Cumhuriyet Science Journal
CSJ

e-ISSN: 2587-246X
ISSN: 2587-2680

Cumhuriyet Sci. J., Vol.39-4(2018) 871-881

A Configuration of Some Soft Decision-Making Algorithms via *fpfs*-matrices

Serdar ENGİNOĞLU*, Samet MEMİŞ

Çanakkale Onsekiz Mart University, Faculty of Arts and Sciences, Department of Mathematics, Çanakkale TURKEY

Received: 27.03.2018; Accepted: 04.12.2018 *http://dx.doi.org/10.17776/csj.409915*

Abstract. Since many problems have a large amount of data or uncertainty, the computer mathematics has become compulsory. To deal with such kinds of these problems, the concept of fuzzy parameterized fuzzy soft matrices (*fpfs*-matrices) has been defined by Enginoğlu. In this paper, we first give some of its basic definitions. We then configure some decision-making algorithms constructed by soft sets, fuzzy soft sets, fuzzy parameterized soft sets, *fpfs*-sets, and their matrix representations. We finally discuss later works.

***Keywords*:** Fuzzy sets, Soft sets, Soft decision-making, Soft matrices, *fpfs*-matrices.

Bazı Esnek Karar Verme Algoritmalarının *fpfs*-matrisler Yoluyla Bir Yapılandırması

Özet. Çoğu problem yüksek miktarda veri veya belirsizlik içerdiğinden bilgisayarlı matematik zorunlu hale gelmiştir. Bu tür problemlerle başa çıkabilmek için, bulanık parametreli bulanık esnek matris (*fpfs*-matris) kavramı Enginoğlu tarafından ortaya atıldı. Bu çalışmada ilk olarak, bu kavramın bazı temel tanımlarını veriyoruz. Daha sonra, esnek kümeler, bulanık esnek kümeler, bulanık parametreli esnek kümeler, *fpfs*-kümeler ve onların matris temsilleri yoluyla inşa edilen bazı esnek karar verme algoritmalarını *fpfs*-matrisler yoluyla yapılandırıyoruz. Son olarak, sonraki çalışmalar hakkında bir tartışmaya yer veriyoruz.

***Anahtar Kelimeler*:** Bulanık kümeler, Esnek kümeler, Esnek karar verme, Esnek matrisler, *fpfs*-matrisler

1. INTRODUCTION

Similar to fuzzy sets [1], the concept of soft sets [2] also has been proposed to cope with some problems containing uncertainties and has been applied to many fields. Right after, theoretical and applied studies about soft sets and fuzzy soft sets have been made [3-5]. Then, the soft set operations have been improved and applied to a decision-making problem [6-8]. Afterwards, fuzzy parameterized fuzzy soft sets ($fpfs$-sets) have been produced [9]. Thereafter, to take advantages of matrices, matrix representations of these sets have been constructed [10-12]. Then a wide variety of studies about these concepts have been conducted [13-23].

The presentation of the rest of this paper is organized as follows. In the next section, we give the concept of $fpfs$-matrices [12]. In Section 3, we configure some soft decision-making algorithms via $fpfs$-matrices. The configurations are first mentioned in the second author's master's thesis. In the final section, we present some concluding comments.

* Corresponding author. *Email address: serdarenginoglu@gmail.com*

2. PRELIMINARIES

In this section, we present some of the definitions related to $fpfs$-matrices [12]. Throughout this paper, let E be a parameter set, $F(E)$ be the set of all fuzzy sets over E, and $\mu \in F(E)$. Here, $\mu := \left\{ {}^{\mu(x)}x : x \in E \right\}$.

Definition 1. [9,12] *Let U be a universal set, $\mu \in F(E)$, and α be a function from μ to $F(U)$. Then the graphic of α, denoted by α, defined by*

$$\alpha := \{({}^{\mu(x)}x, \alpha({}^{\mu(x)}x)) : x \in E\}$$

that is called fuzzy parameterized fuzzy soft set ($fpfs$-set) parameterized via E over U (or briefly over U).

In the present paper, the set of all $fpfs$-sets over U is denoted by $FPFS_E(U)$.

