Annals of Optimization Theory and Practice

Volume 4, Number 1, 41-85

April 2021

DOI: 10.22121/AOTP.2021.287404.1069

SDM methods' configurations (2017-2019) and their application to a performance-based value assignment problem: A follow up study

Serdar Enginoğlu✉ • Tuğçe Aydın • Samet Memiş • Burak Arslan

Çanakkale Onsekiz Mart University, Turkey

✉ serdarenginoglu@gmail.com

(Received: February 22, 2021 / Accepted: April 16, 2021)

Abstract Being a follow-up study, this paper configures soft decision-making (SDM) methods (2017-2019), having been constructed with soft sets, soft matrices, and their fuzzy hybrid versions, to operate them in fuzzy parameterized fuzzy soft (*fpfs*) matrices space faithfully to the original. It then analyses the decision-making performances of the configured methods herein by using five test cases. Afterwards, it applies the methods, producing valid ranking order according to all the test cases, to the ranking of seven known noise-removal filters. This paper completes the configurations that allow the available methods (1999-2019) to operate in the *fpfs*-matrices space. Finally, the need for further research studies is discussed.

Keywords Fuzzy sets; Soft sets; Soft matrices; *fpfs*-matrices; Soft decision-making

1. Introduction

The present paper is a follow-up study of (Enginoğlu and Memiş, 2018a; Aydın and Enginoğlu, 2019; Enginoğlu and Öngel, 2020; Aydın and Enginoğlu, 2020; Enginoğlu and Aydın, 2021). The studies have configured the available soft decision-making (SDM) methods having been proposed between 1999-2016 and having been introduced by soft sets (SS) (Molodtsov, 1999; Çağman and Enginoğlu, 2010b), fuzzy soft sets (FSS) (Maji *et al*., 2001; Çağman *et al*., 2011b), fuzzy parameterized soft sets (FPSS) (Çağman *et al*., 2011a), fuzzy parameterized fuzzy soft sets (FPFSS or *fpfs*-sets) (Çağman *et al*., 2010), soft matrices (SM) (Çağman and Enginoğlu, 2010a), and fuzzy soft matrices (FSM) (Çağman and Enginoğlu, 2012). For the relationships between these concepts and further information, see (Enginoğlu *et al*., 2021). This paper completes the configurations that allow the available methods (1999-2019) (Guan, 2017; Zou *et al*., 2019; Alcantud and Mathew, 2017; Eraslan and Çağman, 2017; Liu *et al*., 2017; Taş *et al*., 2017; Alcantud and Torrecilles, 2018; Karaca and Taş, 2018; Liu and Liu, 2018; Pal, 2018; Porchelvi and Snekaa, 2018; Xiao, 2018; Aggarwal, 2019; Ma *et al*., 2019; Sandhiya and Selvakumari, 2019a; Sandhiya and Selvakumari, 2019b; Sharma and Singh, 2019; Wang and Qin, 2019; Zhang and Zhan, 2019; Riaz *et al*., 2018; Riaz and Hashmi, 2017; Riaz and Hashmi, 2018; Atagün *et al*., 2018; Mondal and De, 2018; Neog and Dutta, 2018; Kamacı *et al*., 2018) to operate in the *fpfs*-matrices space (Enginoğlu and Çağman, 2020).

The following tables provide some information about the considered SDM methods herein. Table 1, 2, and 3 show the abbreviated forms of the configured SDM methods herein employing single, double, and multiple *fpfs*-matrices and their spaces in which they have been first put forward, respectively.

Table 1. SDM methods employing single *fpfs*-matrix

Configured SDM Methods	Original Spaces of the Configured SDM Methods						Descriptions
FPFSM	FPFSS	FPSS	FSM	FSS	SM	SS	
G17(R)						✓	Guan 2017
LQP17(w)				✓			Liu, Qin, Pei 2017
TOD17				✓			Taş, Özgür, Demir 2017
KT18(R)						✓	Karaca, Taş 2018
KT18/2				✓			Karaca, Taş 2018
LL18(λ)				✓			Liu, Liu 2018
X18				✓			Xiao 2018
A19($R, w, \lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5$)						✓	Aggarwal 2019
MLQFG19				✓			Ma, Li, Qin, Fei, Gong 2019
WQ19				✓			Wang, Qin 2019
ZZ19(λ, γ)				✓			Zhang, Zhan 2019

Table 2. SDM methods employing double *fpfs*-matrices

Configured SDM Methods	Original Spaces of the Configured SDM Methods						Descriptions
FPFSM	FPFSS	FPSS	FSM	FSS	SM	SS	
RH17	✓						Riaz, Hashmi 2017
AKO18a					✓		Atagün, Kamacı, Oktay 2018
AKO18o					✓		Atagün, Kamacı, Oktay 2018
RH18	✓						Riaz, Hashmi 2018
X18/2				✓			Xiao 2018

Table 3. SDM methods employing multiple *fpfs*-matrices

Configured SDM Methods	Original Spaces of the Configured SDM Methods						Descriptions
FPFSM	FPFSS	FPSS	FSM	FSS	SM	SS	
AM17(R_1, R_2, …, R_t)				✓			Alcantud, Mathew 2017
AM17/2(R_1, R_2, …, R_t)				✓			Alcantud, Mathew 2017
AM17/3(λ, R_1, R_2, …, R_t)				✓			Alcantud, Mathew 2017
AM17/4(λ, R_1, R_2, …, R_t)				✓			Alcantud, Mathew 2017
AM17/5(λ, R_1, R_2, …, R_t)				✓			Alcantud, Mathew 2017
AM17/6(λ, R_1, R_2, …, R_t)				✓			Alcantud, Mathew 2017
EC17(λ)				✓			Eraslan, Çağman 2017
AT18(λ)				✓			Alcantud, Torrecillas 2018
KAS18aa					✓		Kamacı, Atagün, Sönmezoğlu 2018
KAS18aa/2					✓		Kamacı, Atagün, Sönmezoğlu 2018
MD18				✓			Mondal, De 2018
ND18				✓			Neog, Dutta 2018
P18				✓			Pal 2018
PS18				✓			Porchelvi, Snekaa 2018
RHF18	✓						Riaz, Hashmi, Farooq 2018
A19/2(R)				✓			Aggarwal 2019
SS19				✓			Sandkia, Selvakumari 2019
SS19/2				✓			Sandkia, Selvakumari 2019
SS19/3				✓			Sandkia, Selvakumari 2019
SS19/4				✓			Sandkia, Selvakumari 2019
SS19/5(w)				✓			Sharma, Singh 2019
ZCW19(δ, θ)						✓	Zou, Chen, Wang 2019

Section 2 presents some of the basic notions of *fpfs*-matrices to be required in the following sections. Section 3 configures the SDM methods provided in (Guan, 2017; Zou *et al*., 2019; Alcantud and Mathew, 2017; Eraslan and Çağman, 2017; Liu *et al*., 2017; Taş *et al*., 2017; Alcantud and Torrecilles, 2018; Karaca and Taş, 2018; Liu and Liu, 2018; Pal, 2018; Porchelvi and Snekaa, 2018; Xiao, 2018; Aggarwal, 2019; Ma *et al*., 2019; Sandhiya and Selvakumari, 2019a; Sandhiya and Selvakumari, 2019b; Sharma and Singh, 2019; Wang and Qin, 2019; Zhang and Zhan, 2019; Riaz *et al*., 2018; Riaz and Hashmi, 2017; Riaz and Hashmi, 2018; Atagün *et al*., 2018; Mondal and De, 2018; Neog and Dutta, 2018; Kamacı *et al*., 2018) to operate in the *fpfs*-matrices space (Enginoğlu and Çağman, 2020). Section 4 determines the methods producing a valid ranking order in all the test cases provided in (Enginoğlu *et al*., 2021) among the configured in the previous section. Section 5 applies the methods which accomplish all the tests to a performance-based value assignment (PVA) problem. Final Section discusses the need for further research studies.

2. Preliminaries

In this section, we present the concept of *fpfs*-matrices (Enginoğlu and Çağman, 2020) to be required in the next sections.

Definition 2.1. (Zadeh, 1965) Let E be a parameter set and μ be a function from E to $[0,1]$. Then, the set $\left\{{}^{\mu(x)}x \mid x \in E\right\}$, being the graphic of μ, is called a fuzzy set over E. Besides, $F(E)$ denotes the set of all the fuzzy sets over E.

Definition 2.2. (Çağman *et al.*, 2010) Let U be a universal set, $\mu \in F(E)$, and α be a function from μ to $F(U)$. Then, the set $\left\{\left({}^{\mu(x)}x, \alpha\left({}^{\mu(x)}x\right)\right) \mid x \in E\right\}$, being the graphic of α, is called a fuzzy parameterized fuzzy soft set (*fpfs*-set) parameterized via E over U (or briefly over U).

In the present paper, the set of all the *fpfs*-sets over U is denoted by $FPFS_E(U)$. In $FPFS_E(U)$, since the graph(α) and α generate each other uniquely, the notations are interchangeable. Therefore, as long as it causes no confusion, we denote an *fpfs*-set graph(α) by α.

Example 2.2. Let $E = \{x_1, x_2, x_3\}$ and $U = \{u_1, u_2, u_3, u_4\}$. Then,

$$\alpha = \{({}^{0.7}x_1, \{{}^{0.1}u_2, {}^{0.2}u_3, {}^{0.9}u_4\}), ({}^{0}x_2, \{{}^{0.2}u_1, {}^{0.8}u_2, {}^{0.5}u_4\}), ({}^{1}x_3, \{{}^{0.3}u_1, {}^{0.3}u_3, {}^{1}u_4\})\}$$

is an *fpfs*-set over U.

Definition 2.3. (Enginoğlu and Çağman, 2020) Let $\alpha \in FPFS_E(U)$. Then, $[a_{ij}]$ is called *fpfs*-matrix of α and is defined by

$$[a_{ij}] = \begin{bmatrix} a_{01} & a_{02} & a_{03} & \cdots & a_{0n} & \cdots \\ a_{11} & a_{12} & a_{13} & \cdots & a_{1n} & \cdots \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{m1} & a_{m2} & a_{m3} & \cdots & a_{mn} & \cdots \\ \vdots & \vdots & \vdots & \ddots & \vdots & \ddots \end{bmatrix}$$

such that for $i \in \{0,1,2,\cdots\}$ and $j \in \{1,2,\cdots\}$,

$$a_{ij} := \begin{cases} \mu(x_j), & i = 0 \\ \alpha\left({}^{\mu(x_j)}x_j\right)(u_i), & i \neq 0 \end{cases}$$

Here, if $|U| = m - 1$ and $|E| = n$, then $[a_{ij}]$ has order $m \times n$.

From now on, the set of all the *fpfs*-matrices parameterized via E over U is denoted by $FPFS_E[U]$.

Published by Firouzabad Institute of Higher Education, Firouzabad, Fars, Iran

Example 2.4. The *fpfs*-matrix of α provided in Example 2.2 is as follows:

$$[a_{ij}] = \begin{bmatrix} 0.7 & 0 & 1 \\ 0 & 0.2 & 0.3 \\ 0.1 & 0.8 & 0 \\ 0.2 & 0 & 0.3 \\ 0.9 & 0.5 & 1 \end{bmatrix}$$

Definition 2.5. (Enginoğlu and Çağman, 2020) Let $[a_{ij}]_{m\times n_1} \in FPFS_{E_1}[U]$, $[b_{ik}]_{m\times n_2} \in FPFS_{E_2}[U]$, and $[c_{ip}]_{m\times n_1 n_2} \in FPFS_{E_1\times E_2}[U]$ such that $p = n_2(j-1) + k$. For all i and p, if $c_{ip} \coloneqq \min\{a_{ij}, b_{ik}\}$, then $[c_{ip}]$ is called AND-product of $[a_{ij}]$ and $[b_{ik}]$ and is denoted by $[a_{ij}] \wedge [b_{ik}]$. For all i and p, if $c_{ip} \coloneqq \max\{a_{ij}, b_{ik}\}$, then $[c_{ip}]$ is called OR-product of $[a_{ij}]$ and $[b_{ik}]$ and is denoted by $[a_{ij}] \vee [b_{ik}]$.

Definition 2.6. Let $[s_{i1}] \in M_{(m-1)\times 1}(\mathbb{R})$ such that $m \geq 2$. Then, normalisation $[\hat{s}_{i1}]$ of $[s_{i1}]$ is defined by

$$\hat{s}_{i1} \coloneqq \begin{cases} \dfrac{s_{i1} - \min\limits_k s_{k1}}{\max\limits_k s_{k1} - \min\limits_k s_{k1}}, & \max\limits_k s_{k1} \neq \min\limits_k s_{k1} \\ 1, & \max\limits_k s_{k1} = \min\limits_k s_{k1} \end{cases}$$

3. Configurations of SDM Methods

This section configures the SDM methods constructed by soft sets (Guan, 2017; Zou et al., 2019; Karaca and Taş, 2018; Aggarwal, 2019), fuzzy soft sets (Alcantud and Mathew, 2017; Eraslan and Çağman, 2017; Liu *et al.*, 2017; Taş *et al.*, 2017; Alcantud and Torrecilles, 2018; Karaca and Taş, 2018; Liu and Liu, 2018; Pal, 2018; Porchelvi and Snekaa, 2018; Xiao, 2018; Aggarwal, 2019; Ma *et al.*, 2019; Sandhiya and Selvakumari, 2019a; Sandhiya and Selvakumari, 2019b; Sharma and Singh, 2019; Wang and Qin, 2019; Zhang, and Zhan, 2019; Mondal and De, 2018; Neog and Dutta, 2018), *fpfs*-sets (Riaz et al., 2018; Riaz and Hashmi, 2017; Riaz and Hashmi, 2018), and soft matrices (Atagün *et al.*, 2018; Mondal and De, 2018; Neog and Dutta, 2018; Kamacı *et al.*, 2018). From now on, $I_n = \{1,2,\cdots,n\}$ and $I_n^* = \{0,1,2,\cdots,n\}$.

Alcantud and Mathew (2017) have proposed six SDM methods based on fuzzy soft sets by using the arithmetic mean, geometric mean, Zadeh's fuzzy complement, Sugeno class of fuzzy complements, and Yager class of fuzzy complements (sic. Klir and Yuan, 1995). We configure the proposed methods therein as follows:

Algorithm 3.1. AM17$(\boldsymbol{R_1}, \boldsymbol{R_2}, \ldots, \boldsymbol{R_t})$

Step 1. Construct *fpfs*-matrices $[a^1_{ij_1}]_{m\times n_1}$, $[a^2_{ij_2}]_{m\times n_2}$, ..., $[a^t_{ij_t}]_{m\times n_t}$

Step 2. Determine indices set of undesirable parameters $R_k \subseteq I_{n_k}$, for all $k \in I_t$

Step 3. Obtain $[b^1_{ij_1}]_{m\times n_1}, [b^2_{ij_2}]_{m\times n_2}, \dots, [b^t_{ij_t}]_{m\times n_t}$ defined by

$$b^k_{ij_k} := \begin{cases} 1 - a^k_{ij_k}, & j_k \in R_k \\ a^k_{ij_k}, & j_k \notin R_k \end{cases}$$

such that $i \in I^*_{m-1}$, $j_k \in I_{n_k}$, and $k \in I_t$

Step 4. Obtain $[c_{ik}]_{m\times t}$ defined by

$$c_{ik} := \frac{1}{|I_{n_k}|}\sum_{j=1}^{|I_{n_k}|} b^k_{ij}$$

such that $i \in I^*_{m-1}$ and $k \in I_t$

Step 5. Apply Step 3 and 4 of A16 (Enginoğlu *et al*., 2021) to $[c_{ik}]$

Algorithm 3.2. AM17/2($\boldsymbol{R_1}, \boldsymbol{R_2}, \dots, \boldsymbol{R_t}$)

Step 1. Construct *fpfs*-matrices $[a^1_{ij_1}]_{m\times n_1}, [a^2_{ij_2}]_{m\times n_2}, \dots, [a^t_{ij_t}]_{m\times n_t}$

Step 2. Determine indices set of undesirable parameters $R_k \subseteq I_{n_k}$, for all $k \in I_t$

Step 3. Obtain $[b^1_{ij_1}]_{m\times n_1}, [b^2_{ij_2}]_{m\times n_2}, \dots, [b^t_{ij_t}]_{m\times n_t}$ defined by

$$b^k_{ij_k} := \begin{cases} 1 - a^k_{ij_k}, & j_k \in R_k \\ a^k_{ij_k}, & j_k \notin R_k \end{cases}$$

such that $i \in I^*_{m-1}$, $j_k \in I_{n_k}$, and $k \in I_t$

Step 4. Obtain $[c_{ik}]_{m\times t}$ defined by

$$c_{ik} := \left(\prod_{j=1}^{|I_{n_k}|} b^k_{ij}\right)^{\frac{1}{|I_{n_k}|}}$$

such that $i \in I^*_{m-1}$ and $k \in I_t$

Step 5. Apply Step 3 and 4 of A16 (Enginoğlu *et al*., 2021) to $[c_{ik}]$

Algorithm 3.3. AM17/3($\boldsymbol{\lambda}, \boldsymbol{R_1}, \boldsymbol{R_2}, \dots, \boldsymbol{R_t}$)

Step 1. Construct *fpfs*-matrices $[a^1_{ij_1}]_{m\times n_1}, [a^2_{ij_2}]_{m\times n_2}, \dots, [a^t_{ij_t}]_{m\times n_t}$

Step 2. Determine indices set of undesirable parameters $R_k \subseteq I_{n_k}$, for all $k \in I_t$

