
Serdar Enginoğlu* <serdarenginoglu@gmail.com>
Samet Memiş <sametmemis@gmail.com>
Burak Arslan <tburakarslan@gmail.com>

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

Abstract — The uni-int decision-making method, which selects a set of optimum elements from the alternatives, was defined by Çağman and Enginoğlu [Soft set theory and uni-int decision making, European Journal of Operational Research 207 (2010) 848-855] via soft sets and their soft products. Lately, this method constructed by and-product/or-product has been configured by Enginoğlu and Memiş [A configuration of some soft decision-making algorithms via fpfs-matrices, Cumhuriyet Science Journal 39 (4) (2018) In Press] via fuzzy parameterized fuzzy soft matrices (fpfs-matrices), faithfully to the original, because a more general form is needed for the method in the event that the parameters or objects have uncertainties. In this study, we configure the method via fpfs-matrices and andnot-product/ornot-product, faithfully to the original. However, in the case that a large amount of data is processed, the method still has a disadvantage regarding time and complexity. To deal with this problem and to be able to use this configured method effectively denoted by CE10n, we suggest two new algorithms in this paper, i.e. EMA18an and EMA18on, and prove that CE10n constructed by andnot-product (CE10an) and constructed by ornot-product (CE10on) are special cases of EMA18an and EMA18on, respectively, if first rows of the fpfs-matrices are binary. We then compare the running times of these algorithms. The results show that EMA18an and EMA18on outperform CE10an and CE10on, respectively. Particularly in problems containing a large amount of parameters, EMA18an and EMA18on offer up to 99.9966% and 99.9964% of time advantage, respectively. Latterly, we apply EMA18on to a performance-based value assignment to the methods used in the noise removal, so that we can order them in terms of performance. Finally, we discuss the need for further research.

Keywords — Fuzzy sets, Soft sets, Soft decision-making, Soft matrices, fpfs-matrices

*Corresponding Author.
1 Introduction

The concept of soft sets was produced by Molodtsov [1] to deal with uncertainties, and so far many theoretical and applied studies from algebra to decision-making problems [2–24] have been conducted on this concept.

Recently, some decision-making algorithms constructed by soft sets [3, 5, 25, 26], fuzzy soft sets [2, 8, 27–29], fuzzy parameterized soft sets [9, 30], fuzzy parameterized fuzzy soft sets (fpfs-sets) [7, 31], soft matrices [5, 32] and fuzzy soft matrices [10, 33] have been configured [34] via fuzzy parameterized fuzzy soft matrices (fpfs-matrices) [11], faithfully to the original, because a more general form is needed for the method in the event that the parameters or objects have uncertainties.

One of the configured methods above-mentioned is CE10 [5, 34] constructed by and-product (CE10a) or constructed by or-product (CE10o). Since the authors point to a configuration of these methods by using a different product such as andnot-product and ornot-product, in this study, we configure the uni-int decision-making method constructed by andnot-product/ornot-product via fpfs-matrices, faithfully to the original. However, in the case that a large amount of data is processed, this configured method denoted by CE10n has a disadvantage regarding time and complexity. It can be overcome this problem via simplification of the algorithms but in the event that first rows of the fpfs-matrices are binary, though there exist simplified versions of CE10n constructed by andnot-product (CE10an) and constructed by ornot-product (CE10on), no exist in the other cases. Therefore, in this study, we aim to develop two algorithms which have the ability of CE10an and CE10on and are also faster than them.

In Section 2 of the present study, we introduce the concept of fpfs-matrices. In Section 3, we configure the uni-int decision-making method constructed by andnot-product/ornot-product via fpfs-matrices. In Section 4, we suggest two new algorithms in this paper, i.e. EMA18an and EMA18on, and prove that CE10an and CE10on are special cases of EMA18an and EMA18on, respectively, if first rows of the fpfs-matrices are binary. A part of this section has been presented in [35]. In Section 5, we compare the running times of these algorithms. In Section 6, we apply EMA18on to the decision-making problem in image denoising. Finally, we discuss the need for further research.