Example 1. *Let $E = \{x_1, x_2, x_3, x_4\}$ and $U = \{u_1, u_2, u_3, u_4, u_5\}$. Then*

$$\alpha = \{({}^{0.7}x_1, \{{}^{0.8}u_1, {}^{0.4}u_4\}), ({}^{0.1}x_2, \{{}^{0.2}u_2, {}^{0.4}u_3\}), ({}^{0}x_3, \{{}^{0.6}u_1, {}^{0.7}u_3, {}^{0.5}u_4\}), ({}^{0.9}x_4, \{{}^{0.9}u_3, {}^{0.8}u_5\})\}$$

is an $fpfs$-set over U.

Definition 2. [12] *Let $\alpha \in FPFS_E(U)$. Then $[a_{ij}]$ is called the matrix representation of α (or briefly $fpfs$-matrix of α) and defined by*

$$[a_{ij}] = \begin{bmatrix} a_{01} & a_{02} & a_{03} & \cdots & a_{0n} & \cdots \\ a_{11} & a_{12} & a_{13} & \cdots & a_{1n} & \cdots \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{m1} & a_{m2} & a_{m3} & \cdots & a_{mn} & \cdots \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \end{bmatrix} \quad \textit{for } i = \{0,1,2,\cdots\} \textit{ and } j = \{1,2,\cdots\}$$

such that

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

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

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

Example 2. *Let's consider the $fpfs$-set α provided in Example 1. Then the $fpfs$-matrix of α is as follows:*

$$[a_{ij}] = \begin{bmatrix} 0.7 & 0.1 & 0 & 0.9 \\ 0.8 & 0 & 0.6 & 0 \\ 0 & 0.2 & 0 & 0 \\ 0 & 0.4 & 0.7 & 0.9 \\ 0.4 & 0 & 0.5 & 0 \\ 0 & 0 & 0 & 0.8 \end{bmatrix}$$

Definition 3. [12] *Let* $[a_{ij}],[b_{ik}] \in FPFS_E[U]$ *and* $[c_{ip}] \in FPFS_{E^2}[U]$ *such that* $p = n(j-1)+k$. *For all* i *and* p,

If $c_{ip} = min\{a_{ij}, b_{ik}\}$, *then* $[c_{ip}]$ *is called and-product of* $[a_{ij}]$ *and* $[b_{ik}]$, *denoted by* $[a_{ij}] \wedge [b_{ik}]$.

If $c_{ip} = max\{a_{ij}, b_{ik}\}$, *then* $[c_{ip}]$ *is called or-product of* $[a_{ij}]$ *and* $[b_{ik}]$, *denoted by* $[a_{ij}] \vee [b_{ik}]$.

If $c_{ip} = min\{a_{ij}, 1-b_{ik}\}$, *then* $[c_{ip}]$ *is called andnot-product of* $[a_{ij}]$ *and* $[b_{ik}]$, *denoted by* $[a_{ij}] \overline{\wedge} [b_{ik}]$.

If $c_{ip} = max\{a_{ij}, 1-b_{ik}\}$, *then* $[c_{ip}]$ *is called ornot-product of* $[a_{ij}]$ *and* $[b_{ik}]$, *denoted by* $[a_{ij}] \underline{\vee} [b_{ik}]$.

3. SOME SOFT DECISION-MAKING ALGORITHMS

In this section, we configure some soft decision-making algorithms constructed by soft sets [4,6,14,17], fuzzy soft sets [3,7,13,15,18], fuzzy parameterized soft sets [8,16], $fpfs$-sets [9,21], soft matrices [10,20], and fuzzy soft matrices [11,19] via $fpfs$-matrices [12].

3.1 Algorithm 1 (MBR01) [3]

Step 1. Construct an $fpfs$-matrix $[a_{ij}]$

Step 2. Obtain $[b_{ik}]$ defined by

$$b_{ik} := \sum_{j=1}^{n} a_{0j}\chi(a_{ij}, a_{kj}), \quad i,k \in \{1,2,\dots,m-1\}$$

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 $[c_{i1}]$ defined by

$$c_{i1} := \sum_{k=1}^{m-1} b_{ik}, \quad i \in \{1,2,\dots,m-1\}$$

Step 4. Obtain $[d_{i1}]$, defined by

$$d_{i1} := \sum_{k=1}^{m-1} b_{ki}, \quad i \in \{1,2,\dots,m-1\}$$

Step 5. Obtain $[s_{i1}]$ defined by

$$s_{i1} := c_{i1} - d_{i1}, \quad i \in \{1,2,\dots,m-1\}$$

Step 6. Obtain the set $\{u_k \mid s_{k1} = \max_i (s_{i1})\}$

Preferably, the set $\{^{\mu(u_k)}u_k | u_k \in U\}$ can be attained such that $\mu(u_k) = \frac{s_{k1} + |\min_i s_{i1}|}{\max_i s_{i1} + |\min_i s_{i1}|}$.