Step 3. For $\lambda \in (-1, \infty)$, obtain $\left[b_{ij_1}^1\right]_{m\times n_1}, \left[b_{ij_2}^2\right]_{m\times n_2}, \dots, \left[b_{ij_t}^t\right]_{m\times n_t}$ defined by

$$b_{ij_k}^k := \begin{cases} \dfrac{1 - a_{ij_k}^k}{1 + \lambda a_{ij_k}^k}, & j_k \in R_k \\ a_{ij_k}^k, & j_k \notin R_k \end{cases}$$

such that $i \in I_{m-1}^*$, $j_k \in I_{n_k}$, and $k \in I_t$

Step 4. Obtain $[c_{ik}]_{m\times t}$ defined by

$$c_{ik} := \frac{1}{|I_{n_k}|} \sum_{j=1}^{|I_{n_k}|} b_{ij}^k$$

such that $i \in I_{m-1}^*$ and $k \in I_t$

Step 5. Apply Step 3 and 4 of A16 (Enginoğlu *et al*., 2021) to $[c_{ik}]$

Algorithm 3.4. AM17/4($\boldsymbol{\lambda}, \boldsymbol{R_1}, \boldsymbol{R_2}, \dots, \boldsymbol{R_t}$)

Step 1. Construct *fpfs*-matrices $\left[a_{ij_1}^1\right]_{m\times n_1}, \left[a_{ij_2}^2\right]_{m\times n_2}, \dots, \left[a_{ij_t}^t\right]_{m\times n_t}$

Step 2. Determine indices set of undesirable parameters $R_k \subseteq I_{n_k}$, for all $k \in I_t$

Step 3. For $\lambda \in (-1, \infty)$, obtain $\left[b_{ij_1}^1\right]_{m\times n_1}, \left[b_{ij_2}^2\right]_{m\times n_2}, \dots, \left[b_{ij_t}^t\right]_{m\times n_t}$ defined by

$$b_{ij_k}^k := \begin{cases} \dfrac{1 - a_{ij_k}^k}{1 + \lambda a_{ij_k}^k}, & j_k \in R_k \\ a_{ij_k}^k, & j_k \notin R_k \end{cases}$$

such that $i \in I_{m-1}^*$, $j_k \in I_{n_k}$, and $k \in I_t$

Step 4. Obtain $[c_{ik}]_{m\times t}$ defined by

$$c_{ik} := \left(\prod_{j=1}^{|I_{n_k}|} b_{ij}^k \right)^{\frac{1}{|I_{n_k}|}}$$

such that $i \in I_{m-1}^*$ and $k \in I_t$

Step 5. Apply Step 3 and 4 of A16 (Enginoğlu *et al*., 2021) to $[c_{ik}]$

Algorithm 3.5. AM17/5($\boldsymbol{\lambda}, \boldsymbol{R_1}, \boldsymbol{R_2}, \dots, \boldsymbol{R_t}$)

Step 1. Construct *fpfs*-matrices $\left[a_{ij_1}^1\right]_{m\times n_1}, \left[a_{ij_2}^2\right]_{m\times n_2}, \dots, \left[a_{ij_t}^t\right]_{m\times n_t}$

Step 2. Determine indices set of undesirable parameters $R_k \subseteq I_{n_k}$, for all $k \in I_t$

Step 3. For $\lambda \in (0, \infty)$, obtain $\left[b^1_{ij_1}\right]_{m \times n_1}, \left[b^2_{ij_2}\right]_{m \times n_2}, \ldots, \left[b^t_{ij_t}\right]_{m \times n_t}$ defined by

$$b^k_{ij_k} := \begin{cases} \left(1 - \left(a^k_{ij_k}\right)^\lambda\right)^{\frac{1}{\lambda}}, & j_k \in R_k \\ a^k_{ij_k}, & j_k \notin R_k \end{cases}$$

such that $i \in I^*_{m-1}$, $j_k \in I_{n_k}$, and $k \in I_t$

Step 4. Obtain $[c_{ik}]_{m \times t}$ defined by

$$c_{ik} := \frac{1}{|I_{n_k}|} \sum_{j=1}^{|I_{n_k}|} b^k_{ij}$$

such that $i \in I^*_{m-1}$ and $k \in I_t$

Step 5. Apply Step 3 and 4 of A16 (Enginoğlu *et al.*, 2021) to $[c_{ik}]$

Algorithm 3.6. AM17/6($\boldsymbol{\lambda}, \boldsymbol{R_1}, \boldsymbol{R_2}, \ldots, \boldsymbol{R_t}$)

Step 1. Construct *fpfs*-matrices $\left[a^1_{ij_1}\right]_{m \times n_1}, \left[a^2_{ij_2}\right]_{m \times n_2}, \ldots, \left[a^t_{ij_t}\right]_{m \times n_t}$

Step 2. Determine indices set of undesirable parameters $R_k \subseteq I_{n_k}$, for all $k \in I_t$

Step 3. For $\lambda \in (0, \infty)$, obtain $\left[b^1_{ij_1}\right]_{m \times n_1}, \left[b^2_{ij_2}\right]_{m \times n_2}, \ldots, \left[b^t_{ij_t}\right]_{m \times n_t}$ defined by

$$b^k_{ij_k} := \begin{cases} \left(1 - \left(a^k_{ij_k}\right)^\lambda\right)^{\frac{1}{\lambda}}, & j_k \in R_k \\ a^k_{ij_k}, & j_k \notin R_k \end{cases}$$

such that $i \in I^*_{m-1}$ and $j_k \in I_{n_k}$, and $k \in I_t$

Step 4. Obtain $[c_{ik}]_{m \times t}$ defined by

$$c_{ik} := \left(\prod_{j=1}^{|I_{n_k}|} b^k_{ij}\right)^{\frac{1}{|I_{n_k}|}}$$

such that $i \in I^*_{m-1}$ and $k \in I_t$

Step 5. Apply Step 3 and 4 of A16 (Enginoğlu *et al.*, 2021) to $[c_{ik}]$

Published by Firouzabad Institute of Higher Education, Firouzabad, Fars, Iran

In (Eraslan and Çağman, 2017), the authors have introduced an SDM method via fuzzy soft sets by combining TOPSIS and Grey Relational Analysis. We configure the proposed method therein as follows:

Algorithm 3.7. EC17($\boldsymbol{\lambda}$)

Step 1. Construct *fpfs*-matrices $[a_{ij}^1]_{m\times n}$, $[a_{ij}^2]_{m\times n}$, …, $[a_{ij}^t]_{m\times n}$

Step 2. Obtain $[b_{kj}]_{t\times n}$ defined by $b_{kj} \coloneqq a_{0j}^k$ such that $k \in I_t$ and $j \in I_n$

Step 3. Obtain $[c_{kj}]_{t\times n}$ defined by

$$c_{kj} \coloneqq \begin{cases} \dfrac{b_{kj}}{\sqrt{\sum_{l=1}^{t} {b_{lj}}^2}}, & \displaystyle\sum_{l=1}^{t} {b_{lj}}^2 \neq 0 \\ \dfrac{1}{\sqrt{t}}, & \displaystyle\sum_{l=1}^{t} {b_{lj}}^2 = 0 \end{cases}$$

such that $k \in I_t$ and $j \in I_n$

Step 4. Obtain $[d_{j1}]_{n\times 1}$ defined by

$$d_{j1} \coloneqq \frac{1}{t}\sum_{k=1}^{t} c_{kj}, \quad j \in I_n$$

Step 5. Obtain $[e_{j1}]_{n\times 1}$ defined by

$$e_{j1} \coloneqq \frac{d_{j1}}{\sum_{l=1}^{n} d_{l1}}, \quad j \in I_n$$

Step 6. Obtain $[f_{ij}]_{(m-1)\times n}$ defined by

$$f_{ij} \coloneqq \frac{1}{t}\sum_{k=1}^{t} a_{ij}^k$$

such that $i \in I_{m-1}$ and $j \in I_n$

Step 7. Obtain $[g_{ij}]_{(m-1)\times n}$ defined by $g_{ij} \coloneqq e_{j1} f_{ij}$ such that $i \in I_{m-1}$ and $j \in I_n$

Step 8. Obtain $[g_{1j}^+]_{1\times n}$ and $[g_{1j}^-]_{1\times n}$ defined by

$$g_{1j}^+ \coloneqq \max_{i\in I_{m-1}} \{g_{ij}\} \quad \text{and} \quad g_{1j}^- \coloneqq \min_{i\in I_{m-1}} \{g_{ij}\}, \quad j \in I_n$$

Step 9. For $\lambda \in [0,1]$, obtain $[h_{ij}^+]_{(m-1)\times n}$ and $[h_{ij}^-]_{(m-1)\times n}$ defined by

$$h_{ij}^{+} := \begin{cases} \dfrac{\min\limits_{k\in I_{m-1}} \min\limits_{l\in I_n}\{|g_{1l}^{+}-g_{kl}|\} + \lambda \max\limits_{k\in I_{m-1}} \max\limits_{l\in I_n}\{|g_{1l}^{+}-g_{kl}|\}}{|g_{1j}^{+}-g_{ij}| + \lambda \max\limits_{k\in I_{m-1}} \max\limits_{l\in I_n}\{|g_{1l}^{+}-g_{kl}|\}}, & \max\limits_{k\in I_{m-1}} \max\limits_{l\in I_n}\{|g_{1l}^{+}-g_{kl}|\} \neq 0 \\ 1, & \max\limits_{k\in I_{m-1}} \max\limits_{l\in I_n}\{|g_{1l}^{+}-g_{kl}|\} = 0 \end{cases}$$

and

$$h_{ij}^{-} := \begin{cases} \dfrac{\min\limits_{k\in I_{m-1}} \min\limits_{l\in I_n}\{|g_{1l}^{-}-g_{kl}|\} + \lambda \max\limits_{k\in I_{m-1}} \max\limits_{l\in I_n}\{|g_{1l}^{-}-g_{kl}|\}}{|g_{1j}^{-}-g_{ij}| + \lambda \max\limits_{k\in I_{m-1}} \max\limits_{l\in I_n}\{|g_{1l}^{-}-g_{kl}|\}}, & \max\limits_{k\in I_{m-1}} \max\limits_{l\in I_n}\{|g_{1l}^{-}-g_{kl}|\} \neq 0 \\ 1, & \max\limits_{k\in I_{m-1}} \max\limits_{l\in I_n}\{|g_{1l}^{-}-g_{kl}|\} = 0 \end{cases}$$

such that $i \in I_{m-1}$ and $j \in I_n$

Step 10. Obtain $[s_{i1}^{+}]_{(m-1)\times 1}$ and $[s_{i1}^{-}]_{(m-1)\times 1}$ defined by

$$s_{i1}^{+} := \frac{1}{n}\sum_{j=1}^{n} h_{ij}^{+} \quad \text{and} \quad s_{i1}^{-} := \frac{1}{n}\sum_{j=1}^{n} h_{ij}^{-}, \quad i \in I_{m-1}$$

Step 11. Obtain the score matrix $[s_{i1}]_{(m-1)\times 1}$ defined by

$$s_{i1} := 1 - \frac{s_{i1}^{-}}{s_{i1}^{+} + s_{i1}^{-}}, \quad i \in I_{m-1}$$

Step 12. Obtain the decision set $\{{}^{\hat{s}_{k1}}u_k | u_k \in U\}$

Guan (2017) has suggested an SDM method based on soft sets to select a house. We configure the proposed method therein as follows:

Algorithm 3.8. G17($\boldsymbol{R}$)

Step 1. Construct an *fpfs*-matrix $[a_{ij}]_{m\times n}$

Step 2. Determine a set R of indices such that $R \subseteq I_n$

Step 3. Obtain $[b_{i1}]_{(m-1)\times 1}$ defined by

$$b_{i1} := \sum_{j\in R} a_{0j}a_{ij}, \quad i \in I_{m-1}$$

Step 4. Obtain $[c_{i1}]_{(m-1)\times 1}$ defined by

$$c_{i1} := \sum_{j=1}^{n} a_{0j}a_{ij}, \quad i \in I_{m-1}$$

Step 5. Obtain the set $V = \left\{u_i : b_{i1} = \max\limits_{k\in I_{m-1}} b_{k1}\right\}$

Step 6. Obtain the score matrix $[s_{i1}]_{(m-1)\times 1}$ defined by

$$s_{i1} := \begin{cases} c_{i1}, & u_i \in V \\ b_{i1}, & u_i \in U - V \end{cases}$$

such that $i \in I_{m-1}$

Step 7. Obtain the decision set $\{^{\hat{s}_{k1}}u_k | u_k \in U\}$

In (Liu *et al.*, 2017), the researchers have utilised fuzzy soft sets and ideal solution approaches. We configure the proposed method therein as follows:

Algorithm 3.9. LQP17($\boldsymbol{w}$)

Step 1. Construct an *fpfs*-matrix $[a_{ij}]_{m\times n}$

Step 2. Obtain $[b_{ij}]_{m\times n}$ defined by

$$b_{0j} := \begin{cases} \dfrac{a_{0j}}{\sum_{k=1}^{n} a_{0k}}, & \displaystyle\sum_{k=1}^{n} a_{0k} \neq 0 \\ \dfrac{1}{n}, & \displaystyle\sum_{k=1}^{n} a_{0k} = 0 \end{cases} \quad \text{and} \quad b_{ij} := a_{ij}$$

such that $i \in I_{m-1}$ and $j \in I_n$

Step 3. Construct the parameters' optimum solution matrix $w := [w_{1j}]_{1\times n}$ such that $0 \leq w_{1j} \leq 1$, for all $j \in I_n$

Step 4. Obtain $[c_{i1}]_{(m-1)\times 1}$ defined by

$$c_{i1} := \sum_{j=1}^{n} b_{0j} |w_{1j} - b_{ij}|, \quad i \in I_{m-1}$$

Step 5. Obtain the score matrix $[s_{i1}]_{(m-1)\times 1}$ defined by $s_{i1} := \max_k c_{k1} - c_{i1}$ such that $i \in I_{m-1}$

Step 6. Obtain the decision set $\{^{\hat{s}_{k1}}u_k | u_k \in U\}$

Riaz and Hashmi (2017) have benefited *fpfs*-sets in a problem about determining a student for an announced scholarship. We configure the proposed method therein as follows:

Algorithm 3.10. RH17

Step 1. Construct two *fpfs*-matrices $[a_{ij}]_{m\times n}$ and $[b_{ij}]_{m\times n}$

Step 2. Obtain $[c_{j1}]_{n\times 1}$ and $[d_{j1}]_{n\times 1}$ defined by

$$c_{j1} := \frac{1}{m-1}\sum_{i=1}^{m-1} a_{0j}a_{ij} \text{ and } d_{j1} := \frac{1}{m-1}\sum_{i=1}^{m-1} b_{0j}b_{ij}, \quad j \in I_n$$

Step 3. Obtain $[e_{i1}]_{(m-1)\times 1}$ and $[f_{i1}]_{(m-1)\times 1}$ defined by

$$e_{i1} := \frac{1}{n}\sum_{k=1}^{n} a_{ik}c_{k1} \quad \text{and} \quad f_{i1} := \frac{1}{n}\sum_{k=1}^{n} b_{ik}d_{k1}, \quad i \in I_{m-1}$$

Step 4. Obtain the score matrix $[s_{i1}]_{(m-1)\times 1}$ defined by $s_{i1} := e_{i1} + f_{i1} - e_{i1}f_{i1}$ such that $i \in I_{m-1}$

Step 5. Obtain the decision set $\{^{\hat{s}_{k1}}u_k | u_k \in U\}$

In (Taş *et al*., 2017), the authors have applied fuzzy soft sets to the stock management problem. We configure the proposed method therein as follows:

Algorithm 3.11. TOD17

TOD17 is the same as NRM16(I_n) (Enginoğlu *et al*., 2021) and KM11(I_n) (Enginoğlu and Öngel, 2020). Therefore, we prefer the notation KM11(I_n).

Atagün *et al.* (2018) have introduced soft distributive max-min decision-making methods via soft matrices. We configure the proposed methods therein as follows:

Algorithm 3.12. AKO18a

Step 1. Construct two *fpfs*-matrices $[a_{ij}]_{m\times n_1}$ and $[b_{ik}]_{m\times n_2}$

Step 2. Find AND-product *fpfs*-matrix $[c_{ip}]_{m\times n_1 n_2}$ of $[a_{ij}]$ and $[b_{ik}]$

Step 3. Find AND-product *fpfs*-matrix $[d_{ip}]_{m\times n_1 n_2}$ of $[b_{ik}]$ and $[a_{ij}]$

Step 4. Obtain $[e_{i1}]_{(m-1)\times 1}$ defined by

$$e_{i1} := \max_k \begin{cases} \min\limits_{p\in J_k}(c_{0p}c_{ip}), & J_k \neq \emptyset \\ 0, & J_k = \emptyset \end{cases}$$

such that $i \in I_{m-1}$, $k \in I_{n_1}$, and $J_k := \{p \mid \exists i, c_{0p}c_{ip} \neq 0, (k-1)n_2 < p \leq kn_2\}$

Step 5. Obtain $[f_{i1}]_{(m-1)\times 1}$ defined by

$$f_{i1} := \max_t \begin{cases} \min\limits_{p\in J_t}(d_{0p}d_{ip}), & J_t \neq \emptyset \\ 0, & J_t = \emptyset \end{cases}$$

Published by Firouzabad Institute of Higher Education, Firouzabad, Fars, Iran

such that $i \in I_{m-1}$, $t \in I_{n_2}$, and $J_t := \{p \mid \exists i, d_{0p}d_{ip} \neq 0, (t-1)n_1 < p \leq tn_1\}$

Step 6. Obtain the score matrix $[s_{i1}]_{(m-1)\times 1}$ defined by $s_{i1} := \max\{e_{i1}, f_{i1}\}$ such that $i \in I_{m-1}$

Step 7. Obtain the decision set $\{^{\hat{s}_{k1}}u_k | u_k \in U\}$

Algorithm 3.13. AKO18o

Step 1. Construct two *fpfs*-matrices $[a_{ij}]_{m\times n_1}$ and $[b_{ik}]_{m\times n_2}$

Step 2. Find OR-product *fpfs*-matrix $[c_{ip}]_{m\times n_1 n_2}$ of $[a_{ij}]$ and $[b_{ik}]$

Step 3. Find OR-product *fpfs*-matrix $[d_{ip}]_{m\times n_1 n_2}$ of $[b_{ik}]$ and $[a_{ij}]$

Step 4. Obtain $[e_{i1}]_{(m-1)\times 1}$ defined by

$$e_{i1} := \max_k \begin{cases} \min\limits_{p\in J_k}(c_{0p}c_{ip}), & J_k \neq \emptyset \\ 0, & J_k = \emptyset \end{cases}$$

such that $i \in I_{m-1}$, $k \in I_{n_1}$, and $J_k := \{p \mid \exists i, c_{0p}c_{ip} \neq 0, (k-1)n_2 < p \leq kn_2\}$

Step 5. Obtain $[f_{i1}]_{(m-1)\times 1}$ defined by

$$f_{i1} := \max_t \begin{cases} \min\limits_{p\in J_t}(d_{0p}d_{ip}), & J_t \neq \emptyset \\ 0, & J_t = \emptyset \end{cases}$$

such that $i \in I_{m-1}$, $t \in I_{n_2}$, and $J_t := \{p \mid \exists i, d_{0p}d_{ip} \neq 0, (t-1)n_1 < p \leq tn_1\}$

Step 6. Obtain the score matrix $[s_{i1}]_{(m-1)\times 1}$ defined by $s_{i1} := \max\{e_{i1}, f_{i1}\}$ such that $i \in I_{m-1}$

Step 7. Obtain the decision set $\{^{\hat{s}_{k1}}u_k | u_k \in U\}$

Here, AKO18a and AKO18o denote AKO18 with AND-product and AKO18 with OR-product, respectively. Moreover, SDM methods can also be constructed using products of *fpfs*-matrices and max-min, min-max, max-max, and min-min decision functions.