2 Preliminary

In this section, we present the definition of fpfs-sets and fpfs-matrices. 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, $\mu := \{\mu(x) : x \in E\}$.

**Definition 2.1.** [7, 11] Let $U$ be a universal set, $\mu \in F(E)$, and $\alpha$ be a function from $\mu$ to $F(U)$. Then the graphic of $\alpha$, denoted by $\alpha$, defined by

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

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 2.2. Let $E = \{x_1, x_2, x_3, x_4\}$ and $U = \{u_1, u_2, u_3, u_4, u_5\}$. Then
\[
\alpha = \{(x_1, \{0.3, u_1, 0.7, u_3\}), (x_2, \{0.8, u_2, 0.2, u_3, 0.9, u_5\}), (x_3, \{0.3, u_2, 0.5, u_4, 0.2, u_5\}), (x_4, \{0.9, u_2, 0.9, u_4\})\}
\]
is a $fpfs$-set over $U$.

Definition 2.3. [11] Let $\alpha \in FPFS_E(U)$. Then $[a_{ij}]$ is called the matrix representation of $\alpha$ (or briefly $fpfs$-matrix of $\alpha$) and defined by
\[
[a_{ij}] = \begin{bmatrix}
a_{01} & a_{02} & a_{03} & \cdots & a_{0n} & \cdots \\
a_{11} & a_{12} & a_{13} & \cdots & a_{1n} & \cdots \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
a_{m1} & a_{m2} & a_{m3} & \cdots & a_{mn} & \cdots \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
\end{bmatrix}
\]
for $i = \{0, 1, 2, \cdots\}$ and $j = \{1, 2, \cdots\}$ such that
\[
a_{ij} := \begin{cases} 
\mu(x_j), & i = 0 \\
\alpha(\mu(x_j)^E_i), & i \neq 0
\end{cases}
\]
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 2.4. Let’s consider the $fpfs$-set $\alpha$ provided in Example 2.2. Then the $fpfs$-matrix of $\alpha$ is as follows:
\[
[a_{ij}] = \begin{bmatrix}
1 & 0.8 & 0.3 & 0 \\
0.3 & 0.2 & 0 & 0 \\
0 & 0 & 0.5 & 1 \\
0.7 & 0.2 & 0 & 0 \\
0 & 0 & 0.7 & 0.9 \\
0 & 0.9 & 0.2 & 0 \\
\end{bmatrix}
\]

Definition 2.5. [11] 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 $[b_{ik}]$ and is 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 $[b_{ik}]$ and is 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 $[b_{ik}]$ and is denoted by $[a_{ij}] \wedge [b_{ik}]$.

If $c_{ip} = \max\{a_{ij}, 1 - b_{ik}\}$, then $[c_{ip}]$ is called ornot-product of $[a_{ij}]$ and $[b_{ik}]$ and is denoted by $[a_{ij}] \vee [b_{ik}]$. 
3 A Configuration of the uni-int Decision-Making Method

In this section, we configure the uni-int decision-making method [5] constructed by andnot-product/ornot-product via fpfs-matrices.

Algorithm Steps

**Step 1.** Construct two fpfs-matrices \([a_{ij}]\) and \([b_{ik}]\)

**Step 2.** Find andnot-product/ornot-product fpfs-matrix \([c_{ip}]\) of \([a_{ij}]\) and \([b_{ik}]\)

**Step 3.** Find andnot-product/ornot-product fpfs-matrix \([d_{it}]\) of \([b_{ik}]\) and \([a_{ij}]\)

**Step 4.** Obtain \([s_{i1}]\) denoted by max-min\((c_{ip}, d_{it})\) defined by

\[
s_{i1} := \max\{\max_j\min_k(c_{ip}), \max_k\min_j(d_{it})\}
\]
such that \(i \in \{1, 2, \ldots, m - 1\}, I_a := \{j \mid a_{0j} \neq 0\}, I_b := \{k \mid b_{0k} \neq 0\}, I_a^* := \{j \mid 1 - a_{0j} \neq 0\}, I_b^* := \{k \mid 1 - b_{0k} \neq 0\}, p = n(j - 1) + k, t = n(k - 1) + j,\) and