3.2 Algorithm 2 (MRB02) [4]

Step 1. Construct an $fpfs$-matrix $[a_{ij}]$

Step 2. Obtain $[s_{i1}]$ defined by

$$s_{i1} := \sum_{j=1}^{n} a_{0j} a_{ij}, \quad i \in \{1,2,\dots,m-1\}$$

Step 3. Obtain the set $\{u_k \mid s_{k1} = \max_i (s_{i1})\}$

Preferably, the set $\{^{\mu(u_k)}u_k | u_k \in U\}$ can be attained such that $\mu(u_k) = \frac{s_{k1}}{\max_i s_{i1}}$.

Note 1. The reduction steps in the original algorithm haven't been considered because they lead to some errors [24,25].

3.3 Algorithm 3 (CE10) [6]

Step 1. Construct two $fpfs$-matrices $[a_{ij}]$ and $[b_{ij}]$

Step 2. Find and-product/or-product $fpfs$-matrix $[c_{ip}]$ of $[a_{ij}]$ and $[b_{ij}]$

Step 3. Obtain $[s_{i1}]$ denoted by max-min(c_{ip}) defined by

$$s_{i1} := \max\{\max_j \min_k (c_{ip}), \max_k \min_j (c_{ip})\}$$

such that $i \in \{1,2,\dots,m-1\}$, $I_a := \{j \mid a_{0j} \neq 0\}$, $I_b := \{k \mid b_{0k} \neq 0\}$, $p = n(j-1)+k$, and

$$\max_j \min_k (c_{ip}) := \begin{cases} \max\limits_{j\in I_a}\left\{\min\limits_{k\in I_b}(c_{0p}c_{ip})\right\}, & I_a \neq \emptyset \text{ and } I_b \neq \emptyset \\ 0, & \text{Otherwise} \end{cases}$$

$$\max_k \min_j (c_{ip}) := \begin{cases} \max\limits_{k\in I_b}\left\{\min\limits_{j\in I_a}(c_{0p}c_{ip})\right\}, & I_a \neq \emptyset \text{ and } I_b \neq \emptyset \\ 0, & \text{Otherwise} \end{cases}$$

Step 4. Obtain the set $\{u_k \mid s_{k1} = \max_i (s_{i1})\}$

Preferably, the set $\{^{s_{i1}}u_i | u_i \in U\}$ or $\{^{\mu(u_k)}u_k | u_k \in U\}$ can be attained such that $\mu(u_k) = \frac{s_{k1}}{\max_i s_{i1}}$.

Note 2. It should be noted that different configurations of this method can be constructed for other products such as andnot-product/ornot-product.

3.4 Algorithm 4 (CCE11) [8]

Step 1. Construct an $fpfs$-matrix $[a_{ij}]$

Step 2. Obtain $[s_{i1}]$ defined by

$$s_{i1} := \frac{1}{\sum_{j=1}^{n} \operatorname{sgn}(a_{0j})} \sum_{j=1}^{n} a_{0j} a_{ij}, \qquad i \in \{1,2,\dots,m-1\}$$

Step 3. The set $\{u_k \mid s_{k1} = \max_i (s_{i1})\}$ is attained

Preferably, the set $\{^{s_{i1}}u_i|u_i \in U\}$ or $\{^{\mu(u_k)}u_k|u_k \in U\}$ can be attained such that $\mu(u_k) = \frac{s_{k1}}{\max_i s_{i1}}$.

3.5 Algorithm 5 (CCE10) [9]

Step 1. Construct an $fpfs$-matrix $[a_{ij}]$

Step 2. Obtain $[s_{i1}]$ defined by

$$s_{i1} := \frac{1}{\text{n}}\sum_{j=1}^{n} a_{0j}a_{ij}, \qquad i \in \{1,2,\dots,m-1\}$$

Step 3. Obtain the set $\{u_k \mid s_{k1} = \max_i (s_{i1})\}$

Preferably, the set $\{^{s_{i1}}u_i|u_i \in U\}$ or $\{^{\mu(u_k)}u_k|u_k \in U\}$ can be attained such that $\mu(u_k) = \frac{s_{k1}}{\max_i s_{i1}}$.

