mathcal{S}_2) \) then \( \mathcal{S}_1 > \mathcal{S}_2 \).

**Definition 16** (Khan et al. 2018) Let \( \mathcal{S}_j = (r_{h_{j}}, d_{h_{j}}) \) be a group of all PyPHFNs, and \( \lambda = (\lambda_1, \lambda_2, \ldots, \lambda_d)^T \) be the weights of \( \mathcal{S}_j \), \( \lambda_j \geq 0 \), with \( \sum_{j=1}^d \lambda_j = 1 \). Then PyFHFWA operator \( \text{PyFHFWA} : \text{PyHFN}^d \rightarrow \text{PyHFN} \) can be described as

\[
\text{PyFHFWA}(\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_d) = \lambda_1 \mathcal{S}_1 \oplus \lambda_2 \mathcal{S}_2 \oplus \cdots \oplus \lambda_d \mathcal{S}_d,
\]

where \( \lambda_1, \lambda_2, \ldots, \lambda_d \geq 0 \) and \( \sum_{j=1}^d \lambda_j = 1 \).

**Definition 17** (Khan et al. 2018) Let \( \mathcal{S}_j = (r_{h_{j}}, d_{h_{j}}) \) be a group of all PyPHFNs, and \( \lambda = (\lambda_1, \lambda_2, \ldots, \lambda_d)^T \) be the weights of \( \mathcal{S}_j \), \( \lambda_j \geq 0 \), with \( \sum_{j=1}^d \lambda_j = 1 \). Then PyFHFWG operator \( \text{PyFHFWG} : \text{PyHFN}^d \rightarrow \text{PyHFN} \) can be described as

\[
\text{PyFHFWG}(\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_d) = \mathcal{S}_1^\lambda \otimes \mathcal{S}_2^\lambda \otimes \cdots \otimes \mathcal{S}_d^\lambda,
\]

where \( \lambda = (\lambda_1, \lambda_2, \ldots, \lambda_d)^T \) be the weights of \( \mathcal{S}_j \).

**Definition 18** (Khan et al. 2018) Let \( \mathcal{S}_j = (r_{h_{j}}, d_{h_{j}}) \) be a group of all PyPHFNs, and \( \lambda = (\lambda_1, \lambda_2, \ldots, \lambda_d)^T \) be the weights of \( \mathcal{S}_j \), \( \lambda_j \geq 0 \), with \( \sum_{j=1}^d \lambda_j = 1 \). Then PyFHFOWA operator \( \text{PyFHFOWA} : \text{PyHFN}^d \rightarrow \text{PyHFN} \) can be described as

\[
\text{PyFHFOWA}(\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_d) = \lambda_1 \mathcal{S}_1 \oplus \lambda_2 \mathcal{S}_2 \oplus \cdots \oplus \lambda_d \mathcal{S}_d,
\]

where \( \lambda_1, \lambda_2, \ldots, \lambda_d \geq 0 \) and \( \sum_{j=1}^d \lambda_j = 1 \).

**Definition 19** (Khan et al. 2018) Let \( \mathcal{S}_j = (r_{h_{j}}, d_{h_{j}}) \) (\( j = 1, 2, \ldots, d \)) be a group of all PyPHFNs. Then PyFHFOWG operator \( \text{PyFHFOWG} : \text{PyHFN}^d \rightarrow \text{PyHFN} \) can be described as

\[
\text{PyFHFOWG}(\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_d) = \mathcal{S}_1^{\lambda_1} \otimes \mathcal{S}_2^{\lambda_2} \otimes \cdots \otimes \mathcal{S}_d^{\lambda_d}.
\]

**4 Aggregation information for PyPHFNs**

This section presents some aggregation operators for Pythagorean Probabilistic hesitant fuzzy numbers derived from operational laws.

**Definition 20** Let \( \mathcal{S}_j = (r_{h_{j}}, d_{h_{j}}) \) be any group of PyPHFNs and PyFHFWA operator \( \text{PyFHFWA} : \text{PyHFN}^d \rightarrow \text{PyHFN} \) can be described as

\[
\text{PyFHFWA}(\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_d) = \lambda_1 \mathcal{S}_1 \oplus \lambda_2 \mathcal{S}_2 \oplus \cdots \oplus \lambda_d \mathcal{S}_d,
\]

where \( \lambda = (\lambda_1, \lambda_2, \ldots, \lambda_d)^T \) be the weights of \( \mathcal{S}_j \), \( \lambda_j \geq 0 \), with \( \sum_{j=1}^d \lambda_j = 1 \).