\[
\max_j\min_k(c_{ip}) := \begin{cases} \max_{j \in I_a} \left\{ \min_{k \in I_b^*} c_{ip} \right\}, & I_a \neq \emptyset \text{ and } I_b^* \neq \emptyset \\ 0, & \text{otherwise} \end{cases}
\]

\[
\max_k\min_j(d_{it}) := \begin{cases} \max_{k \in I_b} \left\{ \min_{j \in I_a^*} d_{it} \right\}, & I_a^* \neq \emptyset \text{ and } I_b \neq \emptyset \\ 0, & \text{otherwise} \end{cases}
\]

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

Preferably, the set \(\{s_{i1}u_i \mid u_i \in U\}\) or \(\{\frac{s_{k1}}{\max_k s_{k1}}u_k \mid u_k \in U\}\) can be attained.

4 The Soft Decision-Making Methods: EMA18an and EMA18on

In this section, firstly, we present a fast and simple algorithm denoted by EMA18an [35].

EMA18an’s Algorithm Steps

**Step 1.** Construct two fpfs-matrices \([a_{ij}]\) and \([b_{ik}]\)

**Step 2.** Obtain \([s_{i1}]\) denoted by max-min\((a_{ij}, b_{ik})\) defined by

\[
s_{i1} := \max\{\max_j\min_k(a_{ij}, b_{ik}), \max_k\min_j(b_{ik}, a_{ij})\}
\]
such that \(i \in \{1, 2, \ldots, m - 1\}, I_a := \{j \mid a_{0j} \neq 0\}, I_b := \{k \mid b_{0k} \neq 0\}, I_a^* := \{j \mid 1 - a_{0j} \neq 0\}, I_b^* := \{k \mid 1 - b_{0k} \neq 0\},\) and

\[
\max_j\min_k(a_{ij}, b_{ik}) := \begin{cases} \min_{j \in I_a} \left\{ \max_{k \in I_b^*} a_{ij}, \min_{k \in I_b^*} (1 - b_{0k})(1 - b_{ik}) \right\}, & I_a \neq \emptyset \text{ and } I_b^* \neq \emptyset \\ 0, & \text{otherwise} \end{cases}
\]

\[
\max_k\min_j(b_{ik}, a_{ij}) := \begin{cases} \min_{k \in I_b} \left\{ \max_{j \in I_a^*} b_{ik}, \min_{j \in I_a^*} (1 - a_{0j})(1 - a_{ij}) \right\}, & I_a^* \neq \emptyset \text{ and } I_b \neq \emptyset \\ 0, & \text{otherwise} \end{cases}
\]
Step 3. Obtain the set \( \{ u_k \mid s_k = \max_i s_{i1} \} \)

Preferably, the set \( \{ s_{i1} u_i \mid u_i \in U \} \) or \( \{ \max_i s_{i1} u_k \mid u_k \in U \} \) can be attained.

Secondly, we propose a fast and simple algorithm denoted by EMA18on.

**EMA18on’s Algorithm Steps**

Step 1. Construct two fpfs-matrices \([a_{ij}]\) and \([b_{ik}]\)

Step 2. Obtain \([s_{i1}]\) denoted by \( \max - \min(a_{ij}, b_{ik}) \) defined by

\[
s_{i1} := \max \{ \max_j \min_k (a_{ij}, b_{ik}), \max_k \min_j (b_{ik}, a_{ij}) \}
\]

such that \( i \in \{1, 2, \ldots, m - 1\}, I_a := \{ j \mid a_{0j} \neq 0 \}, I_b := \{ k \mid b_{0k} \neq 0 \}, I_a^* := \{ j \mid 1 - a_{0j} \neq 0 \}, I_b^* := \{ k \mid 1 - b_{0k} \neq 0 \}, \) and