3.6 Algorithm 6 (CE10-2, CE12) [10,11]

Step 1. Construct two $fpfs$-matrices $[a_{ij}]$ and $[b_{ij}]$

Step 2. Find and- product/ or- product $fpfs$-matrix $[c_{ip}]$ of $[a_{ij}]$ and $[b_{ij}]$

Step 3. Obtain $[s_{i1}]$ defined by

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

such that $i \in \{1,2,\dots,m-1\}$ and $I_k := \{p \mid \exists i, c_{0p}c_{ip} \neq 0, (k-1)n < p \leq kn\}$

Step 4. Obtain the set $\{u_k \mid s_{k1} = \max_i (s_{i1})\}$

Preferably, the set $\{^{s_{i1}}u_i|u_i \in U\}$ or $\{^{\mu(u_k)}u_k|u_k \in U\}$ can be attained such that $\mu(u_k) = \frac{s_{k1}}{\max_i s_{i1}}$.

Note 3. It should be noted that different configurations of this method can be constructed for other products such as andnot-product/ornot-product.

3.7 Algorithm 7 (RM11, RM13) [13]

Step 1. Construct three $fpfs$-matrices $[a_{ij}]$, $[b_{ij}]$ and $[c_{ij}]$ such that $\sum_j a_{0j} = \sum_j b_{0j} = \sum_j c_{0j} = 1$

Step 2. Obtain $[A_{ij}]$, $[B_{ij}]$, and $[C_{ij}]$ defined by $A_{ij} := a_{0j}a_{ij}$, $B_{ij} := b_{0j}b_{ij}$, and $C_{ij} := c_{0j}c_{ij}$ such that $i \in \{1,2,\dots,m-1\}$ and $j \in \{1,2,\dots,n\}$

Step 3. Obtain $[d_{ip}]$ defined by $d_{ip} := \min\{A_{ij}, B_{ik}\}$ such that $i \in \{1,2,\dots,m-1\}$, $j,k \in \{1,2,\dots,n\}$, and $p = n(j-1)+k$

Step 4. Obtain $[x_{ij}]$ defined by

$$x_{ij} := \begin{cases} \min\limits_{p\in I_k}\{d_{ip}\}, & I_k \neq \emptyset \\ 0, & I_k = \emptyset \end{cases}$$

such that $i \in \{1,2,\dots,m-1\}$, $j \in \{1,2,\dots,n\}$, and $I_k := \{p \mid \exists i, d_{ip} \neq 0 \ \wedge \ (k-1)n < \text{p} \leq kn\}$

Step 5. Obtain $[e_{ip}]$ defined by $e_{ip} := \min\{x_{ij}, C_{it}\}$ such that $i \in \{1,2,\dots,m-1\}$, $j,t \in \{1,2,\dots,n\}$, and $p = n(j-1)+t$

Step 6. Obtain $[y_{ij}]$ defined by

$$y_{ij} := \begin{cases} \min\limits_{p \in I_t}\{e_{ip}\}, & I_t \neq \emptyset \\ 0, & I_t = \emptyset \end{cases}$$

such that $i \in \{1,2,\dots,m-1\}$, $j \in \{1,2,\dots,n\}$, and $I_t := \{p \mid \exists i, e_{ip} \neq 0 \ \land \ (t-1)n < p \leq tn\}$

Step 7. Obtain $[s_{i1}]$ defined by

$$s_{i1} := \max_j \{y_{ij}\}, \quad i \in \{1,2,\dots,m-1\} \text{ and } j \in \{1,2,\dots,n\}$$

Step 8. Obtain the set $\{u_k \mid s_{k1} = \max\limits_i (s_{i1})\}$

Preferably, the set $\{^{s_{i1}}u_i | u_i \in U\}$ or $\{^{\mu(u_k)}u_k | u_k \in U\}$ can be attained such that $\mu(u_k) = \frac{s_{k1}}{\max_i s_{i1}}$.