**Theorem 1** Let \( \mathcal{S}_j = (r_{h_{j}}, d_{h_{j}}) \) be any group of PyPHFNs. Then the aggregation result using PyFHFWA, we can achieve the following

\[
\text{PyFHFWA}(\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_d) = \left[ \frac{1}{\mathcal{S}_{\mathcal{F}}} \sum_{r_{h_{j}}} \left( 1 - \left( \frac{\sum_{j=1}^d \lambda_j r_{h_{j}}}{\sum_{j=1}^d \lambda_j} \right)^2 \right)^{\mathcal{F}} \right]^{1/\mathcal{F}}
\]

where \( \lambda = (\lambda_1, \lambda_2, \ldots, \lambda_d)^T \) be the weights of \( \mathcal{S}_j \), \( \lambda_j \geq 0 \), with \( \sum_{j=1}^d \lambda_j = 1 \).

**Proof** We will demonstrate the theorem by following the steps mathematical induction on \( r \), and the proof is executed as beneath.
Step 1 When \( d = 2 \), we have \( \mathcal{S}_1 = \left( r_{h_1}/\mathcal{G}_{g_1}, d_{h_1}/T_{g_1} \right) \) and \( \mathcal{S}_2 = \left( r_{h_2}/\mathcal{G}_{g_2}, d_{h_2}/T_{g_2} \right) \). Thus, by the operation of PyPHFES, we achieve

\[
\lambda_1 \mathcal{S}_1 = \left\{ \lambda_1 \in \tau_{h_1}(u_j, T_1) \in T_{g_1} \right\} \cup \left\{ \lambda_1 \in \tau_{h_1}(u_j, T_1) \in T_{g_1} \right\} \\
\lambda_2 \mathcal{S}_2 = \left\{ \lambda_2 \in \tau_{h_2}(u_j, T_2) \in T_{g_2} \right\} \cup \left\{ \lambda_2 \in \tau_{h_2}(u_j, T_2) \in T_{g_2} \right\}
\]

Now,

**PyPHFWA(\( \mathcal{S}_1, \mathcal{S}_2 \))**

\[
\lambda_1 \mathcal{S}_1 \oplus \lambda_2 \mathcal{S}_2 = \left\{ \left( 1 - (1 - \ell^2) \lambda_1, 1 - (1 - \ell^2) \lambda_2 \right) / (\mathcal{G}_{g_1}/\mathcal{G}_{g_2}) \right\}
\]

Thus, the result holds for \( r = 2 \).

Step 2 Assume that the result holds for \( r = n \), we have
Let $\mathbf{\Omega}_j = (\tau_{h_j} / \varphi_{h_j} \cdot \mathcal{A}_{h_j} / T_{h_j}) (j = 1, 2, \ldots, d)$ be any group of PyPHFNs, and PyPHFWG : PyPHFN$^d \rightarrow$ PyPHFN. Then PyPHFWG operator can be described as

$$
\text{PyPHFWG}(\mathbf{\Omega}_1, \mathbf{\Omega}_2, \ldots, \mathbf{\Omega}_d) = \prod_{j=1}^{d} \mathbf{\Omega}_j
$$

and $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_d)^T$ are the weights of $\mathbf{\Omega}_j \in [0, 1]$ with $\sum_{j=1}^{d} \lambda_j = 1$.