\[
\max_j \min_k (a_{ij}, b_{ik}) := \begin{cases} 
\max \left\{ \max_{j \in I_a} \{ a_{0j} a_{ij} \}, \min_{k \in I_b^*} \{ 1 - b_{0k} (1 - b_{ik}) \} \right\}, & I_a \neq \emptyset \text{ and } I_b^* \neq \emptyset \\
0, & \text{otherwise}
\end{cases}
\]

\[
\max_k \min_j (b_{ik}, a_{ij}) := \begin{cases} 
\max \left\{ \max_{k \in I_b^*} \{ b_{0k} b_{ik} \}, \min_{j \in I_a^*} \{ 1 - a_{0j} (1 - a_{ij}) \} \right\}, & I_a^* \neq \emptyset \text{ and } I_b \neq \emptyset \\
0, & \text{otherwise}
\end{cases}
\]

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

Preferably, the set \( \{ s_{i1} u_i \mid u_i \in U \} \) or \( \{ \max_i s_{i1} u_k \mid u_k \in U \} \) can be attained.

**Theorem 4.1.** [35] CE10an is a special case of EMA18an provided that first rows of the fpfs-matrices are binary.

**Proof.** Suppose that first rows of the fpfs-matrices are binary. The functions \( s_{i1} \) provided in CE10an and EMA18an are equal in the event that \( I_a = \emptyset \) or \( I_b^* = \emptyset \). Assume that \( I_a \neq \emptyset \) and \( I_b^* \neq \emptyset \). Since \( a_{0j} = 1 \) and \( b_{0k} = 0 \), for all \( j \in I_a := \{ a_1, a_2, \ldots, a_s \} \) and \( k \in I_b^* := \{ b_1, b_2, \ldots, b_t \} \),

\[
\max_j \min_k (c_{ip}) = \max_j \left\{ \min_{k \in I_b^*} \{ c_{ip} c_{ip} \} \right\}
\]

\[
= \max_j \left\{ \min_{k \in I_b^*} \{ \min\{ a_{0j}, 1 - b_{0k} \}, \min\{ a_{ij}, 1 - b_{ik} \} \} \right\}
\]

\[
= \max_j \left\{ \min_{k \in I_b^*} \{ \min\{ a_{ij}, 1 - b_{ik} \} \} \right\}
\]

\[
= \max \{ \min\{ a_{ia_1}, 1 - b_{ib_1} \}, \min\{ a_{ia_2}, 1 - b_{ib_2} \}, \ldots, \min\{ a_{ia_1}, 1 - b_{ib_1} \} \}
\]

\[
\min\{ a_{ia_1}, 1 - b_{ib_1} \}, \min\{ a_{ia_1}, 1 - b_{ib_2} \}, \ldots, \min\{ a_{ia_2}, 1 - b_{ib_1} \}, \ldots, \min\{ a_{ia_2}, 1 - b_{ib_1} \}, \ldots, \min\{ a_{ia_2}, 1 - b_{ib_2} \}, \ldots, \min\{ a_{ia_s}, 1 - b_{ib_t} \} \}.
\]
\[ \begin{align*}
&= \max \left\{ \min \{ a_{i_1}, \min \{1 - b_{ib_1}, 1 - b_{ib_2}, \ldots, 1 - b_{ib_t} \} \},
\quad \min \{ a_{i_2}, \min \{1 - b_{ib_1}, 1 - b_{ib_2}, \ldots, 1 - b_{ib_t} \} \}, \ldots, \\
&\min \{ a_{i_s}, \min \{1 - b_{ib_1}, 1 - b_{ib_2}, \ldots, 1 - b_{ib_t} \} \} \right\} \\
&= \min \left\{ \max \{ a_{i_1}, a_{i_2}, \ldots, a_{i_s} \}, \min \{1 - b_{ib_1}, 1 - b_{ib_2}, \ldots, 1 - b_{ib_t} \} \right\} \\
&= \min \left\{ \max \{ a_{ij} \}, \min \{1 - b_{ik} \} \right\} \\
&= \min \left\{ \max \{ a_{ij} \}, \min \{1 - b_{ik} \} \right\} \\
&= \max_j \min_k (a_{ij}, b_{ik}) \\
\end{align*} \]