In (Alcantud and Torrecilles, 2018), the researchers have applied fuzzy soft sets containing multiple measurements in the selecting portfolio. We configure the proposed method therein as follows:

Algorithm 3.14. AT18(λ)

Step 1. Construct *fpfs*-matrices $[a_{ij}^1]_{m\times n}, [a_{ij}^2]_{m\times n}, \cdots, [a_{ij}^t]_{m\times n}$

Step 2. For $\lambda \in (0,1)$, obtain $[b_{ij}]_{m\times n}$ defined by

$$b_{ij} := \frac{1-\lambda}{\lambda}\sum_{k=1}^{t}(\lambda)^k a_{ij}^k$$

such that $i \in I_{m-1}^*$ and $j \in I_n$

Step 3. Apply Step 3 and 4 of A16 (Enginoğlu *et al*., 2021) to $[b_{ij}]$

In (Karaca and Taş, 2018), the scholars have suggested two SDM methods using soft sets and fuzzy soft sets for decision-making problem related to life and non-life insurances. We configure the proposed methods therein as follows:

Algorithm 3.15. KT18($\boldsymbol{R}$)

KT18(R) and MS10(R) (Enginoğlu and Öngel, 2020) are the same. Therefore, we prefer the notation MS10(R).

Algorithm 3.16. KT18/2($\boldsymbol{R}$)

KT18/2(R) and KM11(R) (Enginoğlu and Öngel, 2020) are the same. Therefore, we prefer the notation KM11(R).

Liu and Liu (2018) have proposed an SDM method using fuzzy soft sets based on the TOPSIS method with improved entropy weight. We configure the proposed method therein as follows:

Algorithm 3.17. LL18($\boldsymbol{\lambda}$)

Step 1. Construct an *fpfs*-matrix $[a_{ij}]_{m\times n}$

Step 2. Obtain $[b_{ik}^j]_{(m-1)\times(m-1)}$ defined by

$$b_{ik}^j := \begin{cases}\chi(a_{ij}, a_{kj}), & i \neq k \\ 0, & i = k\end{cases}, \qquad i,k \in I_{m-1}$$

such that

$$\chi(a_{ij}, a_{kj}) := \begin{cases}1, & a_{ij} \geq a_{kj} \\ 0, & a_{ij} < a_{kj}\end{cases}$$

Step 3. Obtain $[c_{1j}]_{1\times n}$ defined by

$$c_{1j} := \begin{cases}\dfrac{\sum_{i=1}^{m-1}\sum_{k=1}^{m-1} b_{ik}^j}{\sum_{i=1}^{m-1}\sum_{k=1}^{m-1}\sum_{l=1}^{n} b_{ik}^l}, & \displaystyle\sum_{i=1}^{m-1}\sum_{k=1}^{m-1}\sum_{l=1}^{n} b_{ik}^l \neq 0 \\ \dfrac{1}{n}, & \displaystyle\sum_{i=1}^{m-1}\sum_{k=1}^{m-1}\sum_{l=1}^{n} b_{ik}^l = 0\end{cases}, \qquad j \in I_n$$

Published by Firouzabad Institute of Higher Education, Firouzabad, Fars, Iran

Step 4. Obtain $[d_{ij}]_{(m-1)\times n}$ defined by

$$d_{ij} \coloneqq \sqrt{\sum_{k=1}^{m-1} \left(a_{ij} - a_{kj}\right)^2}$$

such that $i \in I_{m-1}$ and $j \in I_n$

Step 5. Obtain $[e_{1j}]_{1\times n}$ defined by

$$e_{1j} \coloneqq \sum_{i=1}^{m-1} d_{ij}, \quad j \in I_n$$

Step 6. Obtain $[f_{1j}]_{1\times n}$ defined by

$$f_{1j} \coloneqq -\frac{1}{\varepsilon + \ln(m-1)} \sum_{i=1}^{m-1} c_{1j} \frac{\varepsilon + d_{ij}}{\varepsilon + e_{1j}} \ln\left(c_{1j} \frac{\varepsilon + d_{ij}}{\varepsilon + e_{1j}}\right), \quad j \in I_n$$

Here, if $m = 1$, then $\frac{1}{\ln(m-1)}$ is undefined. Similarly, if $e_{1j} = 0$ or $d_{ij} = 0$, then $\ln\left(c_{1j} \frac{d_{ij}}{e_{1j}}\right)$ is undefined. To cope with this drawback, we modify them as $\frac{1}{\varepsilon+\ln(m-1)}$ and $\ln\left(c_{1j} \frac{\varepsilon+d_{ij}}{\varepsilon+e_{1j}}\right)$ such that $\varepsilon \ll 1$ is a positive constant, e.g., $\varepsilon = 0.0001$.

Step 7. Obtain $[g_{1j}]_{1\times n}$ defined by $g_{1j} \coloneqq 1 - f_{1j}$ such that $j \in I_n$

Step 8. Obtain $[h_{1j}]_{1\times n}$ defined by

$$h_{1j} \coloneqq \frac{g_{1j}}{\sum_{l=1}^{n} g_{1l}}, \quad j \in I_n$$

Step 9. For $\lambda \in [0,1]$, obtain $[v_{1j}]_{1\times n}$ defined by

$$v_{1j} \coloneqq \lambda a_{0j} + (1-\lambda) h_{1j}, \quad j \in I_n$$

Step 10. Obtain $[x_{ij}]_{(m-1)\times n}$ defined by $x_{ij} \coloneqq v_{1j} a_{ij}$ such that $i \in I_{m-1}$ and $j \in I_n$

Step 11. Obtain $[x_{ij}^{+}]_{1\times n}$ and $[x_{ij}^{-}]_{1\times n}$ defined by $x_{1j}^{+} \coloneqq \max_{i \in I_{m-1}} \{x_{ij}\}$ and $x_{1j}^{-} \coloneqq \min_{i \in I_{m-1}} \{x_{ij}\}$ such that $j \in I_n$

Step 12. Obtain $[s_{i1}^{+}]_{(m-1)\times 1}$ and $[s_{i1}^{-}]_{(m-1)\times 1}$ defined by

Published by Firouzabad Institute of Higher Education, Firouzabad, Fars, Iran

$$s_{i1}^{+} := \sqrt{\sum_{j=1}^{n}\left(x_{ij} - x_{1j}^{+}\right)^2}, \quad i \in I_{m-1}$$

and

$$s_{i1}^{-} := \sqrt{\sum_{j=1}^{n}\left(x_{ij} - x_{1j}^{-}\right)^2}, \quad i \in I_{m-1}$$

Step 13. Obtain the score matrix $[s_{i1}]_{(m-1)\times 1}$ defined by

$$s_{i1} := \begin{cases} \dfrac{s_{i1}^{-}}{s_{i1}^{+} + s_{i1}^{-}}, & s_{i1}^{+} + s_{i1}^{-} \neq 0 \\ 1, & s_{i1}^{+} + s_{i1}^{-} = 0 \end{cases}, \quad i \in I_{m-1}$$

Step 14. Obtain the decision set $\{^{\hat{s}_{k1}}u_k | u_k \in U\}$

In (Pal, 2018), the researcher has modified the SDM method provided in (Çağman *et al*., 2011b) for the multi-fuzzy soft sets. We configure the proposed method therein as follows:

Algorithm 3.18. P18

Step 1. Construct *fpfs*-matrices $[a_{ij}^1]_{m\times n}$, $[a_{ij}^2]_{m\times n}$, ..., $[a_{ij}^t]_{m\times n}$

Step 2. Obtain $[b_{ij}]_{m\times n}$ defined by

$$b_{ij} := \frac{1}{t}\sum_{k=1}^{t} a_{ij}^k$$

such that $i \in I_{m-1}^*$ and $j \in I_n$

Step 3. Apply CEC11 (Enginoğlu and Öngel, 2020) to $[b_{ij}]$

Porchelvi and Snekaa (2018) have suggested an SDM method using fuzzy soft sets for multi-criteria decision-making problems. We configure the proposed method therein as follows:

Algorithm 3.19. PS18

Step 1. Construct *fpfs*-matrices $[a_{ij}^1]_{m\times n}$, $[a_{ij}^2]_{m\times n}$, ..., $[a_{ij}^t]_{m\times n}$

Step 2. Obtain $[b_{ij}]_{m\times n}$ defined by

$$b_{ij} := \frac{1}{t}\sum_{k=1}^{t} a_{ij}^{k}$$

such that $i \in I_{m-1}^{*}$ and $j \in I_n$

Step 3. Obtain $[c_{ij}]_{(m-1)\times n}$ defined by $c_{ij} := b_{0j}b_{ij}$ such that $i \in I_{m-1}$ and $j \in I_n$

Step 4. Obtain $[d_{ij}]_{(m-1)\times n}$ defined by

$$d_{ij} := \begin{cases} \dfrac{c_{ij}}{\sum_{k=1}^{m-1} c_{kj}}, & \displaystyle\sum_{k=1}^{m-1} c_{kj} \neq 0 \\ \dfrac{1}{m-1}, & \displaystyle\sum_{k=1}^{m-1} c_{kj} = 0 \end{cases}$$

such that $i \in I_{m-1}$ and $j \in I_n$

Step 5. Obtain the score matrix $[s_{i1}]_{(m-1)\times 1}$ defined by

$$s_{i1} := \sum_{j=1}^{n} d_{ij}, \quad i \in I_{m-1}$$

Step 6. Obtain the decision set $\{{}^{\hat{s}_{k1}}u_k | u_k \in U\}$

In (Riaz and Hashmi, 2018), the researchers have modified the SDM method provided in (Çağman *et al.*, 2011a) to work with two *fpfs*-sets. We configure the proposed method therein as follows:

Algorithm 3.20. RH18

Step 1. Construct two *fpfs*-matrices $[a_{ij}]_{m\times n}$ and $[b_{ij}]_{m\times n}$

Step 2. Obtain $[c_{i1}]_{(m-1)\times 1}$ defined by

$$c_{i1} := \begin{cases} \dfrac{1}{\sum_{j=1}^{n} \operatorname{sgn}(a_{0j})}\displaystyle\sum_{j=1}^{n} a_{0j}a_{ij}, & \displaystyle\sum_{j=1}^{n} \operatorname{sgn}(a_{0j}) \neq 0 \\ \dfrac{1}{n}, & \displaystyle\sum_{j=1}^{n} \operatorname{sgn}(a_{0j}) = 0 \end{cases}, \quad i \in I_{m-1}$$

Step 3. Obtain $[d_{i1}]_{(m-1)\times 1}$ defined by

$$d_{i1} := \begin{cases} \dfrac{1}{\sum_{j=1}^{n} \operatorname{sgn}(b_{0j})} \displaystyle\sum_{j=1}^{n} b_{0j} b_{ij}, & \displaystyle\sum_{j=1}^{n} \operatorname{sgn}(b_{0j}) \neq 0 \\ \dfrac{1}{n}, & \displaystyle\sum_{j=1}^{n} \operatorname{sgn}(b_{0j}) = 0 \end{cases}, \quad i \in I_{m-1}$$

Step 4. Obtain the score matrix $[s_{i1}]_{(m-1)\times 1}$ defined by $s_{i1} := c_{i1} + d_{i1} - c_{i1}d_{i1}$ such that $i \in I_{m-1}$

Step 5. Obtain the decision set $\{{}^{\hat{s}_{k1}}u_k | u_k \in U\}$

Riaz *et al.* (2018) have propounded an SDM method based on the support sets of the considered *fpfs*-sets. We configure the proposed method therein as follows:

Algorithm 3.21. RHF18

Step 1. Construct *fpfs*-matrices $[a_{ij}^1]_{m\times n}$, $[a_{ij}^2]_{m\times n}$, …, $[a_{ij}^t]_{m\times n}$

Step 2. For $k \in I_t$, obtain $[b_{ij}^k]_{(m-1)\times n}$ defined by

$$b_{ij}^k := \begin{cases} 1, & a_{0j}^k a_{ij}^k > 0 \\ 0, & a_{0j}^k a_{ij}^k = 0 \end{cases}$$

such that $i \in I_{m-1}$ and $j \in I_n$

Step 3. For $k \in I_t$, obtain $[c_{i1}^k]_{(m-1)\times 1}$ defined by

$$c_{i1}^k := \sum_{j=1}^{n} b_{ij}^k, \quad i \in I_{m-1}$$

Step 4. Obtain the score matrix $[s_{i1}]_{(m-1)\times 1}$ defined by

$$s_{i1} := \sum_{k=1}^{t} c_{i1}^k, \quad i \in I_{m-1}$$

Step 5. Obtain the decision set $\{{}^{\hat{s}_{k1}}u_k | u_k \in U\}$

In (Xiao, 2018), the author has offered two SDM methods using hybrid fuzzy soft sets for medical diagnosis. We configure the proposed methods therein as follows:

Algorithm 3.22. X18

Step 1. Construct an *fpfs*-matrix $[a_{ij}]_{m\times n}$

Step 2. Obtain $[b_{ij}]_{(m-1)\times n}$ defined by

$$b_{ij} := \begin{cases} \dfrac{a_{0j}a_{ij}}{\sum_{k=1}^{m-1} a_{0j}a_{kj}}, & \displaystyle\sum_{k=1}^{m-1} a_{0j}a_{kj} \neq 0 \\ \dfrac{1}{m-1}, & \displaystyle\sum_{k=1}^{m-1} a_{0j}a_{kj} = 0 \end{cases}$$

such that $i \in I_{m-1}$ and $j \in I_n$

Step 3. Obtain $[c_{1j}]_{1\times n}$ defined by

$$c_{1j} := e^{-\sum_{i=1}^{m-1} b_{ij}\log_2(\varepsilon+b_{ij})}, \quad j \in I_n$$

Here, if $b_{ij} = 0$, then $\log_2(b_{ij})$ is undefined. To cope with this drawback, we modify it as $\log_2(\varepsilon + b_{ij})$ such that $\varepsilon \ll 1$ is a positive constant, e.g., $\varepsilon = 0.0001$.