**Proof** We will demonstrate the theorem by following the steps mathematical induction on $r$, and the proof is executed as beneath:
Step 1 When \( d = 2 \), we have \( \mathcal{F}_1 = \left( r_{h_1} / \varphi_{\xi_1}, d_{h_1} / T_{\xi_1} \right) \) and \( \mathcal{F}_2 = \left( r_{h_2} / \varphi_{\xi_2}, d_{h_2} / T_{\xi_2} \right) \). Thus, by the operation of PyPHFEs, we achieve

\[
\left( \mathcal{F}_1 \right)^3 = \left( \mathcal{F}_2 \right)^3
\]

Now,

\[
\text{PyPHFG}(\mathcal{F}_1, \mathcal{F}_2) = \left( \mathcal{F}_1 \right)^3 \otimes \left( \mathcal{F}_2 \right)^3
\]

Thus, the result holds for \( r = 2 \).
Step 2 Assume that the result holds for \( r = n \), we have

\[
PyPHFWG(\mathfrak{F}_1, \mathfrak{F}_2, \ldots, \mathfrak{F}_n) = \begin{cases} 
\bigcup_{\phi_j \in \mathfrak{F}_j, \lambda_j \in \mathfrak{T}_{\lambda_j}} \Pi^n_{j=1} \left( \phi_j \right)^{\lambda_j} / \Pi^n_{j=1} \mathfrak{T}_{\lambda_j}, \\
\bigcup_{\phi_j \in \mathfrak{F}_j, \lambda_j \in \mathfrak{T}_{\lambda_j}} \sqrt{1 - \Pi^n_{j=1} \left( 1 - \phi_j \right)^{2\lambda_j} / \Pi^n_{j=1} \mathfrak{T}_{\lambda_j}}
\end{cases}
\]

When \( r = n + 1 \), then we have

\[
PyPHFWG(\mathfrak{F}_1, \mathfrak{F}_2, \ldots, \mathfrak{F}_{n+1}) = \prod_{j=1}^{n+1} (\mathfrak{F}_j)^{\lambda_j} \otimes (\mathfrak{F}_{n+1})^{\lambda_{n+1}}
\]

Thus

\[
PyPHFWG(\mathfrak{F}_1, \mathfrak{F}_2, \ldots, \mathfrak{F}_d) = \begin{cases} 
\bigcup_{\phi_j \in \mathfrak{F}_j, \lambda_j \in \mathfrak{T}_{\lambda_j}} \Pi^d_{j=1} \left( \phi_j \right)^{\lambda_j} / \Pi^d_{j=1} \mathfrak{T}_{\lambda_j}, \\
\bigcup_{\phi_j \in \mathfrak{F}_j, \lambda_j \in \mathfrak{T}_{\lambda_j}} \sqrt{1 - \Pi^d_{j=1} \left( 1 - \phi_j \right)^{2\lambda_j} / \Pi^d_{j=1} \mathfrak{T}_{\lambda_j}}
\end{cases}
\]

Proved.

Definition 23 Let \( \mathcal{S}_j = \left( \tau_{s_j} / \mathfrak{F}_{s_j}, \mathfrak{T}_{s_j} / \mathfrak{T}_{\lambda_j} \right) \) \((j = 1, 2, \ldots, d)\) be any group of PyPHFNs, and \( PyPHFWG : PyPHFN^d \rightarrow PyPHF \). Then PyPHFWG operator can be described as

\[
PyPHFWG(\mathfrak{F}_1, \mathfrak{F}_2, \ldots, \mathfrak{F}_d) = (\mathfrak{F}_{x(1)})^{\lambda_1} \otimes (\mathfrak{F}_{x(2)})^{\lambda_2} \ldots \otimes (\mathfrak{F}_{x(d)})^{\lambda_d}
\]

\[
= \prod_{j=1}^{d} \left( \mathfrak{F}_{x(j)} \right)^{\lambda_j}
\]

where \( \mathfrak{F}_{x(j)} \) be the jth largest in them and \( \lambda = (\lambda_1, \lambda_2, \ldots, \lambda_d)^T \) are the weights of \( \mathfrak{F}_j \in [0, 1] \) with \( \sum_{j=1}^{d} \lambda_j = 1 \).