In a similar way, \( \max_k \min_j (d_{it}) = \max_k \min_j (b_{ik}, a_{ij}) \). Consequently,

\[ \max - \min (a_{ij}, b_{ik}) = \max - \min (c_{ip}, d_{it}) \]

**Theorem 4.2.** CE10on is a special case of EMA18on provided that first rows of the \( fpfs \)-matrices are binary.

**Proof.** The proof is similar to that of Theorem 4.1.

\[ \square \]

### 5 Simulation Results

In this section, we compare the running times of CE10an-EMA18an and CE10on-EMA18on by using MATLAB R2017b and a workstation with I(R) Xeon(R) CPU E5-1620 v4 @ 3.5 GHz and 64 GB RAM.

We, firstly, present the running times of CE10an and EMA18an in Table 1 and Fig. 1 for 10 objects and the parameters ranging from 10 to 100. We then give their running times in Table 2 and Fig. 2 for 10 objects and the parameters ranging from 1000 to 10000, in Table 3 and Fig. 3 for 10 parameters and the objects ranging from 10 to 100, in Table 4 and Fig. 4 for 10 parameters and the objects ranging from 1000 to 10000, in Table 5 and Fig. 5 for the parameters and the objects ranging from 10 to 100, and in Table 6 and Fig. 6 for the parameters and the objects ranging from 100 to 1000. The results show that EMA18an outperforms CE10an in any number of data under the specified condition.

**Table 1.** The results for 10 objects and the parameters ranging from 10 to 100

<table>
<thead>
<tr>
<th></th>
<th>10</th>
<th>20</th>
<th>30</th>
<th>40</th>
<th>50</th>
<th>60</th>
<th>70</th>
<th>80</th>
<th>90</th>
<th>100</th>
</tr>
</thead>
<tbody>
<tr>
<td>CE10an</td>
<td>0.02798</td>
<td>0.01283</td>
<td>0.00623</td>
<td>0.00531</td>
<td>0.00513</td>
<td>0.00829</td>
<td>0.00966</td>
<td>0.01325</td>
<td>0.01637</td>
<td>0.01919</td>
</tr>
<tr>
<td>EMA18an</td>
<td>0.01249</td>
<td>0.00714</td>
<td>0.00090</td>
<td>0.00052</td>
<td>0.00244</td>
<td>0.00066</td>
<td>0.00039</td>
<td>0.00035</td>
<td>0.00048</td>
<td>0.00024</td>
</tr>
<tr>
<td>Difference</td>
<td>0.0155</td>
<td>0.0057</td>
<td>0.0053</td>
<td>0.0048</td>
<td>0.0086</td>
<td>0.0076</td>
<td>0.0093</td>
<td>0.0129</td>
<td>0.0159</td>
<td>0.0189</td>
</tr>
<tr>
<td>Advantage (%)</td>
<td>55.3709</td>
<td>44.3108</td>
<td>85.5050</td>
<td>90.1242</td>
<td>77.8817</td>
<td>92.0866</td>
<td>95.9250</td>
<td>97.3876</td>
<td>97.0461</td>
<td>98.7574</td>
</tr>
</tbody>
</table>
Fig. 1. The figure for Table 1

Table 2. The results for 10 objects and the parameters ranging from 1000 to 10000