3.8 Algorithm 8 (DB12) [15]

Step 1. Construct $fpfs$-matrices $[a_{ij}]^{(1)}$, $[a_{ij}]^{(2)}$, ..., $[a_{ij}]^{(t)}$ such that $\sum_j a_{0j}^{(1)} \leq 1$, $\sum_j a_{0j}^{(2)} \leq 1$, ..., $\sum_j a_{0j}^{(t)} \leq 1$

Step 2. Obtain $[b_{ij}]$ defined by

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

Step 3. Algorithm 1 is applied to the matrix $[b_{ij}]$

3.9 Algorithm 9 (CD12) [16]

Step 1. Construct $fpfs$-matrices $[a_{ij}]^{(1)}$, $[a_{ij}]^{(2)}$, ..., $[a_{ij}]^{(t)}$

Step 2. Obtain $[b_{ij}]$ defined by

$$b_{0j} := \left(\frac{1}{t}\sum_{k=1}^{t}\left({a_{0j}}^{(\mathrm{k})}\right)^p\right)^{\frac{1}{p}} \text{ and } b_{ij} := \min\{{a_{ij}}^{(1)}, {a_{ij}}^{(2)}, \dots, {a_{ij}}^{(\mathrm{t})}\}$$

such that $i \in \{1,2,\dots,m-1\}$ and $j \in \{1,2,\dots,n\}$

Step 3. Algorithm 5 is applied to the matrix $[b_{ij}]$

3.10 Algorithm 10 (CD12-2) [16]

Step 1. Construct $fpfs$-matrices $[a_{ij}]^{(1)}$, $[a_{ij}]^{(2)}$, ..., $[a_{ij}]^{(t)}$

Step 2. Obtain $[b_{ij}]$ defined by

$$b_{0j} := \left(\frac{1}{t}\sum_{k=1}^{t}\left({a_{0j}}^{(\mathrm{k})}\right)^p\right)^{\frac{1}{p}} \text{ and } b_{ij} := \max\{{a_{ij}}^{(1)}, {a_{ij}}^{(2)}, \dots, {a_{ij}}^{(\mathrm{t})}\}$$

such that $i \in \{1,2,\dots,m-1\}$ and $j \in \{1,2,\dots,n\}$

Step 4. Algorithm 5 is applied to the matrix $[b_{ij}]$

3.11 Algorithm 11 (E15) [17]

Step 1. Construct $fpfs$-matrices $[a_{ij}]^{(1)}$, $[a_{ij}]^{(2)}$, ..., $[a_{ij}]^{(t)}$

Step 2. Obtain $[b_{ip}]$ defined by $b_{ip} \coloneqq a_{ij}^{(k)}$ such that $i \in \{0,1,2,\dots,m-1\}$, $j \in \{1,2,\dots,n\}$, $k \in \{1,2,\dots,t\}$, $p \in \{1,2,\dots,nt\}$, and $p = n(k-1)+j$

Step 3. Obtain $[c_{ij}]$ defined by

$$c_{ij} \coloneqq \sum_{k=1}^{t} b_{0p} b_{ip}, \quad i \in \{1,2,\dots,m-1\}, j \in \{1,2,\dots,n\}, \text{and } p \in \{1,2,\dots,nt\}$$

such that $p = n(k-1)+j$

Step 4. Obtain $[s_{i1}]$ defined by

$$s_{i1} \coloneqq \sum_{j=1}^{n} c_{ij}, \quad i \in \{1,2,\dots,m-1\}$$

Step 5. Obtain the set $\{u_k \mid s_{k1} = \max_i (s_{i1})\}$

Preferably, the set $\{^{\mu(u_k)}u_k | u_k \in U\}$ can be attained such that $\mu(u_k) = \frac{s_{k1}}{\max_i s_{i1}}$.