Theorem 4 Let \( \mathcal{S}_j = \left( \tau_{s_j} / \mathfrak{F}_{s_j}, \mathfrak{T}_{s_j} / \mathfrak{T}_{\lambda_j} \right) \) \((j = 1, 2, \ldots, d)\) be any group of PyPHFNs. Then the aggregation result using PyPHFWG, we can achieve the following

\[
PyPHFWG(\mathfrak{F}_1, \mathfrak{F}_2, \ldots, \mathfrak{F}_d)
\]

\[
= \begin{cases} 
\bigcup_{\phi_j \in \mathfrak{F}_j, \lambda_j \in \mathfrak{T}_{\lambda_j}} \Pi^d_{j=1} \left( \phi_j \right)^{\lambda_j} / \Pi^d_{j=1} \mathfrak{T}_{\lambda_j}, \\
\bigcup_{\phi_j \in \mathfrak{F}_j, \lambda_j \in \mathfrak{T}_{\lambda_j}} \sqrt{1 - \Pi^d_{j=1} \left( 1 - \phi_j \right)^{2\lambda_j} / \Pi^d_{j=1} \mathfrak{T}_{\lambda_j}}
\end{cases}
\]

where \( \mathfrak{F}_{x(j)} \) be the jth largest in them and \( \lambda = (\lambda_1, \lambda_2, \ldots, \lambda_d)^T \) are the weights of \( \mathfrak{F}_j \in [0, 1] \) with \( \sum_{j=1}^{d} \lambda_j = 1 \).

Proof Prove is similarly as Theorem 3.

\[\square\]

5 EDAS approach for MAGDM based on PyPHFNs

Now we establish a framework for solving MAGDM issues under PyPHF information.

Let \( \{ \mathfrak{F}_{s_1}, \mathfrak{F}_{s_2}, \ldots, \mathfrak{F}_{s_p} \} \) be a set of p attributes and let \( \{ \mathfrak{F}_1, \mathfrak{F}_2, \ldots, \mathfrak{F}_d \} \) be a set of attributes with weight vector \( \lambda = (\lambda_1, \lambda_2, \ldots, \lambda_d) \) where \( \lambda_i \in [0, 1] \) and \( \sum_{i=1}^{d} \lambda_i = 1 \). To assess the performance of kth alternative \( \mathfrak{F}_{s_k} \) under the kth attribute \( \mathfrak{F}_k \), let \( \{ D_1, D_2, \ldots, D_j \} \) be a set of experts and \( \sigma = (\sigma_1, \sigma_2, \ldots, \sigma_j) \) be the weighted vector of experts with \( \sigma_j \in [0, 1] \) and \( \sum_{j=1}^{d} \sigma_j = 1 \). On the basis of the conventional EDAS algorithm, the EDAS algorithm for MAGDM is established under PyPHF environment. Key steps are described as:

Step 1 Construct the PyHF decision matrix based on the experts evaluations.

\[
C = \left[ \mathfrak{F}_{x(s)} \right]_{p \times d} = \left[ \left( \tau_{s_j} / \mathfrak{F}_{s_k}, \mathfrak{T}_{s_k} / \mathfrak{T}_{\lambda_j} \right) \right]_{p \times d}
\]

\((s = 1, 2, \ldots, \bar{s} \in 1, 2, \ldots, p; \bar{j} = 1, 2, \ldots, d)\).

Step 2 Utilize Pythagorean probabilistic hesitant fuzzy weighted averaging aggregation operators to achieve the overall PyPHF information.

\[
PyPHFWA(\mathfrak{F}_1, \mathfrak{F}_2, \ldots, \mathfrak{F}_d)
\]

\[
= \begin{cases} 
\bigcup_{\lambda_j \in \mathfrak{T}_{\lambda_j}} \Pi^d_{j=1} \left( \phi_j \right)^{\lambda_j} / \Pi^d_{j=1} \mathfrak{T}_{\lambda_j}, \\
\bigcup_{\phi_j \in \mathfrak{F}_j, \lambda_j \in \mathfrak{T}_{\lambda_j}} \sqrt{1 - \Pi^d_{j=1} \left( 1 - \phi_j \right)^{2\lambda_j} / \Pi^d_{j=1} \mathfrak{T}_{\lambda_j}}
\end{cases}
\]
Table 1 Normalized collective data of experts

| \( \mathcal{I}_i \) | \( \mathcal{J}_2 \) | \( \mathcal{J}_3 \) |
|-----------------|-----------------|-----------------|
| \( \mathcal{D}_1 \) | (0.48/0.4, 0.4/0.6) | (0.65/1) | (0.7/0.1, 0.7/0.9) |
| \( \mathcal{D}_2 \) | (0.5/0.8, 0.5/2.0) | (0.3/1) | (0.8/1) |
| \( \mathcal{D}_3 \) | (0.2/0.7, 0.4/0.3) | (0.2/1) | (0.3/0.2, 0.6/0.8) |
| \( \mathcal{D}_4 \) | (0.52/0.8, 0.7/0.2) | (0.6/1) | (0.2/1) |