Step 4. Obtain $[d_{1j}]_{1\times n}$ defined by

$$d_{1j} := \frac{c_{1j}}{\sum_{k=1}^{n} c_{1k}}, \quad j \in I_n$$

Step 5. Obtain $[e_{ij}]_{n\times n}$ defined by

$$e_{ij} := \begin{cases} 0.5, & i = j \text{ or } n = 2 \\ \dfrac{Var(d_{1i})}{Var(d_{1i}) + Var(d_{1j})}, & i \neq j, n \neq 2, \text{and } \mathrm{Var}(d_{1i}) + \mathrm{Var}(d_{1j}) \neq 0 \\ 0, & \text{otherwise} \end{cases}$$

such that $i, j \in I_n$ and

$$\mathrm{Var}(d_{1k}) := \mathrm{Var}(\{d_{11}, d_{12}, \ldots, d_{1(k-1)}, d_{1(k+1)}, \ldots, d_{1n}\}) = \frac{\sum_{\substack{i=1 \\ i\neq k}}^{n} \left(d_{1i} - \sum_{\substack{t=1 \\ t\neq k}}^{n} \frac{d_{1t}}{n-1}\right)^2}{n-1}$$

Step 6. Obtain $[f_{ij}]_{n\times n}$ defined by

$$f_{ij} := \frac{1}{n}\left(\sum_{k=1}^{n} (e_{ik} + e_{kj})\right) - 0.5, \quad i, j \in I_n$$

Step 7. Obtain $[g_{1j}]_{1\times n}$ defined by

$$g_{1j} := \frac{2}{n^2}\sum_{k=1}^{n} f_{jk}, \quad j \in I_n$$

Step 8. Obtain $[h_{1j}]_{1\times n}$ defined by $h_{1j} := g_{1j}d_{1j}$ such that $j \in I_n$

Step 9. Obtain $[v_{1j}]_{1\times n}$ defined by

$$v_{1j} := \frac{h_{1j}}{\sum_{k=1}^{n} h_{1k}}, \quad j \in I_n$$

Step 10. Obtain $[x_{ij}]_{(m-1)\times n}$ defined by $x_{ij} := b_{ij}(1 - v_{1j})$ such that $i \in I_{m-1}$ and $j \in I_n$

Step 11. Obtain $[y_{1j}]_{1\times n}$ defined by

$$y_{1j} := 1 - \sum_{i=1}^{m-1} x_{ij}, \quad j \in I_n$$

Step 12. Apply Step 8-10 of XWL14 (Enginoğlu *et al.*, 2021) to $[x_{ij}]$ and $[y_{1j}]$

Algorithm 3.23. X18/2

Step 1. Construct two *fpfs*-matrix $[a_{ij}]_{m\times n_1}$ and $[b_{ik}]_{m\times n_2}$

Step 2. Find AND-product *fpfs*-matrix $[c_{ip}]_{m\times n_1 n_2}$ of $[a_{ij}]$ and $[b_{ik}]$

Step 3. Apply X18 to $[c_{ip}]$

Aggarwal (2019) has proposed two SDM methods based on soft sets and fuzzy soft sets. We configure the proposed methods therein as follows:

Algorithm 3.24. A19($\boldsymbol{R}, \boldsymbol{w}, \boldsymbol{\lambda_1}, \boldsymbol{\lambda_2}, \boldsymbol{\lambda_3}, \boldsymbol{\lambda_4}, \boldsymbol{\lambda_5}$)

Step 1. Construct an *fpfs*-matrix $[a_{ij}]_{m\times n}$

Step 2. Construct $w := [w_{1j}]_{1\times n}$ such that $0 \leq w_{1j} \leq 1$, for all $\mathrm{j} \in \mathrm{I_n}$

Step 3. For $\lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5 \in \mathbb{R}$, obtain $[b_{ij}]_{m\times n}$ defined by

$$b_{ij} := \begin{cases} e^{-\left(\lambda_1 (a_{ij})^3 + \lambda_2 (a_{ij})^2 + \lambda_3 a_{ij} + \lambda_4\right)^{\lambda_5}}, & a_{ij} \geq w_{1j} \\ 0, & a_{ij} < w_{1j} \end{cases}$$

such that $i \in I_{m-1}^*$ and $j \in I_n$

Step 4. Obtain $[c_{ij}]_{m\times n}$ defined by $c_{ij} := a_{ij} b_{ij}$ such that $i \in I_{m-1}^*$ and $j \in I_n$

Step 5. Apply MS10(R) (Enginoğlu and Öngel, 2020) to $[c_{ij}]$ such that $R \subseteq I_n$

Algorithm 3.25. A19/2($\boldsymbol{R}$)

Step 1. Construct *fpfs*-matrices $[a^1_{ij_1}]_{m\times n_1}$, $[a^2_{ij_2}]_{m\times n_2}$, ..., $[a^t_{ij_t}]_{m\times n_t}$

Step 2. Determine a set R of indices such that $R \subseteq I_{n_1 n_2 \cdots n_t}$

Step 3. Obtain $[b_{ip}]_{m\times n_1 n_2 \cdots n_t}$ defined by

$$b_{ip} := \prod_{k=1}^{t} a^k_{ij_k}$$

such that $i \in I^*_{m-1}$ and $p = (j_1 - 1)n_2 n_3 \dots n_t + (j_2 - 1)n_3 n_4 \dots n_t + \cdots + j_t$

Step 4. Obtain the score matrix $[s_{i1}]_{(m-1)\times 1}$ defined by

$$s_{i1} := \max_{p\in R}\{b_{0p} b_{ip}\}$$

such that $i \in I_{m-1}$

Step 5. Obtain the decision set $\{^{\hat{s}_{k1}}u_k | u_k \in U\}$

In (Ma *et al.*, 2019), the authors have applied fuzzy soft sets to measure the similarity of the websites. We configure the proposed method therein as follows:

Algorithm 3.26. MLQFG19

MLQFG19 is the same as FJLL10/2m (Enginoğlu and Öngel, 2020). Therefore, we will prefer the notation FJLL10/2m.

Sandhiya and Selvakumari (2019a) have applied fuzzy soft sets to specify an eligible candidate for a company. We configure the proposed method therein as follows:

Algorithm 3.27. SS19

Step 1. Construct *fpfs*-matrices $[a^1_{ij}]_{m\times n}$, $[a^2_{ij}]_{m\times n}$, …, $[a^t_{ij}]_{m\times n}$

Step 2. Obtain $[b_{ij}]_{m\times n}$ defined by $b_{ij} := \underset{k\in I_t}{max}\{a^k_{ij}\}$ such that $i \in I^*_{m-1}$ and $j \in I_n$

Step 3. Obtain $[c_{ik}]_{(m-1)\times(m-1)}$ defined by

$$c_{ik} := \sum_{j=1}^{n} b_{0j}\chi(b_{ij}, b_{kj}), \quad i,k \in I_{m-1}$$

such that

$$\chi(b_{ij}, b_{kj}) := \begin{cases} 1, & b_{ij} \geq b_{kj} \\ 0, & b_{ij} < b_{kj} \end{cases}$$

Step 4. Obtain $[d_{i1}]_{(m-1)\times 1}$ defined by

$$d_{i1} := \sum_{k=1}^{m-1} c_{ik}, \quad i \in I_{m-1}$$

Step 5. Obtain $[e_{i1}]_{(m-1)\times 1}$ defined by

$$e_{i1} := \sum_{k=1}^{m-1} c_{ki}, \quad i \in I_{m-1}$$

Step 6. Obtain the score matrix $[s_{i1}]_{(m-1)\times 1}$ defined by

$$s_{i1} := d_{i1} + e_{i1}, \quad i \in I_{m-1}$$

Step 7. Obtain the decision set $\{^{\hat{s}_{i1}}u_k | u_k \in U\}$

In (Sandhiya and Selvakumari, 2019b), the scholars have applied fuzzy soft sets to a decision-making problem based on teaching evaluation performance. We configure the proposed methods therein as follows:

Algorithm 3.28. SS19/2

Step 1. Construct *fpfs*-matrices $[a_{ij}^1]_{m\times n}, [a_{ij}^2]_{m\times n}, \ldots, [a_{ij}^t]_{m\times n}$

Step 2. Obtain $[b_{ij}]_{m\times n}$ defined by

$$b_{ij} := \frac{1}{t}\sum_{k=1}^{t} a_{ij}^k$$

such that $i \in I_{m-1}^*$and $j \in I_n$

Step 3. Obtain $[c_{ij}]_{m\times n}$ defined by

$$c_{0j} := \begin{cases} \dfrac{b_{0j}}{\sum_{l=1}^{n} b_{0l}}, & \displaystyle\sum_{l=1}^{n} b_{0l} \neq 0 \\ \dfrac{1}{n}, & \displaystyle\sum_{l=1}^{n} b_{0l} = 0 \end{cases}$$

and

$$c_{ij} := \begin{cases} \dfrac{b_{ij}}{\max\limits_{l\in I_{m-1}} b_{lj}}, & \max\limits_{l\in I_{m-1}} b_{lj} \neq 0 \\ 1, & \max\limits_{l\in I_{m-1}} b_{lj} = 0 \end{cases}$$

Published by Firouzabad Institute of Higher Education, Firouzabad, Fars, Iran

such that $i \in I_{m-1}$ and $j \in I_n$

Step 4. Obtain the score matrix $[s_{i1}]_{(m-1)\times 1}$ defined by

$$s_{i1} \coloneqq \sum_{j=1}^{n} c_{0j}c_{ij}\,, \quad i \in I_{m-1}$$

Step 5. Obtain the decision set $\{^{\hat{s}_{k1}}u_k | u_k \in U\}$

Algorithm 3.29. SS19/3

Step 1. Construct *fpfs*-matrices $\left[a_{ij}^1\right]_{m\times n}$, $\left[a_{ij}^2\right]_{m\times n}$, ..., $\left[a_{ij}^t\right]_{m\times n}$

Step 2. Obtain $\left[b_{ij}\right]_{m\times n}$ defined by

$$b_{ij} \coloneqq \frac{1}{t}\sum_{k=1}^{t} a_{ij}^k$$

such that $i \in I_{m-1}^{*}$and $j \in I_n$

Step 3. Obtain $\left[c_{ij}\right]_{m\times n}$ defined by

$$c_{0j} \coloneqq \begin{cases} \dfrac{b_{0j}}{\sum_{l=1}^{n} b_{0l}}, & \displaystyle\sum_{l=1}^{n} b_{0l} \neq 0 \\ \dfrac{1}{n}, & \displaystyle\sum_{l=1}^{n} b_{0l} = 0 \end{cases}$$

and

$$c_{ij} \coloneqq \begin{cases} \dfrac{b_{ij}}{\sum_{l=1}^{m-1} b_{lj}}, & \displaystyle\sum_{l=1}^{m-1} b_{lj} \neq 0 \\ \dfrac{1}{m-1}, & \displaystyle\sum_{l=1}^{m-1} b_{lj} = 0 \end{cases}$$

such that $i \in I_{m-1}$ and $j \in I_n$

Step 4. Obtain the score matrix $[s_{i1}]_{(m-1)\times 1}$ defined by

$$s_{i1} \coloneqq \sum_{j=1}^{n} c_{0j}c_{ij}\,, \quad i \in I_{m-1}$$

Step 5. Obtain the decision set $\{^{\hat{s}_{k1}}u_k | u_k \in U\}$

Algorithm 3.30. SS19/4

Step 1. Construct *fpfs*-matrices $[a_{ij}^1]_{m\times n}$, $[a_{ij}^2]_{m\times n}$, ..., $[a_{ij}^t]_{m\times n}$

Step 2. Obtain $[b_{ij}]_{m\times n}$ defined by

$$b_{ij} := \frac{1}{t}\sum_{k=1}^{t} a_{ij}^k$$

such that $i \in I_{m-1}^*$ and $j \in I_n$

Step 3. Obtain $[c_{ij}]_{m\times n}$ defined by

$$c_{0j} := \begin{cases} \dfrac{b_{0j}}{\sum_{l=1}^{n} b_{0l}}, & \displaystyle\sum_{l=1}^{n} b_{0l} \neq 0 \\ \dfrac{1}{n}, & \displaystyle\sum_{l=1}^{n} b_{0l} = 0 \end{cases}$$

and

$$c_{ij} := \begin{cases} \dfrac{b_{ij} - \min\limits_{l\in I_{m-1}} b_{lj}}{\max\limits_{l\in I_{m-1}} b_{lj} - \min\limits_{l\in I_{m-1}} b_{lj}}, & \max\limits_{l\in I_{m-1}} b_{lj} \neq \min\limits_{l\in I_{m-1}} b_{lj} \\ 1, & \max\limits_{l\in I_{m-1}} b_{lj} = \min\limits_{l\in I_{m-1}} b_{lj} \end{cases}$$

such that $i \in I_{m-1}$ and $j \in I_n$

Step 4. Obtain the score matrix $[s_{i1}]_{(m-1)\times 1}$ defined by

$$s_{i1} := \sum_{j=1}^{n} c_{0j}c_{ij}, \quad i \in I_{m-1}$$

Step 5. Obtain the decision set $\{^{\hat{s}_{k1}}u_k | u_k \in U\}$

Sharma and Singh (2019) have examined the cleanliness ranking of public health centres using fuzzy soft sets. We configure the proposed method therein as follows:

Algorithm 3.31. SS19/5(w)

Step 1. Construct *fpfs*-matrices $[a_{ij}^1]_{m\times n}$, $[a_{ij}^2]_{m\times n}$, ..., $[a_{ij}^t]_{m\times n}$

Step 2. Obtain $[b_{kl}]_{t\times t}$ defined by

$$b_{kl} := \begin{cases} \dfrac{\sum_{j=1}^{n}\sum_{i=1}^{m-1} a_{0j}^k a_{ij}^k a_{0j}^l a_{ij}^l}{\sqrt{\sum_{j=1}^{n}\sum_{i=1}^{m-1}\left(a_{0j}^k a_{ij}^k\right)^2}\sqrt{\sum_{j=1}^{n}\sum_{i=1}^{m-1}\left(a_{0j}^l a_{ij}^l\right)^2}}, & \sqrt{\sum_{j=1}^{n}\sum_{i=1}^{m-1}\left(a_{0j}^k a_{ij}^k\right)^2}\sqrt{\sum_{j=1}^{n}\sum_{i=1}^{m-1}\left(a_{0j}^l a_{ij}^l\right)^2} \neq 0 \\ 0, & otherwise \end{cases}$$

such that $k, l \in I_t$

Step 3. Obtain $[c_{k1}]_{t\times 1}$ defined by

$$c_{k1} := \frac{\sum_{l=1, l\neq k}^{t} b_{kl}}{t-1}, \quad k \in I_t$$

Step 4. Obtain $[d_{k1}]_{t\times 1}$ defined by

$$d_{k1} := \begin{cases} \dfrac{c_{k1}}{\sum_{l=1}^{t} c_{l1}}, & \sum\limits_{l=1}^{t} c_{l1} \neq 0 \\ \dfrac{1}{t}, & \sum\limits_{l=1}^{t} c_{l1} = 0 \end{cases}, \quad k \in I_t$$

Step 5. Obtain $[e_{ij}]_{(m-1)\times n}$ defined by $e_{ij} := \sum_{k=1}^{t} d_{k1} a_{ij}^k$ such that $i \in I_{m-1}$ and $j \in I_n$

Step 6. Construct $w := [w_{1j}]_{1\times n}$ such that $0 \leq w_{1j} \leq 1$ and $\sum_{j=1}^{n} w_{1j} = 1$

Step 7. Obtain the score matrix $[s_{i1}]_{(m-1)\times 1}$ defined by $s_{i1} := \sum_{j=1}^{n} w_{1j} e_{ij}$ such that $i \in I_{m-1}$

Step 8. Obtain the decision set $\{^{\hat{s}_{k1}}u_k | u_k \in U\}$

In (Wang and Qin, 2019), the authors have provided an SDM approach using modal-style operators of fuzzy soft sets. We configure the proposed method therein as follows:

Algorithm 3.32. WQ19

Step 1. Construct an *fpfs*-matrix $[a_{ij}]_{m\times n}$

Step 2. Obtain $[b_{i1}]_{(m-1)\times 1}$ defined by

$$b_{i1} := \min_{j\in I_n}\{\chi_{ij}\}, \quad i \in I_{m-1}$$

such that

$$\chi_{ij} := \begin{cases} 1, & a_{0j} \leq a_{ij} \\ a_{ij}, & a_{0j} > a_{ij} \end{cases}$$

Step 3. Obtain $[c_{i1}]_{(m-1)\times 1}$ defined by

$$c_{i1} := \max_{j\in I_n}\{\min\{a_{0j}, a_{ij}\}\}, \quad i \in I_{m-1}$$

Step 4. Obtain the score matrix $[s_{i1}]_{(m-1)\times 1}$ defined by

$$s_{i1} := b_{i1} + c_{i1}, \quad i \in I_{m-1}$$

Step 5. Obtain the decision set $\{^{\hat{s}_{k1}}u_k | u_k \in U\}$

Zou *et al*. (2019) have constructed an SDM method based on soft sets and aggregation operators. We configure the proposed method therein as follows:

Algorithm 3.33. ZCW19($\boldsymbol{\delta}, \boldsymbol{\theta}$)

Step 1. Construct *fpfs*-matrices $[a_{ij}^1]_{m\times n}$, $[a_{ij}^2]_{m\times n}$, ..., $[a_{ij}^t]_{m\times n}$

Step 2. Obtain $[b_{ik}]_{(m-1)\times t}$ defined by

$$b_{ik} := \sum_{j=1}^{n} a_{0j}^k a_{ij}^k$$

such that $i \in I_{m-1}$ and $k \in I_t$

Step 3. Construct $\delta := [\delta_{1k}]_{1\times t}$ such that $0 \leq \delta_{1k} \leq 1$ and $\sum_{k=1}^{t} \delta_{1k} = 1$

Step 4. Obtain $[c_{ik}]_{(m-1)\times t}$ defined by $c_{ik} := t\delta_{1k} b_{ik}$ such that $i \in I_{m-1}$ and $k \in I_t$

Step 5. For all $i \in I_{m-1}$, obtain $[d_{1k}^i]_{1\times t}$ such that $[d_{1k}^i]$ denote the non-increasing-sorted elements of the i^{th} row of $[c_{ik}]$

Step 6. Construct $\theta := [\theta_{1k}]_{1\times t}$ such that $0 \leq \theta_{1k} \leq 1$ and $\sum_{k=1}^{t} \theta_{1k} = 1$

Step 7. Obtain the score matrix $[s_{i1}]_{(m-1)\times 1}$ defined by

$$s_{i1} := \sum_{k=1}^{t} \theta_{1k} d_{1k}^i, \qquad i \in I_{m-1}$$

Step 8. Obtain the decision set $\{^{\hat{s}_{k1}}u_k | u_k \in U\}$

In (Zhang and Zhan, 2019), the researchers have presented an SDM method modelling a company's recruitment scenario via fuzzy soft β-covering sets. We configure the proposed method therein as follows:

Algorithm 3.34. ZZ19($\boldsymbol{\lambda}, \boldsymbol{\gamma}$)

Step 1. Construct an *fpfs*-matrix $[a_{ij}]_{m\times n}$

Step 2. Construct $\gamma := [\gamma_{i1}]_{(m-1)\times 1}$ such that $0 \leq \gamma_{i1} \leq 1$, for all $i \in I_{m-1}$

Step 3. For $\lambda \in (0,1]$, obtain $[b_{ik}]_{(m-1)\times(m-1)}$ and $[c_{ik}]_{(m-1)\times(m-1)}$ defined by

Published by Firouzabad Institute of Higher Education, Firouzabad, Fars, Iran

$$b_{ik} := \begin{cases} \min\limits_{j\in R_k}\{a_{0j}a_{ij}\}, & R_k \neq \emptyset \\ 0, & R_k = \emptyset \end{cases}, \quad i,k \in I_{m-1}$$

and

$$c_{ik} := \begin{cases} \min\limits_{j\in R_i}\{a_{0j}a_{kj}\}, & R_i \neq \emptyset \\ 0, & R_i = \emptyset \end{cases}, \quad i,k \in I_{m-1}$$

such that $R_l := \{j : a_{lj} \geq \lambda\}$, for all $l \in I_{m-1}$

Step 4. Obtain $[d_{i1}]_{(m-1)\times 1}$ defined by

$$d_{i1} := \min_{k\in I_{m-1}} \{\max\{1 - b_{ki}, 1 - c_{ki}, \gamma_{k1}\}\}, \qquad i \in I_{m-1}$$

Step 5. Obtain $[e_{i1}]_{(m-1)\times 1}$ defined by

$$e_{i1} := \max_{k\in I_{m-1}} \{\min\{b_{ki}, c_{ki}, \gamma_{k1}\}\}, \qquad i \in I_{m-1}$$

Step 6. Obtain the score matrix $[s_{i1}]_{(m-1)\times 1}$ defined by $s_{i1} := \gamma_{i1} + d_{i1} + e_{i1}$ such that $i \in I_{m-1}$

Step 7. Obtain the decision set $\{^{\hat{s}_{k1}}u_k | u_k \in U\}$

Differently from the above methods ranking the alternatives, MD18 (Mondal and De, 2018) and ND18 (Neog and Dutta, 2018) rank the alternatives' sets. Therefore, in the following section, they have not been compared with the others.