3.12 Algorithm 12 (EK15) [18]

Step 1. Construct $fpfs$-matrices $[a_{ij}]^{(1)}$, $[a_{ij}]^{(2)}$, ..., $[a_{ij}]^{(t)}$

Step 2. Obtain $[b_{ij}]$ defined by $b_{ij} \coloneqq a_{0j}^{(i)}$ such that $i \in \{1,2,\dots,t\}$ and $j \in \{1,2,\dots,n\}$

Step 3. Obtain $[c_{ij}]$ defined by

$$c_{ij} \coloneqq \frac{b_{ij}}{\sqrt{\sum_{k=1}^{t} {b_{kj}}^2}}$$

such that $i \in \{1,2,\dots,m-1\}$, $j \in \{1,2,\dots,n\}$, and $k \in \{1,2,\dots,t\}$

Step 4. Obtain $[d_{i1}]$ defined by

$$d_{i1} \coloneqq \frac{1}{t} \sum_{j=1}^{t} c_{ji}$$

Step 5. Obtain $[e_{i1}]$ defined by

$$e_{i1} \coloneqq \frac{d_{i1}}{\sum_{k=1}^{n} d_{k1}}$$

Step 6. Obtain $[f_{ij}]$ defined by

$$f_{ij} \coloneqq \frac{1}{t} \sum_{k=1}^{t} {a_{ij}}^{(k)}, \quad i \in \{1,2,\dots,m-1\} \text{ and } j \in \{1,2,\dots,n\}$$

Step 7. Obtain $[g_{ij}]$ defined by $g_{ij} \coloneqq e_{j1} f_{ij}$ such that $i \in \{1,2,\dots,m-1\}$ and $j \in \{1,2,\dots,n\}$

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

$$g_{1j}^{+} := \begin{cases} \max\limits_{i}\{g_{ij}\}, & e_{j1} \neq 0 \\ \min\limits_{i}\{g_{ij}\}, & e_{j1} = 0 \end{cases} \quad \text{and} \quad g_{1j}^{-} := \begin{cases} \min\limits_{i}\{g_{ij}\}, & e_{j1} \neq 0 \\ \max\limits_{i}\{g_{ij}\}, & e_{j1} = 0 \end{cases}$$

such that $i \in \{1,2,\dots,m-1\}$ and $j \in \{1,2,\dots,n\}$

Step 9. Obtain $[s_{i1}^{+}]$ and $[s_{i1}^{-}]$ defined by

$$s_{i1}^{+} := \sqrt{\sum_{j=1}^{n} (g_{ij} - g_{1j}^{+})^2} \quad \text{and} \quad s_{i1}^{-} := \sqrt{\sum_{j=1}^{n} (g_{ij} - g_{1j}^{-})^2}$$

such that $i \in \{1,2,\dots,m-1\}$ and $j \in \{1,2,\dots,n\}$

Step 10. Obtain $[s_{i1}]$ defined by

$$s_{i1} := \frac{s_{i1}^{-}}{s_{i1}^{+} + s_{i1}^{-}}, \quad i \in \{1,2,\dots,m-1\}$$

Step 11. Obtain the set $\{u_k \mid s_{k1} = \max\limits_{i} (s_{i1})\}$

Preferably, the set $\{^{\mu(u_k)}u_k | u_k \in U\}$ can be attained such that $\mu(u_k) = \frac{s_{k1}}{\max\limits_{i} s_{i1}}$.

3.13 Algorithm 13 (KKT13) [19]

Step 1. Construct an $fpfs$-matrix $[a_{ij}]_{(n+1)\times n}$

Step 2. Obtain $[b_{ij}]$ defined by

$$b_{i1} := \frac{1}{n}\sum_{j=1}^{n} a_{ij}, \quad i \in \{1,2,\dots,n\}$$

Step 3. Obtain $[s_{i1}]$ defined by

$$s_{i1} := a_{0i} b_{i1}, \quad i \in \{1,2,\dots,n\}$$

Step 4. Obtain the set $\{u_k \mid s_{k1} = \max\limits_{i} (s_{i1})\}$

Preferably, the set $\{^{\mu(u_k)}u_k | u_k \in U\}$ can be attained such that $\mu(u_k) = \frac{s_{k1}}{\max\limits_{i} s_{i1}}$.