Table 2 \( S(AS_i) \)

-0.19716 0.24639 0.02494

Table 3 The score matrix of Table 1

| \( \mathcal{J}_1 \) | \( \mathcal{J}_2 \) | \( \mathcal{J}_3 \) |
|-----------------|-----------------|-----------------|
| \( \mathcal{D}_1 \) | -0.1934 | 0.3901 | 0.1201 |
| \( \mathcal{D}_2 \) | -0.1865 | 0.0781 | 0.6058 |
| \( \mathcal{D}_3 \) | -0.1431 | -0.0176 | -0.0127 |
| \( \mathcal{D}_4 \) | -0.0827 | 0.3375 | 0.0182 |

Step 3 Determine the AS of all the alternatives under each attribute.

\[
AS = (AS_1, AS_2, \ldots, AS_d),
\]

where

\[
AS_i = \left( \tau_i / \mathcal{P}_{k1} \mathcal{J}_1, \ldots, \tau_i / \mathcal{P}_{kd} \mathcal{J}_d \right) = \frac{1}{p} \mathcal{J}_1 \mathcal{P}_{kl} + \frac{1}{p} \mathcal{J}_2 \mathcal{P}_{kl} + \cdots + \frac{1}{p} \mathcal{J}_d \mathcal{P}_{kl}
\]

\[
\tau_i = \left\{ \begin{array}{l}
\bigcup_{\mathcal{P}_{kl} \in \mathcal{P}_{kl}, \mathcal{P}_{kl} \in \mathcal{P}_{kl}} \sqrt{1 - \Pi_{j=1}^p \left( 1 - \left( \lambda_{ij} \right) ^2 \right) / \Pi_{j=1}^p \mathcal{P}_{kl}}, \\
\bigcup_{\mathcal{P}_{kl} \in \mathcal{P}_{kl}} \Pi_{j=1}^p \mathcal{P}_{kl}\end{array} \right.
\]

\[
(t = 1, 2, \ldots, d)
\]

Step 4 Determine the PDAS and NDAS matrices.

\[
PDAS = (PDAS_{kl})_{p \times d}
\]

\[
NDAS = (NDAS_{kl})_{p \times d}
\]

where

\[
PDAS_{kl} = \max \left( 0, \left( S(\mathcal{J}_1) - S(AS_i) \right) \right) / S(AS_i)
\]

\[
NDAS_{kl} = \max \left( 0, \left( S(AS_i) - S(\mathcal{J}_1) \right) \right) / S(AS_i)
\]

Step 5 Calculate the positive weighted distance \( SP_k (k = 1, 2, \ldots, p) \) and the negative weighted distance \( SN_k (k = 1, 2, \ldots, p) \):

\[
SP_k = \sum_{i=1}^d \lambda_i PDAS_{kl}
\]

\[
SN_k = \sum_{i=1}^d \lambda_i NDAS_{kl},
\]

where \( \lambda_i \in [0, 1] \), \( \sum_{i=1}^d \lambda_i = 1 \).

Step 6 Normalize the \( SP_k (k = 1, 2, \ldots, p) \) and \( SN_k (k = 1, 2, \ldots, p) \) in the way given as:

\[
NSP_k = \frac{SP_k}{\max \left( SP_1, SP_2, \ldots, SP_p \right)}
\]

\[
NSN_k = \frac{SN_k}{\max \left( SN_1, SN_2, \ldots, SN_p \right)}
\]
Scores Ranking
\(\Omega_1, \Omega_2, \Omega_3, \Omega_4\)

| Methods | \(\Omega_1\) | \(\Omega_2\) | \(\Omega_3\) | \(\Omega_4\) |
|---------|-------------|-------------|-------------|-------------|
| TOPSIS  | 0.03        | 0.81        | 0.86        | 0.99        |
| EDAS    | 0.60        | 0.86        | 0.84        | 0.95        |

| Step 7 | Calculate the integrative appraisal score \((I_{AS})\) \((k = 1, 2, \ldots, \rho)\) in the way given as:
|         | \(I_{AS} = \frac{1}{2}(NS_{P} + 1 - NS_{N})\) |
|         | Where \(I_{AS} \in [0, 1]\). |
| Step 8 | Calculate the ordering according to the result of \((I_{AS})\) \((k = 1, 2, \ldots, \rho)\). The larger \(I_{AS}\), is the best alternative. |