In (Mondal and De, 2018), the authors have provided an SDM method using fuzzy soft sets. We configure the proposed method therein as follows:

Algorithm 3.35. MD18

MD18 and ND18 are the same. Therefore, we prefer the notation ND18.

In (Neog and Dutta, 2018), the authors have provided an application of fuzzy soft sets to decision-making. We configure the proposed method therein as follows:

Algorithm 3.36. ND18

Step 1. Construct *fpfs*-matrices $[a^1_{ij}]_{m\times n}$, $[a^2_{ij}]_{m\times n}$, ..., $[a^t_{ij}]_{m\times n}$ such that $U_1 \neq U_2 \neq \cdots \neq U_t$

Step 2. For $k \in I_t$, obtain $[b^k_{1j}]_{1\times n}$ defined by

$$b^k_{1j} := \frac{1}{t(m-1)} \sum_{i=1}^{m-1} a^k_{0j} a^k_{ij}, \quad j \in I_n$$

Step 3. Obtain the score matrix $[s_{k1}]_{t\times 1}$ defined by

$$s_{k1} := \sum_{j=1}^{n} b_{1j}^{k}, \quad k \in I_t$$

Step 4. Obtain the decision set $\left\{{}^{\hat{s}_{k1}}U_k | U_k \in \{U_1, U_2, \dots, U_t\}\right\}$

Moreover, KAS18aa and KAS18aa/2 (Kamacı *et al.*, 2018) have dealt with multi-case constructed by multi-expert via soft matrices. Therefore, in the following section, they have not been compared with the others. We configure the proposed method therein as follows:

Algorithm 3.37. KAS18aa

Step 1. Construct *fpfs*-matrices

$$[a_{i_1j_1}^{11}]_{(m_1+1)\times n_1}, [a_{i_2j_1}^{21}]_{(m_2+1)\times n_1}, \cdots, [a_{i_tj_1}^{t1}]_{(m_t+1)\times n_1},$$

$$[a_{i_1j_2}^{12}]_{(m_1+1)\times n_2}, [a_{i_2j_2}^{22}]_{(m_2+1)\times n_2}, \cdots, [a_{i_tj_2}^{t2}]_{(m_t+1)\times n_2},$$

$$\vdots$$

$$[a_{i_1j_s}^{1s}]_{(m_1+1)\times n_s}, [a_{i_2j_s}^{2s}]_{(m_2+1)\times n_s}, \cdots, [a_{i_tj_s}^{ts}]_{(m_t+1)\times n_s}$$

Step 2. Obtain

$$[b_{i_1j_1}^{11}] = [a_{0j_1}^{11} a_{i_1j_1}^{11}]_{m_1\times n_1}, [b_{i_2j_1}^{21}] = [a_{0j_1}^{21} a_{i_2j_1}^{21}]_{m_2\times n_1}, \cdots, [b_{i_tj_1}^{t1}] = [a_{0j_1}^{t1} a_{i_tj_1}^{t1}]_{m_t\times n_1},$$

$$[b_{i_1j_2}^{12}] = [a_{0j_1}^{11} a_{i_1j_2}^{12}]_{m_1\times n_2}, [b_{i_2j_2}^{22}] = [a_{0j_1}^{21} a_{i_2j_2}^{22}]_{m_2\times n_2}, \cdots, [b_{i_tj_2}^{t2}] = [a_{0j_1}^{t1} a_{i_tj_2}^{t2}]_{m_t\times n_2},$$

$$\vdots$$

$$[b_{i_1j_s}^{1s}] = [a_{0j_1}^{11} a_{i_1j_s}^{1s}]_{m_1\times n_s}, [b_{i_2j_s}^{2s}] = [a_{0j_1}^{21} a_{i_2j_s}^{2s}]_{m_2\times n_s}, \cdots, [b_{i_tj_s}^{ts}] = [a_{0j_1}^{t1} a_{i_tj_s}^{ts}]_{m_t\times n_s}$$

Step 3. Find AND-product *fs*-matrices (Çağman and Enginoğlu, 2012)

$[c_{j_1p}^{1}]_{n_1\times m_1m_2\dots m_t}$ of $[b_{i_1j_1}^{11}]^T, [b_{i_2j_1}^{21}]^T, \cdots, [b_{i_tj_1}^{t1}]^T$

$[c_{j_2p}^{2}]_{n_2\times m_1m_2\dots m_t}$ of $[b_{i_1j_2}^{12}]^T, [b_{i_2j_2}^{22}]^T, \cdots, [b_{i_tj_2}^{t2}]^T$

$\vdots$

$[c_{j_sp}^{s}]_{n_s\times m_1m_2\dots m_t}$ of $[b_{i_1j_s}^{1s}]^T, [b_{i_2j_s}^{2s}]^T, \cdots, [b_{i_tj_s}^{ts}]^T$

such that $p = (i_1 - 1)m_2m_3 \dots m_t + (i_2 - 1)m_3m_4 \dots m_t + \cdots + i_t$

Step 4. Find AND-product *fs*-matrix (Çağman and Enginoğlu, 2012) $[c_{pv}]_{m_1m_2\dots m_t\times n_1n_2\cdots n_s}$ of $[c^1_{j_1p}]^T$, $[c^2_{j_2p}]^T$, …, $[c^s_{j_sp}]^T$ such that $v=(j_1-1)n_2n_3\cdots n_s+(j_2-1)n_3n_4\cdots n_s+\cdots+j_s$

Step 5. Obtain the score matrix $[s_{p1}]_{m_1m_2\dots m_t\times 1}$ defined by

$$s_{p1}:=\frac{1}{n_1n_2\cdots n_s}\sum_{v=1}^{n_1n_2\cdots n_s}c_{pv}\,,\quad p\in I_{m_1m_2\dots m_t}$$

such that $p=(k_1-1)m_2m_3\dots m_t+(k_2-1)m_3m_4\dots m_t+\cdots+k_t$ and $(u_{k_1},u_{k_2},\cdots,u_{k_t})\in U_1\times U_2\times\cdots\times U_t$

Step 6. Obtain the decision set $\{{}^{\hat{s}_{k1}}u_k|u_k=(u_{k_1},u_{k_2},\cdots,u_{k_t})\}$

Algorithm 3.38. KAS18aa/2

Step 1. Construct *fpfs*-matrices $[a^1_{i_1j}]_{(m_1+1)\times n_1}$, $[a^2_{i_2j}]_{(m_2+1)\times n_1}$, …, $[a^t_{i_tj}]_{(m_t+1)\times n_1}$ and $[b^1_{i_1k}]_{(m_1+1)\times n_2}$, $[b^2_{i_2k}]_{(m_2+1)\times n_2}$, …, $[b^t_{i_tk}]_{(m_t+1)\times n_2}$

Step 2. Obtain

$[c^1_{i_1j}]=[a^1_{0j}a^1_{i_1j}]_{m_1\times n_1}$, $[c^2_{i_2j}]=[a^2_{0j}a^2_{i_2j}]_{m_2\times n_1}$, …, $[c^t_{i_tj}]=[a^t_{0j}a^t_{i_tj}]_{m_t\times n_1}$ and

$[d^1_{i_1k}]=[b^1_{0k}b^1_{i_1k}]_{m_1\times n_2}$, $[d^2_{i_2k}]=[b^2_{0k}b^2_{i_2k}]_{m_2\times n_2}$, …, $[d^t_{i_tk}]=[b^t_{0k}b^t_{i_tk}]_{m_t\times n_2}$

Step 3. Find AND-product *fs*-matrices (Çağman and Enginoğlu, 2012)

$[e_{jp}]_{n_1\times m_1m_2\dots m_t}$ of $[c^1_{i_1j}]^T$,$[c^2_{i_2j}]^T$, …, $[c^t_{i_tj}]^T$ and $[f_{kp}]_{n_2\times m_1m_2\dots m_t}$ of $[d^1_{i_1k}]^T$, $[d^2_{i_2k}]^T$, …, $[d^t_{i_tk}]^T$ such that $p=(i_1-1)m_2m_3\dots m_t+(i_2-1)m_3m_4\dots m_t+\cdots+i_t$

Step 4. Find AND-product *fs*-matrices (Çağman and Enginoğlu, 2012)

$[g_{pv}]_{m_1m_2\dots m_t\times n_1n_2}$ of $[e_{pj}]_{m_1m_2\dots m_t\times n_1}$ and $[f_{pk}]_{m_1m_2\dots m_t\times n_2}$ such that $v=(j-1)n_2+k$ and

$[h_{pv}]_{m_1m_2\dots m_t\times n_1n_2}$ of $[f_{pk}]_{m_1m_2\dots m_t\times n_2}$ and $[e_{pj}]_{m_1m_2\dots m_t\times n_1}$ such that $v=(k-1)n_1+j$

Step 5. Obtain $[x_{p1}]_{m_1m_2\dots m_t\times 1}$ defined by

$$x_{p1}:=\max_k\begin{cases}\min\limits_{v\in I_k}(g_{pv}), & J_k\neq\emptyset\\ 0, & J_k=\emptyset\end{cases}$$

such that $J_k := \{v \mid \exists p, g_{pv} \neq 0, (k-1)n_2 < v \leq kn_2\}$, $p \in I_{m_1 m_2 \dots m_t}$, and $k \in I_{n_1}$

Step 6. Obtain $[y_{p1}]_{m_1 m_2 \dots m_t \times 1}$ defined by

$$y_{p1} := \max_{t} \begin{cases} \min\limits_{v \in J_t}(h_{pv}), & J_t \neq \emptyset \\ 0, & J_t = \emptyset \end{cases}$$

such that $J_t := \{v \mid \exists p, h_{pv} \neq 0, (t-1)n_1 < v \leq tn_1\}$, $p \in I_{m_1 m_2 \dots m_t}$, and $t \in I_{n_2}$

Step 7. Obtain the score matrix $[s_{p1}]_{m_1 m_2 \dots m_t \times 1}$ defined by $s_{p1} := \max\{x_{p1}, y_{p1}\}$ such that $p \in I_{m_1 m_2 \dots m_t}$, $p = (k_1 - 1)m_2 m_3 \dots m_t + (k_2 - 1)m_3 m_4 \dots m_t + \cdots + k_t$, and $(u_{k_1}, u_{k_2}, \cdots, u_t) \in U_1 \times U_2 \times \cdots \times U_t$

Step 8. Obtain the decision set $\{^{\hat{s}_{k1}}u_k | u_k = (u_{k_1}, u_{k_2}, \cdots, u_t)\}$

Here, the notation aa in KAS18aa and KAS18aa/2 indicates that AND-product is used in the methods two times. KAS18 and KAS18/2 employ two of AND-product, OR-product, ANDNOT-product, ORNOT-product etc. If the methods use AND-product firstly and OR-product secondly, then the methods are denoted KAS18ao and KAS18ao/2.

4. Results of Test Cases

This section tests the configured SDM methods using five test cases provided in (Enginoğlu *et al*., 2021). Test cases are based on five situations in which an expert can naturally rank alternatives. If an SDM method produces the ranking order provided in a test case, it is said to accomplish the test case. Table 4 shows in which test cases the methods are successful. For example, while P18 pass in all the tests cases, PS18 only in Test Case 1, 2, and 5. For more details about the test cases, see (Enginoğlu *et al*., 2021).

In these test cases, the methods employing a single matrix work with the first *fpfs*-matrices in each test case. Similarly, the methods utilising double matrices employ the first two *fpfs*-matrices. Furthermore, the other methods utilise all the *fpfs*-matrices (Enginoğlu *et al*., 2021). Table 4 shows that 18 of 30 methods, namely EC17(λ), G17(R), LQP17(w), RH17, AKO18a, AKO18o, AT18(λ), LL18(λ), P18, RH18, A19$(R, w, \lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5)$, A19/2$(R)$, SS19/2, SS19/3, SS19/4, SS19/5(w), ZCW19(δ, θ), and ZZ19(λ, γ), pass all the tests. Moreover, the numbers of the passed tests are provided in the last column of Table 4.

Table 4. Success of the methods in the test cases

	Algorithms\Test Cases	Test Case 1	Test Case 2	Test Case 3	Test Case 4	Test Case 5	Passed Test's Numbers
1.	AM17($\emptyset, \emptyset, \emptyset$)	✓	✓			✓	3
2.	AM17/2($\emptyset, \emptyset, \emptyset$)	✓	✓			✓	3
3.	AM17/3($5, \emptyset, \emptyset, \emptyset$)	✓	✓			✓	3
4.	AM17/4($5, \emptyset, \emptyset, \emptyset$)	✓	✓			✓	3
5.	AM17/5($5, \emptyset, \emptyset, \emptyset$)	✓	✓			✓	3
6.	AM17/6($5, \emptyset, \emptyset, \emptyset$)	✓	✓			✓	3
7.	EC17(0.5)	✓	✓	✓	✓	✓	5
8.	G17(I_4)	✓	✓	✓	✓	✓	5
9.	LQP17([1 1 1 1])	✓	✓	✓	✓	✓	5
10.	RH17	✓	✓	✓	✓	✓	5
11.	AKO18a	✓	✓	✓	✓	✓	5
12.	AKO18o	✓	✓	✓	✓	✓	5
13.	AT18(0.95)	✓	✓	✓	✓	✓	5
14.	LL18(0.5)	✓	✓	✓	✓	✓	5
15.	P18	✓	✓	✓	✓	✓	5
16.	PS18	✓	✓			✓	3
17.	RH18	✓	✓	✓	✓	✓	5
18.	RHF18					✓	1
19.	X18	✓	✓			✓	3
20.	X18/2	✓	✓			✓	3
21.	A19(I_4, [0.4 0.4 0.4 0.4], $1, 1, 1, 1, 2$)	✓	✓	✓	✓	✓	5
22.	A19/2(I_{64})	✓	✓	✓	✓	✓	5
23.	SS19					✓	1
24.	SS19/2	✓	✓	✓	✓	✓	5
25.	SS19/3	✓	✓	✓	✓	✓	5
26.	SS19/4	✓	✓	✓	✓	✓	5
27.	SS19/5([0.25 0.25 0.25 0.25]) (Test 1,2,5) SS19/5([0.1818 0.2272 0.2727 0.3181]) (Test 3) SS19/5([0.3181 0.2727 0.2272 0.1818]) (Test 4)	✓	✓	✓	✓	✓	5
28.	WQ19		✓	✓	✓	✓	4
29.	ZCW19$\left(\left[\frac{1}{3}\ \frac{1}{3}\ \frac{1}{3}\right], \left[\frac{1}{3}\ \frac{1}{3}\ \frac{1}{3}\right]\right)$	✓	✓	✓	✓	✓	5
30.	ZZ19(0.5, $[0.35\ 0.45\ 0.55\ 0.65]^T$) (Test 1) ZZ19(0.5, $[0.65\ 0.55\ 0.45\ 0.35]^T$) (Test 2) ZZ19(0.5, $[0.25\ 0.25\ 0.25\ 0.25]^T$) (Test 3, 4) ZZ19(0.5, $[0.5\ 0.5\ 0.5\ 0.5]^T$) (Test 5)	✓	✓	✓	✓	✓	5
	Total	27	28	19	19	30	18 (5)

5. An Application of Some of the Configured Methods to a PVA Problem

This section ranks the noise-removal filters provided in (Erkan and Gökrem, 2018; Srinivasan and Ebenezer, 2007; Esakkirajan *et al*., 2012; Toh and Isa, 2010; Tang *et al*., 2016; Erkan *et al*., 2018; Enginoğlu *et al*., 2019a), obtained by the configured methods herein. Therefore, firstly, we present the results of the filters in (Enginoğlu *et al*., 2019a) produced by the quality metrics Peak Signal-to-Noise Ratio (PSNR), Structural Similarity (SSIM) (Wang *et al*., 2004), and Visual Information Fidelity (VIF) (Sheikh and Bovik, 2006) for 20 traditional images at noise density occurring between 10% and 90% in Table 5, 6, and 7, respectively. Moreover, the bold values in the tables signify the filters with the best performance.