3.14 Algorithm 14 (VR13) [20]

Step 1. Construct two $fpfs$-matrices $[a_{ij}]$ and $[b_{ij}]$

Step 2. Find and- product/ or- product $fpfs$-matrix $[c_{ip}]$ of $[a_{ij}]$ and $[b_{ij}]$

Step 3. Obtain $[d_{ip}]$ defined by $d_{ip} := \frac{a_{ij}+b_{ik}}{2}$ such that $i \in \{1,2,\dots,m-1\}$, $j,k \in \{1,2,\dots,n\}$, and $p = n(j-1)+k$

Step 4. Obtain $[e_{ip}]$ defined by $e_{ip} := \min\{c_{ip}, d_{ip}\}$

Step 5. Obtain $[f_{ip}]$ defined by

$$f_{ip} := \begin{cases} 1, & e_{ip} \geq \max\limits_{k}\{e_{kp}\} \\ 0, & e_{ip} < \max\limits_{k}\{e_{kp}\} \end{cases} \quad \text{and} \quad f_{0p} := e_{0p}$$

such that $i, k \in \{1,2, \dots, m-1\}$ and $p \in \{1,2, \dots, n^2\}$

Step 6. Obtain $[s_{i1}]$ defined by

$$s_{i1} := \sum_{p=1}^{n^2} f_{0p} f_{ip}$$

Step 7. Obtain the set $\{u_k \mid s_{k1} = \max_i (s_{i1})\}$

Preferably, the set $\{^{\mu(u_k)}u_k | u_k \in U\}$ can be attained such that $\mu(u_k) = \frac{s_{k1}}{\max_i s_{i1}}$.

3.15 Algorithm 15 (ZZ16) [21]

Step 1. Construct two $fpfs$-matrices $[a_{ij}]$ and $[b_{ij}]$

Step 2. Obtain $[c_{ij}]$ defined by

$$c_{0j} := \frac{a_{0j} b_{0j}}{2 - \left(a_{0j} + b_{0j} - a_{0j} b_{0j}\right)} \quad \text{and} \quad c_{ij} := \min\{a_{ij}, b_{ij}\}$$

such that $i \in \{1,2, \dots, m-1\}$ and $j \in \{1,2, \dots, n\}$

Step 3. Obtain $[d_{ij}]$ defined by

$$d_{0j} := c_{0j} \ \text{and} \ d_{ij} := \begin{cases} c_{ij}, & c_{ij} \geq r \\ 0, & c_{ij} < r \end{cases}$$

such that $r \in [0,1]$, $i \in \{1,2, \dots, m-1\}$, and $j \in \{1,2, \dots, n\}$

Step 4. Algorithm 5 is applied to the matrix $[d_{ij}]$

3.16 Algorithm 16 (ZZ16-2) [21]

Step 1. Construct $fpfs$-matrices $[a_{ij}]$ and $[b_{ij}]$

Step 2. Obtain $[c_{ij}]$ defined by

$$c_{0j} := \frac{a_{0j} + b_{0j}}{1 + a_{0j} b_{0j}} \quad \text{and} \quad c_{ij} := \max\{a_{ij}, b_{ij}\}$$

such that $i \in \{1,2, \dots, m-1\}$ and $j \in \{1,2, \dots, n\}$

Step 3. Obtain $[d_{ij}]$ defined by

$$d_{ij} := \begin{cases} c_{ij}, & c_{ij} \geq r \\ 0, & c_{ij} < r \end{cases} \text{and } d_{0j} := c_{0j}$$

such that $r \in [0,1]$, $i \in \{1,2, \dots, m-1\}$, and $j \in \{1,2, \dots, n\}$

Step 4. Algorithm 5 is applied to the matrix $[d_{ij}]$

4. CONCLUSION

In this study, we have configured 18 soft decision-making algorithms faithfully to the original via $fpfs$-matrices. Of course, these algorithms are open to different generalizations needed to deal with the heavier types of uncertainty. Moreover, it is worth doing the study on their simplifications and their different configurations can be constructed. For example, Algorithm 3 and 6 can be configurated via andnot-product/ornot-product. On the other hand, the reliability and validity tests of all these algorithms are still an open problem.

In the literature, there are more than 200 soft decision-making algorithms; however, their names are not available. This situation has caused some difficulties. To overcome this problem, we have suggested a notation for the algorithms, by using the first letters of their author's names and the last two digits of their publication year. For example, the algorithm in [3] has been denoted by MBR01.

Acknowledgements

This work was supported by the Office of Scientific Research Projects Coordination at Çanakkale Onsekiz Mart University, Grant number: FYL-2015-684.