### 6 Numerical example

To validate our established algorithm we consider the case of selecting an optimal drug to treat Coronavirus’s ailment.

#### 6.1 Case study

In the present climate, Coronavirus’s ailment is one of the fatal ailments among the elderly, which is rapidly spreading all over the world and characterized by fever, cold, cough, breathing problems, sore throat, headache and so on. There is as of now no immediate and effective drug to treat it. In clinical practice, drugs to treat Coronavirus’s ailment are fundamentally symbolic medicines to control the flue, cough, weak immune system, etc. In the clinical practice, specialists generally combine many drugs to treat Coronavirus’s ailment.

Let \(\{\Omega_1, \Omega_2, \Omega_3, \Omega_4\}\) be a set of drugs and let \(\{S_1, S_2, S_3\}\) be a set of symptoms. Generally, from a medical opinion, the border of an attribute value is normally imprecise. We can accomplish that inadequacy, fuzziness and vagueness are inherent structures of clinical practice. Consequently, the efficacy value of drugs w.r.t. a symptom can be viewed as a fuzzy set. The weight vector for symptoms is \(\varphi = (0.314, 0.355, 0.331)^T\). So as to keep away from the danger of abuse and misdiagnosis in the treatment of Coronavirus’s ailment, specialists ought to assess drug’s efficacy values w.r.t. all symptoms in combination with their involvement in clinic and select an appropriate drug to treat it.

\(S_1 =\) Fever,

\(S_2 =\) Breathing problem,

\(S_3 =\) Cough

The estimation values of the alternatives regarding each criterion provided by the specialists are developed by PyPH-FNs as revealed in the PyPH decision matrix given in Table 1. To solve the MCDM issue by developed operators, the following calculations are achieved:

| Step 1 | Calculate the \(AS\) of all the alternatives under each attribute by using PyPHFWA operator. |
|        | \(AS_i = \left(\bigcup_{\lambda_{k1} \in \tau_{\alpha_{k1}}} \sqrt{1 - \Pi_{k=1}^{4} \left(1 - \left(\lambda_{\alpha_{k1}}\right)^{2}\right)^{1/4}} / \Pi_{k=1}^{4} \varphi_{\alpha_{k1}}\right)\) |
|        | Similarly for other attribute. |
| Step 2 | Calculated score values of \(AS_i(t = 1, 2, 3, \ldots)\) is shown in Table 2. |

After that, we calculated the score of Table 1 in Table 3.

| Step 3 | By exploited the score values, calculate the PDAS and NDAS in Table 4. |
|        | Step 4 | Calculated the positive weighted distance \(SP_i(k = 1, 2, 3, 4)\) and the negative weighted distance \(SN_i(k = 1, 2, \ldots, \rho)\), where \(\varphi = (0.314, 0.355, 0.331)^T\) in Table 5. |

| Step 5 | Normalized the values \(SP_i(k = 1, 2, 3, 4)\) and \(SN_i(k = 1, 2, \ldots, \rho)\) is shown in Table 6. |

| Step 6 | Calculated the values of \(I_{AS}(k = 1, 2, 3, 4)\) in Table 7. |

\(I_{AS} > I_{AS} > I_{AS} > I_{AS}\)

Therefore,

\(\Omega_4 > \Omega_2 > \Omega_3 > \Omega_1\).

The best drug is \(\Omega_4\).
We can conclude from this above computational process that $\mathcal{O}^4$ is the best drug for the COVID-19 patients, among others, and therefore it is highly recommended.