Table 5. Mean-PSNR results for the 20 traditional images with different noise densities

Filters/Noise Densities	10%	20%	30%	40%	50%	60%	70%	80%	90%
BPDF	36.98	33.54	31.03	28.88	26.82	24.60	21.98	17.74	10.51
DBAIN	37.52	34.29	31.96	29.83	27.86	25.89	23.90	21.55	18.55
MDBUTMF	36.80	32.18	29.02	28.48	28.81	28.34	26.95	23.42	15.29
NAFSMF	36.08	33.27	31.49	30.15	29.02	27.96	26.82	25.47	22.34
DAMF	39.58	36.33	34.14	32.45	30.99	29.64	28.28	26.69	24.35
AWMF	36.34	35.00	33.83	32.69	31.47	30.14	28.68	26.99	24.70
ARmF	40.04	37.12	35.14	33.53	31.99	30.45	28.86	27.08	24.74

Table 6. Mean-SSIM results for the 20 traditional images with different noise densities

Filters/Noise Densities	10%	20%	30%	40%	50%	60%	70%	80%	90%
BPDF	0.9783	0.9536	0.9229	0.8838	0.8323	0.7634	0.6680	0.5096	0.2585
DBAIN	0.9796	0.9584	0.9315	0.8968	0.8520	0.7949	0.7213	0.6265	0.4966
MDBUTMF	0.9774	0.9197	0.8117	0.7973	0.8399	0.8410	0.8025	0.7023	0.3566
NAFSMF	0.9748	0.9504	0.9248	0.8973	0.8666	0.8320	0.7910	0.7357	0.6190
DAMF	0.9854	0.9699	0.9516	0.9303	0.9051	0.8748	0.8368	0.7846	0.6964
AWMF	0.9728	0.9622	0.9484	0.9315	0.9098	0.8816	0.8437	0.7904	0.7028
ARmF	0.9868	0.9735	0.9581	0.9400	0.9173	0.8880	0.8491	0.7947	0.7056

Table 7. Mean-VIF results for the 20 traditional images with different noise densities

Filters/Noise Densities	10%	20%	30%	40%	50%	60%	70%	80%	90%
BPDF	0.8188	0.6858	0.5659	0.4564	0.3529	0.2541	0.1614	0.0783	0.0334
DBAIN	0.8548	0.7319	0.6179	0.5119	0.4095	0.3128	0.2229	0.1365	0.0635
MDBUTMF	0.8272	0.6713	0.5044	0.4420	0.4310	0.3978	0.3302	0.2212	0.0730
NAFSMF	0.7902	0.6751	0.5828	0.5030	0.4307	0.3604	0.2897	0.2129	0.1226
DAMF	0.8787	0.7816	0.6943	0.6162	0.5437	0.4731	0.3998	0.3096	0.1913
AWMF	0.7896	0.7366	0.6789	0.6181	0.5533	0.4833	0.4066	0.3129	0.1928
ARmF	0.8832	0.7975	0.7210	0.6474	0.5741	0.4974	0.4158	0.3182	0.1955

In this PVA problem, the alternatives are indicated as $u_1 :=$ "BPDF", $u_2 :=$ "DBAIN", $u_3 :=$ "MDBUTMF", $u_4 :=$ "NAFSMF", $u_5 :=$ "DAMF", $u_6 :=$ "AWMF", and $u_7 :=$ "ARmF" such that $U = \{u_1, u_2, u_3, u_4, u_5, u_6, u_7\}$. Moreover, the parameters are denoted by $x_1 :=$ "SPN ratio 10%", $x_2 :=$ "SPN ratio 20%", $x_3 :=$ "SPN ratio 30%", $x_4 :=$ "SPN ratio 40%", $x_5 :=$ "SPN ratio 50%", $x_6 :=$ "SPN ratio 60%", $x_7 :=$ "SPN ratio 70%", $x_8 :=$ "SPN ratio 80%", and $x_9 :=$ "SPN ratio 90%" such that $E = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9\}$.

Suppose that the noise removal performances of the filters at high noise densities are more significant than at the other densities. In such a case, it is anticipated that the membership degrees at high noise densities are greater than at the other noise densities. In other words, the first rows of the *fpfs*-matrices are considered to be $[0.1\ 0.2\ 0.3\ 0.4\ 0.5\ 0.6\ 0.7\ 0.8\ 0.9]$ herein. Furthermore, while the SSIM and VIF values are in the interval $[0,1]$, the PSNR values are not. Hence, the PSNR values are normalised via the maximum value provided in Table 5 to construct the *fpfs*-matrix $[a_{ij}]$. Thus, Table 5, 6, and 7 can be represented with *fpfs*-matrices $[a_{ij}]_{8\times 9}$, $[b_{ij}]_{8\times 9}$, and $[c_{ij}]_{8\times 9}$ as follows:

$$[a_{ij}] := \begin{bmatrix} 0.1 & 0.2 & 0.3 & 0.4 & 0.5 & 0.6 & 0.7 & 0.8 & 0.9 \\ 0.9236 & 0.8377 & 0.7750 & 0.7213 & 0.6698 & 0.6144 & 0.5490 & 0.4431 & 0.2625 \\ 0.9371 & 0.8564 & 0.7982 & 0.7450 & 0.6958 & 0.6466 & 0.5969 & 0.5382 & 0.4633 \\ 0.9191 & 0.8037 & 0.7248 & 0.7113 & 0.7195 & 0.7078 & 0.6731 & 0.5849 & 0.3819 \\ 0.9011 & 0.8309 & 0.7865 & 0.7530 & 0.7248 & 0.6983 & 0.6698 & 0.6361 & 0.5579 \\ 0.9885 & 0.9073 & 0.8526 & 0.8104 & 0.7740 & 0.7403 & 0.7063 & 0.6666 & 0.6081 \\ 0.9076 & 0.8741 & 0.8449 & 0.8164 & 0.7860 & 0.7527 & 0.7163 & 0.6741 & 0.6169 \\ 1.0000 & 0.9271 & 0.8776 & 0.8374 & 0.7990 & 0.7605 & 0.7208 & 0.6763 & 0.6179 \end{bmatrix}$$

$$[b_{ij}] := \begin{bmatrix} 0.1 & 0.2 & 0.3 & 0.4 & 0.5 & 0.6 & 0.7 & 0.8 & 0.9 \\ 0.9783 & 0.9536 & 0.9229 & 0.8838 & 0.8323 & 0.7634 & 0.6680 & 0.5096 & 0.2585 \\ 0.9796 & 0.9584 & 0.9315 & 0.8968 & 0.8520 & 0.7949 & 0.7213 & 0.6265 & 0.4966 \\ 0.9774 & 0.9197 & 0.8117 & 0.7973 & 0.8399 & 0.8410 & 0.8025 & 0.7023 & 0.3566 \\ 0.9748 & 0.9504 & 0.9248 & 0.8973 & 0.8666 & 0.8320 & 0.7910 & 0.7357 & 0.6190 \\ 0.9854 & 0.9699 & 0.9516 & 0.9303 & 0.9051 & 0.8748 & 0.8368 & 0.7846 & 0.6964 \\ 0.9728 & 0.9622 & 0.9484 & 0.9315 & 0.9098 & 0.8816 & 0.8437 & 0.7904 & 0.7028 \\ 0.9868 & 0.9735 & 0.9581 & 0.9400 & 0.9173 & 0.8880 & 0.8491 & 0.7947 & 0.7056 \end{bmatrix}$$

and

Published by Firouzabad Institute of Higher Education, Firouzabad, Fars, Iran

$$[c_{ij}] \coloneqq \begin{bmatrix} 0.1 & 0.2 & 0.3 & 0.4 & 0.5 & 0.6 & 0.7 & 0.8 & 0.9 \\ 0.8188 & 0.6858 & 0.5659 & 0.4564 & 0.3529 & 0.2541 & 0.1614 & 0.0783 & 0.0334 \\ 0.8548 & 0.7319 & 0.6179 & 0.5119 & 0.4095 & 0.3128 & 0.2229 & 0.1365 & 0.0635 \\ 0.8272 & 0.6713 & 0.5044 & 0.4420 & 0.4310 & 0.3978 & 0.3302 & 0.2212 & 0.0730 \\ 0.7902 & 0.6751 & 0.5828 & 0.5030 & 0.4307 & 0.3604 & 0.2897 & 0.2129 & 0.1226 \\ 0.8787 & 0.7816 & 0.6943 & 0.6162 & 0.5437 & 0.4731 & 0.3998 & 0.3096 & 0.1913 \\ 0.7896 & 0.7366 & 0.6789 & 0.6181 & 0.5533 & 0.4833 & 0.4066 & 0.3129 & 0.1928 \\ 0.8832 & 0.7975 & 0.7210 & 0.6474 & 0.5741 & 0.4974 & 0.4158 & 0.3182 & 0.1955 \end{bmatrix}$$

Eight of the SDM methods having passed all the test cases, namely G17(I_9), LQP17([1 1 1 1 1 1 1 1 1]), LL18(0.5), A19(I_9, [0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4], 1, 1, 1, 1, 2), and ZZ19(0.5, $[0.6440\ \ 0.6975\ \ 0.6918\ \ 0.7287\ \ 0.7838\ \ 0.7766\ \ 0.8018]^T$), employ only an *fpfs*-matrix. Similarly, RH17, AKO18a, AKO18o, and RH18 utilise two *fpfs*-matrices, and EC17(0.5), AT18(0.05), P18, A19/2(I_{729}), SS19/2, SS19/3, SS19/4, SS19/5([0.0222 0.0444 0.0667 0.0889 0.1111 0.1333 0.1556 0.1778 0.2000]), and ZCW19([1/3 1/3 1/3], [1/3 1/3 1/3]) work with multiple *fpfs*-matrices.

Secondly, we apply the SDM methods to the aforesaid *fpfs*-matrices $[a_{ij}]_{8\times9}$, $[b_{ij}]_{8\times9}$, and $[c_{ij}]_{8\times9}$. The decision sets and ranking orders produced by these SDM methods are manifested in Table 8 and 9, respectively. The last column in Table 9 shows the number of the methods producing the same ranking order. The results provided in Table 8 are obtained by MATLAB R2020b using the aforesaid *fpfs*-matrices.

Table 8. Decision sets produced by SDM methods (in the event of more-importance-attached noise removal performance at high noise densities)

Algorithms	Matrices	Decision Sets
G17(I_9)	$[a_{ij}]$	$\{^{0}$BPDF, $^{0.3904}$DBAIN, $^{0.4110}$MDBUTMF, $^{0.6751}$NAFSMF, $^{0.9152}$DAMF, $^{0.9376}$AWMF, ^{1}ARmF$\}$
LQP17([1 1 1 1 1 1 1 1 1])	$[a_{ij}]$	$\{^{0}$BPDF, $^{0.3904}$DBAIN, $^{0.4110}$MDBUTMF, $^{0.6751}$NAFSMF, $^{0.9152}$DAMF, $^{0.9376}$AWMF, ^{1}ARmF$\}$
LL18(0.5)	$[a_{ij}]$	$\{^{0}$BPDF, $^{0.4464}$DBAIN, $^{0.4054}$MDBUTMF, $^{0.7352}$NAFSMF, $^{0.9273}$DAMF, $^{0.9305}$AWMF, ^{1}ARmF$\}$
A19(I_9, [0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4], 1, 1, 1, 1, 2)	$[a_{ij}]$	$\{^{0}$BPDF, $^{0.1118}$DBAIN, $^{0.0266}$MDBUTMF, $^{0.3943}$NAFSMF, $^{0.8381}$DAMF, $^{0.9731}$AWMF, ^{1}ARmF$\}$
ZZ19(0.5, $[0.6440\ 0.6975\ 0.6918\ 0.7287\ 0.7838\ 0.7766\ 0.8018]^T$)	$[a_{ij}]$	$\{^{0}$BPDF, $^{0.3388}$DBAIN, $^{0.3026}$MDBUTMF, $^{0.5365}$NAFSMF, $^{0.8856}$DAMF, $^{0.8397}$AWMF, ^{1}ARmF$\}$
RH17	$[a_{ij}]$, $[b_{ij}]$	$\{^{0}$BPDF, $^{0.3941}$DBAIN, $^{0.3715}$MDBUTMF, $^{0.6784}$NAFSMF, $^{0.9287}$DAMF, $^{0.9463}$AWMF, ^{1}ARmF$\}$
AKO18a	$[a_{ij}]$, $[b_{ij}]$	$\{^{0.3230}$BPDF, $^{0.6830}$DBAIN, ^{0}MDBUTMF, $^{0.4536}$NAFSMF, $^{0.9728}$DAMF, $^{0.5026}$AWMF, ^{1}ARmF$\}$
AKO18o	$[a_{ij}]$, $[b_{ij}]$	$\{^{0}$BPDF, $^{0.2219}$DBAIN, $^{0.5604}$MDBUTMF, $^{0.7193}$NAFSMF, $^{0.9520}$DAMF, $^{0.9807}$AWMF, ^{1}ARmF$\}$
RH18	$[a_{ij}]$, $[b_{ij}]$	$\{^{0}$BPDF, $^{0.4242}$DBAIN, $^{0.3926}$MDBUTMF, $^{0.7137}$NAFSMF, $^{0.9384}$DAMF, $^{0.9581}$AWMF, ^{1}ARmF$\}$
EC17(0.5)	$[a_{ij}]$, $[b_{ij}]$, $[c_{ij}]$	$\{^{0}$BPDF, $^{0.3234}$DBAIN, $^{0.2366}$MDBUTMF, $^{0.4325}$NAFSMF, $^{0.8968}$DAMF, $^{0.8261}$AWMF, ^{1}ARmF$\}$
AT18(0.05)	$[a_{ij}]$, $[b_{ij}]$, $[c_{ij}]$	$\{^{0}$BPDF, $^{0.4073}$DBAIN, $^{0.4180}$MDBUTMF, $^{0.7009}$NAFSMF, $^{0.9244}$DAMF, $^{0.9504}$AWMF, ^{1}ARmF$\}$
P18	$[a_{ij}]$, $[b_{ij}]$, $[c_{ij}]$	$\{^{0.7430}$BPDF, $^{0.8280}$DBAIN, $^{0.8356}$MDBUTMF, $^{0.8867}$NAFSMF, $^{0.9782}$DAMF, $^{0.9796}$AWMF, ^{1}ARmF$\}$

Published by Firouzabad Institute of Higher Education, Firouzabad, Fars, Iran

Algorithms	Matrices	Decision Sets
A19/2(I_{729})	$[a_{ij}]$, $[b_{ij}]$, $[c_{ij}]$	$\{^{0}$BPDF, $^{0.1610}$DBAIN, $^{0.4178}$MDBUTMF, $^{0.4396}$NAFSMF, $^{0.8899}$DAMF, $^{0.9494}$AWMF, ^{1}ARmF$\}$
SS19/2	$[a_{ij}]$, $[b_{ij}]$, $[c_{ij}]$	$\{^{0}$BPDF, $^{0.3440}$DBAIN, $^{0.4016}$MDBUTMF, $^{0.6020}$NAFSMF, $^{0.9258}$DAMF, $^{0.9423}$AWMF, ^{1}ARmF$\}$
SS19/3	$[a_{ij}]$, $[b_{ij}]$, $[c_{ij}]$	$\{^{0}$BPDF, $^{0.3678}$DBAIN, $^{0.3892}$MDBUTMF, $^{0.6355}$NAFSMF, $^{0.9357}$DAMF, $^{0.9522}$AWMF, ^{1}ARmF$\}$
SS19/4	$[a_{ij}]$, $[b_{ij}]$, $[c_{ij}]$	$\{^{0}$BPDF, $^{0.3260}$DBAIN, $^{0.3535}$MDBUTMF, $^{0.5379}$NAFSMF, $^{0.9115}$DAMF, $^{0.9046}$AWMF, ^{1}ARmF$\}$
SS19/5([0.0222 0.0444 0.0667 0.0889 0.1111 0.1333 0.1556 0.1778 0.2000])	$[a_{ij}]$, $[b_{ij}]$, $[c_{ij}]$	$\{^{0}$BPDF, $^{0.3536}$DBAIN, $^{0.3830}$MDBUTMF, $^{0.6089}$NAFSMF, $^{0.9280}$DAMF, $^{0.9414}$AWMF, ^{1}ARmF$\}$
ZCW19$\left(\left[\frac{1}{3}\ \frac{1}{3}\ \frac{1}{3}\right], \left[\frac{1}{3}\ \frac{1}{3}\ \frac{1}{3}\right]\right)$	$[a_{ij}]$, $[b_{ij}]$, $[c_{ij}]$	$\{^{0}$BPDF, $^{0.3523}$DBAIN, $^{0.3832}$MDBUTMF, $^{0.6067}$NAFSMF, $^{0.9278}$DAMF, $^{0.9411}$AWMF, ^{1}ARmF$\}$

The ranking orders in Table 9 demonstrate that the ranking orders of G17(I_9), LQP17([1 1 1 1 1 1 1 1 1]), AKO18o, AT18(0.05), P18, A19/2(I_{729})**,** SS19/2, SS19/3, SS19/5([0.0222 0.0444 0.0667 0.0889 0.1111 0.1333 0.1556 0.1778 0.2000]), and ZCW19$\left(\left[\frac{1}{3}\ \frac{1}{3}\ \frac{1}{3}\right], \left[\frac{1}{3}\ \frac{1}{3}\ \frac{1}{3}\right]\right)$ are the same.

Moreover, LL18(0.5), A19(I_9, [0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4], 1, 1, 1, 1, 2), RH17, and RH18 produces the same ranking orders just as ZZ19(0.5, [0.6440 0.6975 0.6918 0.7287 0.7838 0.7766 0.8018]^{T}) and EC17(0.5) do. On the other hands, the ranking order of AKO18a is more incoherent than the others.

The results manifest that the decision-making skills of the SDM methods herein are almost the same except AKO18a, and they agree that ARmF performs better than the other filters according to their SPN removal performance. Furthermore, the SDM methods except AKO18a agree that BPDF displays the minimum SPN removal performance compared to the others.