REFERENCES

[1] Zadeh L. A., Fuzzy Sets, Information and Control, 8 (1965) 338-353.
[2] Molodtsov D., Soft Set Theory-First Results, Computers and Mathematics with Applications, 37 (1999) 19-31.
[3] Maji P. K., Biswas R. and Roy A.R., Fuzzy Soft Sets, The Journal of Fuzzy Mathematics, 9-3 (2001) 589-602.
[4] Maji P.K., Roy A.R. and Biswas R., An Application of Soft Sets in A Decision Making Problem, Computers and Mathematics with Applications, 44 (2002) 1077-1083.
[5] Maji P.K., Biswas R. and Roy A.R., Soft Set Theory, Computers and Mathematics with Applications, 45 (2003) 555-562.
[6] Çağman N. and Enginoğlu S., Soft Set Theory and Uni-Int Decision Making, European Journal of Operational Research, 207 (2010) 848-855.
[7] Çağman N., Enginoğlu S. and Çıtak F., Fuzzy Soft Set Theory and Its Applications, Iranian Journal of Fuzzy Systems, 8-3 (2011) 137-147.
[8] Çağman N., Çıtak F. and Enginoğlu S., FP-Soft Set Theory and Its Applications, Annals of Fuzzy Mathematics Information, 2-2 (2011) 219-226.
[9] Çağman N., Çıtak F. and Enginoğlu S., Fuzzy Parameterized Fuzzy Soft Set Theory and Its Applications, Turkish Journal of Fuzzy Systems, 1-1 (2010) 21-35.
[10] Çağman N. and Enginoğlu S., Soft Matrix Theory and Its Decision Making, Computers and Mathematics with Applications, 59 (2010) 3308-3314.
[11] Çağman N. and Enginoğlu S., Fuzzy Soft Matrix Theory and Its Application in Decision Making, Iranian Journal of Fuzzy Systems, 9-1 (2012) 109-119.
[12] Enginoğlu S., Soft Matrices, PhD Diss., Gaziosmanpaşa University, Turkey, 2012.
[13] Razak S. and Mohamad D., A Decision Making Method using Fuzzy Soft Sets, Malaysian Journal of Fundamental and Applied Science, 9-2 (2013) 99-104.
[14] Razak S. and Mohamad D., A Soft Set Based Group Decision Making Method with Criteria Weight, World Academy of Science, Engineering and Technology, 5-10 (2011) 1641-1646.
[15] Das P. K. and Borgohain R., An Application of Fuzzy Soft Set in Multicriteria Decision Making Problem, International Journal of Computer Applications, 38-12 (2012) 33-37.
[16] Çağman N. and Deli İ., Means of FP-Soft Sets and Their Applications, Hacettepe Journal of Mathematics and Statistics, 41-5 (2012) 615-625.

[17] Eraslan S., A Decision Making Method via TOPSIS on Soft Sets, Journal of New Results in Science, 8 (2015) 57-71.
[18] Eraslan S. and Karaaslan F., A Group Decision Making Method Based on TOPSIS under Fuzzy Soft Environment, Journal of New Theory, 3 (2015) 30-40.
[19] Khan N., Khan F. H. and Thakur G. S., Weighted Fuzzy Soft Matrix Theory and Its Decision Making, International Journal of Advances in Computer Science and Technology, 2-10 (2013) 214-218.
[20] Vijayabalaji S. and Ramesh A., A New Decision Making Theory in Soft Matrices, International Journal of Pure and Applied Mathematics, 86-6, (2013) 927-939.
[21] Zhu K. and Zhan J., Fuzzy Parameterized Fuzzy Soft Sets and Decision Making, International Journal of Machine Learning and Cybernetics, 7 (2016) 1207-1212.
[22] Çağman N. and Deli İ., Products of FP-soft Sets and Their Applications, Hacettepe Journal of Mathematics and Statistics, 41-3 (2012) 365-374.
[23] Çağman N. and Deli İ., Relations on FP-soft Sets Applied to Decision Making Problems, Journal of New Theory 3 (2015) 98–107.
[24] Chen D., Tsang E.C.C., Yeung D. S. and Wang X., The Parameterization Reduction of Soft Sets and Its Applications, Computers and Mathematics with Applications, 49 (2005) 757-763.
[25] Kong Z., Gao L., Wang L. and Li S., The Normal Parameter Reduction of Soft Sets and Its Algorithm, Computers and Mathematics with Applications, 56 (2008) 3029-3037.