7 Comparison analysis

To validate the effectiveness of PyPHF-EDAS algorithm, a comparison between PyPHF-EDAS algorithm (Table 8) and PyPHF-TOPSIS algorithm is taken into account. The ranking of the drugs with weight vector $\varphi = (0.01, 0.35, 0.64)$ is listed in Table-8.

Comparison between TOPSIS algorithm and extended form of EDAS algorithm shows that the best and worst alternative are same. EDAS algorithm is steady and progressive when various weights for the criteria are allocated. The evaluations of alternatives in EDAS algorithm depend on the distance measure from the average solution dissimilar to TOPSIS algorithm. As Compare to other MCDM algorithms, EDAS algorithm has required less calculations.

Thus, the EDAS algorithm is an enhanced form of the existing algorithms because it considers the conflicting attributes. Moreover, the best and worst alternative chosen by the EDAS algorithm remains the same as that with that of the TOPSIS algorithm signifies that the established algorithm is an enhanced version of existing algorithms. The recently developed EDAS technique is utilized to choose an appropriate drug to treat Coronavirus patients based on the average solution under PyPHF environment. We Compare TOPSIS algorithm and recently developed EDAS algorithm which shows that the best and worst alternative are same. The EDAS approach has more statistical simplicity and the potential to produce more accurate results. The evaluations of alternatives in EDAS algorithm depend on the distance measure from the average solution dissimilar to TOPSIS algorithm. As Compare to other MCDM algorithms, EDAS algorithm has required less calculations. Moreover, the best and worst alternative chosen by the EDAS algorithm remains the same as that with that of the TOPSIS algorithm signifies that the established algorithm is an enhanced version of existing algorithm.

8 Conclusion

Pythagorean probabilistic hesitant fuzzy set is an enhanced form of Pythagorean fuzzy set and probabilistic hesitant fuzzy set. It consider probabilistic hesitant information, so, there is no loss of information. Nowadays, selection of an best drug to treat COVID-19 patients has become serious issue between specialists and doctors. To tackle this issue, we have established a PyPHF-EDAS based aggregation information technique to tackle the uncertain information in multi-attribute emergency decision making situation of COVID-19. In addition using proposed technique, we developed an algorithm to tackle MADM problems. We developed a method to select an ideal drug from certain drugs with efficacy values for coronavirus disease. We established a multi attribute decision-making approach based on the EDAS method under Pythagorean probabilistic hesitant fuzzy information. We choose the EDAS method because it has a significant role in the decision-making problems especially when more conflict criteria exist in MCGDM problems. This method is based on PDAS and NDAS from Average solution. These two measures indicate the difference between each solution and the Average solution. Superior value of PDAS and inferior value of NDAS is considered the optimal choice. To study the hybrid structure of EDAS method with PyPHF, we get PyPHF-EDAS method. The aim of this manuscript is to present PyPHF-EDAS method based on PyPHF averaging aggregation operator. We used the knowledge of conventional EDAS algorithm and the PyPHF in managing with uncertain data to establish another decision-making algorithm and select an appropriate drug to treat Coronavirus ailment. The main benefit of established algorithm is that it takes the probabilistic information to each positive and negative hesitant membership degrees into account which give more details without any loss of information. The established algorithm has been signified with a medical diagnosis example (to select an appropriate drug to treat Coronavirus’s patients) to show the validity and effectiveness of our established technique under PyPHF information. Lastly, a comparative study has been considered between established algorithm and TOPSIS algorithm to shows the validity and applicability of the proposed technique.

In the future, we will establish the TODIM and VIKOR methodology based generalized aggregation information within the framework of Pythagorean probabilistic hesitant fuzzy information to tackle uncertain information in decision making problems.

Acknowledgements This work was supported by the Deanship of Scientific Research (DSR), King Abdulaziz University, Jeddah, under the grant No. (DF-192-980-1441). The authors, therefore, gratefully acknowledge DSR technical and financial support.

References

Ashraf S, Abdullah S (2020) Emergency decision support modeling for COVID-19 based on spherical fuzzy information. Int J Intell Syst 35(11):1601–1645
Ashraf S, Abdullah S, Almagrabi AO (2020) A new emergency response of spherical intelligent fuzzy decision process to diagnose of COVID19. Soft Comput. https://doi.org/10.1007/s00500-020-05287-8