Table 9. Ranking orders produced by SDM methods (in the event of more-importance-attached noise removal performance at high noise densities)

Algorithms	Ranking Orders	Frequency
G17(I_9)	BPDF≺DBAIN≺MDBUTMF≺NAFSMF≺DAMF≺AWMF≺ARmF	10
LQP17([1 1 1 1 1 1 1 1 1])	BPDF≺DBAIN≺MDBUTMF≺NAFSMF≺DAMF≺AWMF≺ARmF	10
LL18(0.5)	BPDF≺MDBUTMF≺DBAIN≺NAFSMF≺DAMF≺AWMF≺ARmF	4
A19(I_9, [0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4], 1, 1, 1, 1, 2)	BPDF≺MDBUTMF≺DBAIN≺NAFSMF≺DAMF≺AWMF≺ARmF	4
ZZ19(0.5, [0.6440 0.6975 0.6918 0.7287 0.7838 0.7766 0.8018]^{T})	BPDF≺MDBUTMF≺DBAIN≺NAFSMF≺AWMF≺DAMF≺ARmF	2
RH17	BPDF≺MDBUTMF≺DBAIN≺NAFSMF≺DAMF≺AWMF≺ARmF	4
AKO18a	MDBUTMF≺BPDF≺NAFSMF≺AWMF≺DBAIN≺DAMF≺ARmF	1
AKO18o	BPDF≺DBAIN≺MDBUTMF≺NAFSMF≺DAMF≺AWMF≺ARmF	10
RH18	BPDF≺MDBUTMF≺DBAIN≺NAFSMF≺DAMF≺AWMF≺ARmF	4
EC17(0.5)	BPDF≺MDBUTMF≺DBAIN≺NAFSMF≺AWMF≺DAMF≺ARmF	2

Published by Firouzabad Institute of Higher Education, Firouzabad, Fars, Iran

Algorithms	Ranking Orders	Frequency
AT18(0.05)	BPDF≺DBAIN≺MDBUTMF≺NAFSMF≺DAMF≺AWMF≺ARmF	10
P18	BPDF≺DBAIN≺MDBUTMF≺NAFSMF≺DAMF≺AWMF≺ARmF	10
A19/2(I_{729})	BPDF≺DBAIN≺MDBUTMF≺NAFSMF≺DAMF≺AWMF≺ARmF	10
SS19/2	BPDF≺DBAIN≺MDBUTMF≺NAFSMF≺DAMF≺AWMF≺ARmF	10
SS19/3	BPDF≺DBAIN≺MDBUTMF≺NAFSMF≺DAMF≺AWMF≺ARmF	10
SS19/4	BPDF≺DBAIN≺MDBUTMF≺NAFSMF≺AWMF≺DAMF≺ARmF	1
SS19/5([0.0222 0.0444 0.0667 0.0889 0.1111 0.1333 0.1556 0.1778 0.2000])	BPDF≺DBAIN≺MDBUTMF≺NAFSMF≺DAMF≺AWMF≺ARmF	10
ZCW19$\left(\left[\frac{1}{3}\ \frac{1}{3}\ \frac{1}{3}\right],\left[\frac{1}{3}\ \frac{1}{3}\ \frac{1}{3}\right]\right)$	BPDF≺DBAIN≺MDBUTMF≺NAFSMF≺DAMF≺AWMF≺ARmF	10

On the other hand, assume that the noise removal performances of the filters at low noise densities are more significant than at the higher densities. In such a case, it is anticipated that the membership degrees at low noise densities are greater than at the higher noise densities. In other words, the first rows of the *fpfs*-matrices are considered to be $[0.9\ 0.8\ 0.7\ 0.6\ 0.5\ 0.4\ 0.3\ 0.2\ 0.1]$ herein. Therefore, Table 5, 6, and 7 can be represented with *fpfs*-matrices $[d_{ij}]_{8\times 9}$, $[e_{ij}]_{8\times 9}$, and $[f_{ij}]_{8\times 9}$ as follows:

$$[d_{ij}] \coloneqq \begin{bmatrix} 0.9 & 0.8 & 0.7 & 0.6 & 0.5 & 0.4 & 0.3 & 0.2 & 0.1 \\ 0.9236 & 0.8377 & 0.7750 & 0.7213 & 0.6698 & 0.6144 & 0.5490 & 0.4431 & 0.2625 \\ 0.9371 & 0.8564 & 0.7982 & 0.7450 & 0.6958 & 0.6466 & 0.5969 & 0.5382 & 0.4633 \\ 0.9191 & 0.8037 & 0.7248 & 0.7113 & 0.7195 & 0.7078 & 0.6731 & 0.5849 & 0.3819 \\ 0.9011 & 0.8309 & 0.7865 & 0.7530 & 0.7248 & 0.6983 & 0.6698 & 0.6361 & 0.5579 \\ 0.9885 & 0.9073 & 0.8526 & 0.8104 & 0.7740 & 0.7403 & 0.7063 & 0.6666 & 0.6081 \\ 0.9076 & 0.8741 & 0.8449 & 0.8164 & 0.7860 & 0.7527 & 0.7163 & 0.6741 & 0.6169 \\ 1.0000 & 0.9271 & 0.8776 & 0.8374 & 0.7990 & 0.7605 & 0.7208 & 0.6763 & 0.6179 \end{bmatrix}$$

$$[e_{ij}] \coloneqq \begin{bmatrix} 0.9 & 0.8 & 0.7 & 0.6 & 0.5 & 0.4 & 0.3 & 0.2 & 0.1 \\ 0.9783 & 0.9536 & 0.9229 & 0.8838 & 0.8323 & 0.7634 & 0.6680 & 0.5096 & 0.2585 \\ 0.9796 & 0.9584 & 0.9315 & 0.8968 & 0.8520 & 0.7949 & 0.7213 & 0.6265 & 0.4966 \\ 0.9774 & 0.9197 & 0.8117 & 0.7973 & 0.8399 & 0.8410 & 0.8025 & 0.7023 & 0.3566 \\ 0.9748 & 0.9504 & 0.9248 & 0.8973 & 0.8666 & 0.8320 & 0.7910 & 0.7357 & 0.6190 \\ 0.9854 & 0.9699 & 0.9516 & 0.9303 & 0.9051 & 0.8748 & 0.8368 & 0.7846 & 0.6964 \\ 0.9728 & 0.9622 & 0.9484 & 0.9315 & 0.9098 & 0.8816 & 0.8437 & 0.7904 & 0.7028 \\ 0.9868 & 0.9735 & 0.9581 & 0.9400 & 0.9173 & 0.8880 & 0.8491 & 0.7947 & 0.7056 \end{bmatrix}$$

and

Published by Firouzabad Institute of Higher Education, Firouzabad, Fars, Iran

$$[f_{ij}] \coloneqq \begin{bmatrix} 0.9 & 0.8 & 0.7 & 0.6 & 0.5 & 0.4 & 0.3 & 0.2 & 0.1 \\ 0.8188 & 0.6858 & 0.5659 & 0.4564 & 0.3529 & 0.2541 & 0.1614 & 0.0783 & 0.0334 \\ 0.8548 & 0.7319 & 0.6179 & 0.5119 & 0.4095 & 0.3128 & 0.2229 & 0.1365 & 0.0635 \\ 0.8272 & 0.6713 & 0.5044 & 0.4420 & 0.4310 & 0.3978 & 0.3302 & 0.2212 & 0.0730 \\ 0.7902 & 0.6751 & 0.5828 & 0.5030 & 0.4307 & 0.3604 & 0.2897 & 0.2129 & 0.1226 \\ 0.8787 & 0.7816 & 0.6943 & 0.6162 & 0.5437 & 0.4731 & 0.3998 & 0.3096 & 0.1913 \\ 0.7896 & 0.7366 & 0.6789 & 0.6181 & 0.5533 & 0.4833 & 0.4066 & 0.3129 & 0.1928 \\ 0.8832 & 0.7975 & 0.7210 & 0.6474 & 0.5741 & 0.4974 & 0.4158 & 0.3182 & 0.1955 \end{bmatrix}$$

Thirdly, we apply the SDM methods to the *fpfs*-matrices $[d_{ij}]_{8\times 9}$, $[e_{ij}]_{8\times 9}$, and $[f_{ij}]_{8\times 9}$. The decision sets and ranking orders generated by the SDM methods are provided in Table 10 and 11, respectively. The last column in Table 11 shows the number of the methods producing the same ranking order. The results provided in Table 10 are obtained by MATLAB R2020b using the last-mentioned *fpfs*-matrices.

Table 10. Decision sets produced by SDM methods (in the event of more-importance-attached noise removal performance at low noise densities)

Algorithms	Matrices	Decision Sets
G17(I_9)	$[d_{ij}]$	$\{{}^{0}\text{BPDF}, {}^{0.2543}\text{DBAIN}, {}^{0.1251}\text{MDBUTMF}, {}^{0.3098}\text{NAFSMF}, {}^{0.8371}\text{DAMF}, {}^{0.6796}\text{AWMF}, {}^{1}\text{ARmF}\}$
LQP17([1 1 1 1 1 1 1 1 1])	$[d_{ij}]$	$\{{}^{0}\text{BPDF}, {}^{0.2543}\text{DBAIN}, {}^{0.1251}\text{MDBUTMF}, {}^{0.3098}\text{NAFSMF}, {}^{0.8371}\text{DAMF}, {}^{0.6796}\text{AWMF}, {}^{1}\text{ARmF}\}$
LL18(0.5)	$[d_{ij}]$	$\{{}^{0}\text{BPDF}, {}^{0.2326}\text{DBAIN}, {}^{0.1246}\text{MDBUTMF}, {}^{0.2875}\text{NAFSMF}, {}^{0.8175}\text{DAMF}, {}^{0.5752}\text{AWMF}, {}^{1}\text{ARmF}\}$
A19(I_9, [0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4], 1, 1, 1, 1, 2)	$[d_{ij}]$	$\{{}^{0.0201}\text{BPDF}, {}^{0.0453}\text{DBAIN}, {}^{0.0144}\text{MDBUTMF}, {}^{0}\text{NAFSMF}, {}^{0.5978}\text{DAMF}, {}^{0.0042}\text{AWMF}, {}^{1}\text{ARmF}\}$
ZZ19(0.5, [0.6440 0.6975 0.6918 0.7287 0.7838 0.7766 0.8018]^T)	$[d_{ij}]$	$\{{}^{0}\text{BPDF}, {}^{0.3388}\text{DBAIN}, {}^{0.3026}\text{MDBUTMF}, {}^{0.5365}\text{NAFSMF}, {}^{0.8856}\text{DAMF}, {}^{0.8397}\text{AWMF}, {}^{1}\text{ARmF}\}$
RH17	$[d_{ij}]$, $[e_{ij}]$	$\{{}^{0.0369}\text{BPDF}, {}^{0.2986}\text{DBAIN}, {}^{0}\text{MDBUTMF}, {}^{0.3738}\text{NAFSMF}, {}^{0.8649}\text{DAMF}, {}^{0.7381}\text{AWMF}, {}^{1}\text{ARmF}\}$
AKO18a	$[d_{ij}]$, $[e_{ij}]$	$\{{}^{0}\text{BPDF}, {}^{0.5283}\text{DBAIN}, {}^{0.2695}\text{MDBUTMF}, {}^{0.8046}\text{NAFSMF}, {}^{0.9792}\text{DAMF}, {}^{0.9937}\text{AWMF}, {}^{1}\text{ARmF}\}$
AKO18o	$[d_{ij}]$, $[e_{ij}]$	$\{{}^{0.2022}\text{BPDF}, {}^{0.2500}\text{DBAIN}, {}^{0.1691}\text{MDBUTMF}, {}^{0.0735}\text{NAFSMF}, {}^{0.5772}\text{DAMF}, {}^{0}\text{AWMF}, {}^{1}\text{ARmF}\}$
RH18	$[d_{ij}]$, $[e_{ij}]$	$\{{}^{0.}\text{BPDF}, {}^{0.2872}\text{DBAIN}, {}^{0.0347}\text{MDBUTMF}, {}^{0.3974}\text{NAFSMF}, {}^{0.8674}\text{DAMF}, {}^{0.7635}\text{AWMF}, {}^{1}\text{ARmF}\}$
EC17(0.5)	$[d_{ij}]$, $[e_{ij}]$, $[f_{ij}]$	$\{{}^{0}\text{BPDF}, {}^{0.3234}\text{DBAIN}, {}^{0.2366}\text{MDBUTMF}, {}^{0.4325}\text{NAFSMF}, {}^{0.8968}\text{DAMF}, {}^{0.8261}\text{AWMF}, {}^{1}\text{ARmF}\}$
AT18(0.05)	$[d_{ij}]$, $[e_{ij}]$, $[f_{ij}]$	$\{{}^{0}\text{BPDF}, {}^{0.2683}\text{DBAIN}, {}^{0.1623}\text{MDBUTMF}, {}^{0.3639}\text{NAFSMF}, {}^{0.8469}\text{DAMF}, {}^{0.7272}\text{AWMF}, {}^{1}\text{ARmF}\}$
P18	$[d_{ij}]$, $[e_{ij}]$, $[f_{ij}]$	$\{{}^{0.8722}\text{BPDF}, {}^{0.9091}\text{DBAIN}, {}^{0.8766}\text{MDBUTMF}, {}^{0.9041}\text{NAFSMF}, {}^{0.9823}\text{DAMF}, {}^{0.9595}\text{AWMF}, {}^{1}\text{ARmF}\}$
A19/2(I_{729})	$[d_{ij}]$, $[e_{ij}]$, $[f_{ij}]$	$\{{}^{0.2577}\text{BPDF}, {}^{0.5105}\text{DBAIN}, {}^{0.2761}\text{MDBUTMF}, {}^{0}\text{NAFSMF}, {}^{0.9119}\text{DAMF}, {}^{0.0171}\text{AWMF}, {}^{1}\text{ARmF}\}$
SS19/2	$[d_{ij}]$, $[e_{ij}]$, $[f_{ij}]$	$\{{}^{0}\text{BPDF}, {}^{0.2988}\text{DBAIN}, {}^{0.1558}\text{MDBUTMF}, {}^{0.3640}\text{NAFSMF}, {}^{0.8787}\text{DAMF}, {}^{0.7807}\text{AWMF}, {}^{1}\text{ARmF}\}$
SS19/3	$[d_{ij}]$, $[e_{ij}]$, $[f_{ij}]$	$\{{}^{0}\text{BPDF}, {}^{0.3031}\text{DBAIN}, {}^{0.1765}\text{MDBUTMF}, {}^{0.3869}\text{NAFSMF}, {}^{0.8829}\text{DAMF}, {}^{0.7975}\text{AWMF}, {}^{1}\text{ARmF}\}$
SS19/4	$[d_{ij}]$, $[e_{ij}]$, $[f_{ij}]$	$\{{}^{0}\text{BPDF}, {}^{0.2921}\text{DBAIN}, {}^{0.0326}\text{MDBUTMF}, {}^{0.1840}\text{NAFSMF}, {}^{0.8614}\text{DAMF}, {}^{0.5875}\text{AWMF}, {}^{1}\text{ARmF}\}$
SS19/5([0.0222 0.0444 0.0667 0.0889 0.1111 0.1333 0.1556 0.1778 0.2000])	$[d_{ij}]$, $[e_{ij}]$, $[f_{ij}]$	$\{{}^{0}\text{BPDF}, {}^{0.2929}\text{DBAIN}, {}^{0.0988}\text{MDBUTMF}, {}^{0.3117}\text{NAFSMF}, {}^{0.8700}\text{DAMF}, {}^{0.7384}\text{AWMF}, {}^{1}\text{ARmF}\}$
ZCW19$\left(\left[\frac{1}{3}\ \frac{1}{3}\ \frac{1}{3}\right], \left[\frac{1}{3}\ \frac{1}{3}\ \frac{1}{3}\right]\right)$	$[d_{ij}]$, $[e_{ij}]$, $[f_{ij}]$	$\{{}^{0}\text{BPDF}, {}^{0.2930}\text{DBAIN}, {}^{0.0990}\text{MDBUTMF}, {}^{0.3115}\text{NAFSMF}, {}^{0.8701}\text{DAMF}, {}^{0.7384}\text{AWMF}, {}^{1}\text{ARmF}\}$

The ranking orders in Table 11 manifest that G17(I_9), LQP17([1 1 1 1 1 1 1 1 1]), LL18(0.5), ZZ19(0.5, [0.6440 0.6975 0.6918 0.7287 0.7838 0.7766 0.8018]^T), AKO18a, RH18, EC17(0.5), AT18(0.05), SS19/2, SS19/3, SS19/5([0.0222 0.0444 0.0667 0.0889 0.1111 0.1333 0.1556 0.1778 0.2000]), and ZCW19$\left(\left[\frac{1}{3}\ \frac{1}{3}\ \frac{1}{3}\right],\left[\frac{1}{3}\ \frac{1}{3}\ \frac{1}{3}\right]\right)$ produce the same ranking orders. Besides, the ranking orders of P18 and SS19/4 are the same. On the other hands, A19(I_9, [0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4], 1, 1, 1, 1, 2), RH17, AKO18o, and A19/2(I_{729}) generate the unordinary ranking orders compared to the others. Moreover, all the SDM methods herein indicate that ARmF outperforms the other SPN filters and BPDF has the minimum SPN removal performance according to all the SDM methods' ranking orders apart from A19(I_9, [0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4], 1, 1, 1, 1, 2), RH17, AKO18o, and A19/2(I_{729}).

Table 11. Ranking orders produced by SDM methods (in the event of more-importance-attached noise removal performance at low noise densities)

Algorithms	Ranking Orders	Frequency
G17(I_9)	BPDF≺MDBUTMF≺DBAIN≺NAFSMF≺AWMF≺DAMF≺ARmF	12
LQP17([1 1 1 1 1 1 1 1 1])	BPDF≺MDBUTMF≺DBAIN≺NAFSMF≺AWMF≺DAMF≺ARmF	12
LL18(0.5)	BPDF≺MDBUTMF≺DBAIN≺NAFSMF≺AWMF≺DAMF≺ARmF	12
A19(I_9, [0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4], 1, 1, 1, 1, 2)	NAFSMF≺AWMF≺MDBUTMF≺BPDF≺DBAIN≺DAMF≺ARmF	1
ZZ19(0.5, [0.6440 0.6975 0.6918 0.7287 0.7838 0.7766 0.8018]^T)	BPDF≺MDBUTMF≺DBAIN≺NAFSMF≺AWMF≺DAMF≺ARmF	12
RH17	MDBUTMF≺BPDF≺DBAIN≺NAFSMF≺AWMF≺DAMF≺ARmF	1
AKO18a	BPDF≺MDBUTMF≺DBAIN≺NAFSMF≺AWMF≺DAMF≺ARmF	12
AKO18o	AWMF≺NAFSMF≺MDBUTMF≺BPDF≺DBAIN≺DAMF≺ARmF	1
RH18	BPDF≺MDBUTMF≺DBAIN≺NAFSMF≺AWMF≺DAMF≺ARmF	12
EC17(0.5)	BPDF≺MDBUTMF≺DBAIN≺NAFSMF≺AWMF≺DAMF≺ARmF	12
AT18(0.05)	BPDF≺MDBUTMF≺DBAIN≺NAFSMF≺AWMF≺DAMF≺ARmF	12
P18	BPDF≺MDBUTMF≺NAFSMF≺DBAIN≺AWMF≺DAMF≺ARmF	2
A19/2(I_{729})	NAFSMF≺AWMF≺BPDF≺MDBUTMF≺DBAIN≺DAMF≺ARmF	1
SS19/2	BPDF≺MDBUTMF≺DBAIN≺NAFSMF≺AWMF≺DAMF≺ARmF	12
SS19/3	BPDF≺MDBUTMF≺DBAIN≺NAFSMF≺AWMF≺DAMF≺ARmF	12
SS19/4	BPDF≺MDBUTMF≺NAFSMF≺DBAIN≺AWMF≺DAMF≺ARmF	2
SS19/5([0.0222 0.0444 0.0667 0.0889 0.1111 0.1333 0.1556 0.1778 0.2000])	BPDF≺MDBUTMF≺DBAIN≺NAFSMF≺AWMF≺DAMF≺ARmF	12
ZCW19$\left(\left[\frac{1}{3}\ \frac{1}{3}\ \frac{1}{3}\right],\left[\frac{1}{3}\ \frac{1}{3}\ \frac{1}{3}\right]\right)$	BPDF≺MDBUTMF≺DBAIN≺NAFSMF≺AWMF≺DAMF≺ARmF	12

6. Conclusion

The present study configured SDM methods propounded with the concepts of soft sets, fuzzy soft sets, *fpfs*-sets, soft matrices, and fuzzy soft matrices to the *fpfs*-matrices space, faithfully to the original. Thus, this paper completed the configurations of the methods proposed via these concepts in 2017-2019. Then, the configured methods were applied to five test cases. Hereby, the methods producing valid ranking orders for all the test cases were determined. Afterwards, they were applied to a PVA problem to order the well-known filters concerning their noise-removal performance.

This study excluded SDM methods proposed by the superstructures of *fpfs*-sets/matrices. Therefore, in the next studies, researchers can also focus on their configurations to be able to operate methods, constructed via these superstructures, in the appropriate spaces, such as intuitionistic fuzzy parameterized intuitionistic fuzzy soft matrices space (Enginoğlu and Arslan, 2020) and interval-valued intuitionistic fuzzy parameterized interval-valued intuitionistic fuzzy soft sets/matrices space (Aydın and Enginoğlu, 2021; Aydın, 2021). Moreover, it is now possible to compare all the SDM methods operated in *fpfs*-matrices spaces and apply them to different fields, such as machine learning (Memiş *et al*., 2019; Memiş and Enginoğlu, 2019) and archaeology (Enginoğlu *et al*., 2019b). For more details about similar studies, see (Enginoğlu and Memiş, 2018b, c, d; Enginoğlu *et al*., 2018a, b, c, d; Enginoğlu *et al*., 2019c, d; Enginoğlu and Memiş, 2020).

Conflict of Interest

The authors declare no conflict of interest.

Acknowledgement

This work was supported by the Office of Scientific Research Projects Coordination at Çanakkale Onsekiz Mart University, Grant Number: FBA-2020-3259.

References

1. A. O. Atagün, H. Kamacı, O. Oktay, Reduced Soft Matrices and Generalized Products with Applications in Decision Making, Neural and Computing Application 29 (2018) 445-456.
2. B. Mondal, A. K. De, A Study on Some New Operations of Fuzzy Soft Sets and Established forward a Decision-Making Problem, International Journal of Mathematical Archive, 9(4) (2018) 36-44.
3. D. Molodtsov, Soft Set Theory-First Results, Computers Mathematics with Applications 37(4-5) (1999) 19-31.
4. F. Karaca, N. Taş, Decision Making Problem for Life and Non-Life Insurances, Journal of Balıkesir University Institute of Science and Technology 20(1) (2018) 572-588.
5. F. Xiao, A Hybrid Fuzzy Soft Sets Decision Making Method in Medical Diagnosis, IEEE Access 6 (2018) 25300-25312.

6. G. J. Klir, B. Yuan, Fuzzy Sets and Fuzzy Logic Theory and Applications, Prentice-Hall, Upper Saddle River, New Jersey, (1995) 574 pages.
7. H R Sheikh, A C Bovik, Image Information and Visual Quality, IEEE Transactions on Image Processing 15(2) (2006) 430-444.
8. H. Kamacı, A. O. Atagün, A. Sönmezoğlu, Row-products of Soft Matrices with Applications in Multiple-Disjoint Decision Making, Applied Soft Computing 62 (2018) 892-914.
9. J. C. R. Alcantud, M. J. M. Torrecilles, Intertemporal Choice of Fuzzy Soft Sets, Symmetry 10(9) (2018) Article No. 371.
10. J. C. R. Alcantud, T. J. Mathew, Separable Fuzzy Soft Sets and Decision Making with Positive and Negative Attributes, Applied Soft Computing 59 (2017) 586-595.
11. K K V Toh, N A M Isa, Noise Adaptive Fuzzy Switching Median Filter for Salt-and-Pepper Noise Reduction, IEEE Signal Processing Letters 17(3) (2010) 281-284.
12. K S Srinivasan, D Ebenezer, A new fast and efficient decision-based algorithm for removal of high-density impulse noises, IEEE Signal Processing Letters 14(3) (2007) 189-192.
13. L. A. Zadeh, Fuzzy Sets, Information and Control 8(3) (1965) 338-353.
14. L. Wang, K. Qin, Modal-Style Operators on Fuzzy Soft Sets and Their Application to Decision Making, Journal of Intelligent & Fuzzy Systems 36 (2019) 6381–6392.
15. L. Zhang, J. Zhan, Fuzzy Soft β-Covering Based Fuzzy Rough Sets and Corresponding Decision-Making Applications, International Journal of Machine Learning and Cybernetics 10 (2019) 1487-1502.
16. M. Aggarwal, Soft Information Set for Multicriteria Decision Making, International Journal of Intelligent Systems 34 (2019) 3241-3259.
17. M. Riaz, M. R. Hashmi, A. Farooq, Fuzzy Parameterized Fuzzy Soft Metric Spaces, Journal of Mathematical Analysis 9(2) (2018) 25-36.
18. M. Riaz, M. R. Hashmi, Fuzzy Parameterized Fuzzy Soft Compact Spaces with Decision-Making, Journal of Mathematics 50(2) (2018) 131-145.
19. M. Riaz, M. R. Hashmi, Fuzzy Parameterized Fuzzy Soft Topology with Applications, Annals of Fuzzy Mathematics and Informatics 13(5) (2017) 593-613.
20. N. Çağman, F. Çıtak, S. Enginoğlu, FP-Soft Set Theory and Its Applications, Annals of Fuzzy Mathematics and Informatics 2(2) (2011a) 219-226.
21. N. Çağman, F. Çıtak, S. Enginoğlu, Fuzzy Parameterized Fuzzy Soft Set Theory and Its Applications, Turkish Journal of Fuzzy Systems 1(1) (2010) 21-35.
22. N. Çağman, S. Enginoğlu, F. Çıtak, Fuzzy Soft Set Theory and Its Applications, Iranian Journal of Fuzzy Systems 8(3) (2011b) 137-147.
23. N. Çağman, S. Enginoğlu, Fuzzy Soft Matrix Theory and Its Application in Decision Making, Iranian Journal of Fuzzy Systems 9(1) (2012) 109-119.
24. N. Çağman, S. Enginoğlu, Soft Matrix Theory and Its Decision Making, Computers and Mathematics with Applications 59(10) (2010a) 3308-3314.
25. N. Çağman, S. Enginoğlu, Soft Sets Theory and uni-int Decision-Making, European Journal of Operational Research 207(2) (2010b) 848-855.

Published by Firouzabad Institute of Higher Education, Firouzabad, Fars, Iran

26. N. Taş, N. Y. Özgür, P. Demir, An Application of Soft Set and Fuzzy Soft Set Theories to Stock Management, Journal of Natural and Applied Sciences 21(3) (2017) 791-796.
27. P. K. Maji, R. Biswas, A. R. Roy, Fuzzy Soft Sets, The Journal of Fuzzy Mathematics 9(3) (2001) 589-602.
28. R. K. Pal, Application of Average fs-Aggregate Algorithm for Multi-criteria Decision Making in Real Life Problem, Asian Journal of Probability and Statistics 2(2) (2018) 1-11.
29. R. S. Porchelvi, B. Snekaa, On Solving a Multi-Criteria Decision Making Problem Using Fuzzy Soft Sets in Sports, Asia Pacific Journal of Research, 1(87) (2018) 32-38.
30. S Enginoğlu, U Erkan, S Memiş, Pixel Similarity-based Adaptive Riesz Mean Filter for Salt-and-Pepper Noise Removal, Multimedia Tools and Applications 78 (2019a) 35401-35418.
31. S Esakkirajan, T Veerakumar, A N Subramanyam, C H PremChand, Removal of High Density Salt and Pepper Noise Through Modified Decision Based Unsymmetric Trimmed Median Filter, IEEE Signal Processing Letters 18(5) (2012) 287-290.
32. S. Enginoğlu, B. Arslan, Intuitionistic Fuzzy Parameterized Intuitionistic Fuzzy Soft Matrices and Their Application in Decision-Making, Computational and Applied Mathematics 39 (2020) Article No. 325.
33. S. Enginoğlu, M. Ay, N. Çağman, V. Tolun, Classification of The Monolithic Columns Produced in Troad and Mysia Region Ancient Granite Quarries in Northwestern Anatolia via Soft Decision-Making, Bilge International Journal of Science and Technology Research 3(Special Issue) (2019b) 21-34.
34. S. Enginoğlu, N. Çağman, Fuzzy Parameterized Fuzzy Soft Matrices and Their Application in Decision-Making, TWMS Journal of Applied and Engineering Mathematics 10(4) (2020) 1105-1115.
35. S. Enginoğlu, S. Memiş, A Configuration of Some Soft Decision-Making Algorithms via *fpfs*-matrices, Cumhuriyet Science Journal 39(4) (2018a) 871-881.
36. S. Enginoğlu, S. Memiş, A New Approach to the Criteria-Weighted Fuzzy Soft Max-Min Decision-Making Method and Its Application to a Performance-Based Value Assignment Problem, Journal of New Results in Science 9(1) (2020) 19-36.
37. S. Enginoğlu, S. Memiş, A Review on An Application of Fuzzy Soft Set in Multicriteria Decision Making Problem [P. K. Das, R. Borgohain, International Journal of Computer Applications 38(12) (2012) 33-37], Eds: Akgül, M., Yılmaz, İ., İpek, A. International Conference on Mathematical Studies and Applications, pp. 173-178, October 4-6, (2018b) Karaman, Turkey.
38. S. Enginoğlu, S. Memiş, A Review on Some Soft Decision-Making Methods, Eds: Akgül, M., Yılmaz, İ., İpek, A. International Conference on Mathematical Studies and Applications, pp. 437-442, October 4-6, (2018c) Karaman, Turkey.
39. S. Enginoğlu, S. Memiş, B. Arslan, Comment (2) on Soft Set Theory and uni-int Decision Making [European Journal of Operational Research, (2010) 207, 848-855], Journal of New Theory (25) (2018a) 84-102.

40. S. Enginoğlu, S. Memiş, B. Arslan, A Fast and Simple Soft Decision-Making Algorithm: EMA18an, Eds: Akgül, M., Yılmaz, İ., İpek, A. International Conference on Mathematical Studies and Applications, pp. 428-436, October 4-6, (2018b) Karaman, Turkey.
41. S. Enginoğlu, S. Memiş, Comment on "Fuzzy soft sets" [The Journal of Fuzzy Mathematics, 9(3), 2001, 589-602], International Journal of Latest Engineering Research and Applications 3(9) (2018d) 1-9.
42. S. Enginoğlu, S. Memiş, F. Karaaslan, A New Approach to Group Decision-Making Method Based on TOPSIS under Fuzzy Soft Environment, Journal of New Results in Science 8(2) (2019c) 42-52.
43. S. Enginoğlu, S. Memiş, N. Çağman, A Generalisation of Fuzzy Soft Max-Min Decision-Making Method and Its Application to A Performance-Based Value Assignment in Image Denoising, El-Cezerî Journal of Science and Engineering 6(3) (2019d) 466-481.
44. S. Enginoğlu, S. Memiş, T. Öngel, Comment on Soft Set Theory and Uni-Int Decision Making [European Journal of Operational Research, (2010) 207, 848-855], Journal of New Results in Science 7(3) (2018c) 28-43.
45. S. Enginoğlu, S. Memiş, T. Öngel, A Fast and Simple Soft Decision-Making Algorithm: EMO18o, Eds: Akgül, M., Yılmaz, İ., İpek, A. International Conference on Mathematical Studies and Applications, pp. 179-187, October 4-6, (2018d) Karaman, Turkey.
46. S. Enginoğlu, T. Aydın, S. Memiş, B. Arslan, Operability-Oriented Configurations of the Soft Decision-Making Methods Proposed between 2013 and 2016 and Their Comparisons, Journal of New Theory (34) (2021) 82-114.
47. S. Enginoğlu, T. Öngel, Configurations of Several Soft Decision-Making Methods to Operate in Fuzzy Parameterized Fuzzy Soft Matrices Space, Eskişehir Technical University Journal of Science and Technology A-Applied Sciences and Engineering 21(1) (2020) 58-71.
48. S. Eraslan, N. Çağman, A Decision Making Method by Combining TOPSIS and Grey Relation Method under Fuzzy Soft Sets, Sigma Journal of Engineering and Natural Sciences 8(1) (2017) 53-64.
49. S. Memiş, S. Enginoğlu, An Application of Fuzzy Parameterized Fuzzy Soft Matrices in Data Classification, M. Kılıç, K. Özkan, M. Karaboyacı, K. Taşdelen, H. Kandemir, A. Beram, (Ed.), International Conferences on Science and Technology, Natural Science and Technology (2019) 68-77 Prizren, Kosovo.
50. S. Memiş, S. Enginoğlu, U. Erkan, A Data Classification Method in Machine Learning Based on Normalised Hamming Pseudo-Similarity of Fuzzy Parameterized Fuzzy Soft Matrices, Bilge International Journal of Science and Technology Research 3(Special Issue) (2019) 1-8.
51. S. Sandhiya, K. Selvakumari, Application of Fuzzy Soft Sets in Job Requirement Problem, Journal of Computer and Mathematical Sciences 10(1) (2019a) 184-189.
52. S. Sandhiya, K. Selvakumari, Teaching Evaluation Problem on Decision Making Using Fuzzy Soft Sets, A Journal of Composition Theory 12(9) (2019b) 162-167.

53. S. Sharma, S. Singh, On Some Generalized Correlation Coefficients of the Fuzzy Sets and Fuzzy Soft Sets with Application in Cleanliness Ranking of Public Health Centres, Journal of Intelligent and Fuzzy Systems 36 (2019) 3671–3683.
54. T. Aydın, Interval-Valued Intuitionistic Fuzzy Parameterized Interval-Valued Intuitionistic Fuzzy Soft Matrices and Their Application to A Performance-Based Value Assignment, Doctoral Dissertation, Çanakkale Onsekiz Mart University, Çanakkale, Turkey (2021) (In Turkish).
55. T. Aydın, S. Enginoğlu, A Configuration of Five of the Soft Decision-Making Methods via Fuzzy Parameterized Fuzzy Soft Matrices and Their Application to a Performance-Based Value Assignment Problem, M. Kılıç, K. Özkan, M. Karaboyacı, K. Taşdelen, H. Kandemir, A. Beram, (Ed.), International Conferences on Science and Technology, Natural Science and Technology (2019) 56-67 Prizren, Kosovo.
56. T. Aydın, S. Enginoğlu, Configurations of SDM Methods Proposed between 1999 and 2012: A Follow-Up Study, K. Yıldırım, (Ed.) International Conference on Mathematics: "An Istanbul Meeting for World Mathematicians" (2020) 192-211 Istanbul.
57. T. Aydın, S. Enginoğlu, Interval-Valued Intuitionistic Fuzzy Parameterized Interval-Valued Intuitionistic Fuzzy Soft Sets and Their Application in Decision-Making, Journal of Ambient Intelligence and Humanized Computing 12 (2021) 1541-1558.
58. T. J. Neog, B. K. Dutta, On Some New Operations of Fuzzy Soft Sets, International Journal of Mathematical Archive 9(1) (2018) 184-190.
59. U Erkan, L Gökrem, A new method based on pixel density in salt and pepper noise removal, Turkish Journal of Electrical Engineering and Computer Sciences 26 (2018) 162-171.
60. U Erkan, L Gökrem, S Enginoğlu, Different applied median filter in salt and pepper noise, Computers & Electrical Engineering 70 (2018) 789-798.
61. X. Guan, On Central Soft Sets: Definitions and Basic Operations, Journal of Intelligent and Fuzzy Systems 33(2) (2017) 819-827.
62. X. Ma, H. Li, H. Qin, Q. Fei, Z. Gong, The Application of a New Approach of Fuzzy Soft Sets Based Decision-Making in Website Rank and Similarity, 6th International Conference on Systems and Informatics (ICSAI) (2019) 665-670.
63. Y. Liu, X. Liu, Fuzzy Soft Set Multi-Attribute Decision Making Method Based on TOPSIS with Improved Entropy Weight, Advances in Intelligent Systems Research, 147 (2018) 321-332.
64. Y. Zou, W. Chen, D. Wang, Determination of Criteria Weights Based on Soft Sets and Aggregation Operators under Preference Orderings in MCGDM, 2019 International Conference on Information Technology, Electrical and Electronic Engineering (ITEEE 2019) 101–105.
65. Z Tang, Z Yang, K Liu, Z Pei, A new adaptive weighted mean filter for removing high density impulse noise, Proceeding SPIE 10033, Eighth International Conference on Digital Image Processing (ICDIP), Eds: C M Falco, X Jiang, August 2016 1003353 pp. 1-5.

66. Z Wang, A Bovik, H Sheikh, E Simoncelli, Image Quality Assessment: From Error Visibility to Structural Similarity, IEEE Transactions on Image Processing 13(4) (2004) 600-612.
67. Z. Liu, K. Qin, Z. Pei, A Method for Fuzzy Soft Sets in Decision-Making Based on an Ideal Solution, Symmetry 9 (2017) 1-22.

Published by Firouzabad Institute of Higher Education, Firouzabad, Fars, Iran