<table>
<thead>
<tr>
<th>Parameter (e)</th>
<th>1000</th>
<th>2000</th>
<th>3000</th>
<th>4000</th>
<th>5000</th>
<th>6000</th>
<th>7000</th>
<th>8000</th>
<th>9000</th>
<th>10000</th>
</tr>
</thead>
<tbody>
<tr>
<td>CE10an</td>
<td>1.7420</td>
<td>5.9795</td>
<td>12.4333</td>
<td>21.8066</td>
<td>34.2186</td>
<td>46.9271</td>
<td>66.0375</td>
<td>88.0452</td>
<td>110.2487</td>
<td>143.4280</td>
</tr>
<tr>
<td>EMA18an</td>
<td>0.0140</td>
<td>0.0050</td>
<td>0.0024</td>
<td>0.0027</td>
<td>0.0051</td>
<td>0.0053</td>
<td>0.0039</td>
<td>0.0044</td>
<td>0.0048</td>
<td>0.0049</td>
</tr>
<tr>
<td>Difference</td>
<td>1.7280</td>
<td>5.9745</td>
<td>12.4310</td>
<td>21.7979</td>
<td>34.2135</td>
<td>46.9218</td>
<td>66.0336</td>
<td>88.0408</td>
<td>110.2439</td>
<td>143.4230</td>
</tr>
</tbody>
</table>

Fig. 2. The figure for Table 2

Table 3. The results for 10 parameters and the objects ranging from 10 to 100

<table>
<thead>
<tr>
<th>Parameter (e)</th>
<th>10</th>
<th>20</th>
<th>30</th>
<th>40</th>
<th>50</th>
<th>60</th>
<th>70</th>
<th>80</th>
<th>90</th>
<th>100</th>
</tr>
</thead>
<tbody>
<tr>
<td>CE10an</td>
<td>0.0229</td>
<td>0.0087</td>
<td>0.0025</td>
<td>0.0023</td>
<td>0.0066</td>
<td>0.0095</td>
<td>0.0060</td>
<td>0.0060</td>
<td>0.0064</td>
<td>0.0072</td>
</tr>
<tr>
<td>EMA18an</td>
<td>0.0094</td>
<td>0.0040</td>
<td>0.0008</td>
<td>0.0009</td>
<td>0.0024</td>
<td>0.0023</td>
<td>0.0011</td>
<td>0.0012</td>
<td>0.0012</td>
<td>0.0018</td>
</tr>
<tr>
<td>Difference</td>
<td>0.0136</td>
<td>0.0048</td>
<td>0.0017</td>
<td>0.0015</td>
<td>0.0042</td>
<td>0.0072</td>
<td>0.0048</td>
<td>0.0048</td>
<td>0.0053</td>
<td>0.0054</td>
</tr>
<tr>
<td>Advantage (%)</td>
<td>59.1357</td>
<td>54.5995</td>
<td>67.8236</td>
<td>62.7065</td>
<td>63.9276</td>
<td>76.1559</td>
<td>81.2134</td>
<td>80.4675</td>
<td>81.6589</td>
<td>74.9437</td>
</tr>
</tbody>
</table>
**Fig. 3.** The figure for Table 3

**Table 4.** The results for 10 parameters and the objects ranging from 1000 to 10000

<table>
<thead>
<tr>
<th></th>
<th>1000</th>
<th>2000</th>
<th>3000</th>
<th>4000</th>
<th>5000</th>
<th>6000</th>
<th>7000</th>
<th>8000</th>
<th>9000</th>
<th>10000</th>
</tr>
</thead>
<tbody>
<tr>
<td>CE10an</td>
<td>0.1075</td>
<td>0.2303</td>
<td>0.4306</td>
<td>0.6850</td>
<td>1.0900</td>
<td>1.4666</td>
<td>1.9348</td>
<td>2.5576</td>
<td>3.1432</td>
<td>3.8415</td>
</tr>
<tr>
<td>EMA18an</td>
<td>0.0199</td>
<td>0.0250</td>
<td>0.0324</td>
<td>0.0447</td>
<td>0.0594</td>
<td>0.0736</td>
<td>0.0742</td>
<td>0.0993</td>
<td>0.1153</td>
<td>0.1313</td>
</tr>
<tr>
<td>Difference</td>
<td>0.0877</td>
<td>0.2053</td>
<td>0.3982</td>
<td>0.6404</td>
<td>1.0306</td>
<td>1.3930</td>
<td>1.8605</td>
<td>2.4583</td>
<td>3.0280</td>
<td>3.7102</td>
</tr>
<tr>
<td>Advantage (%)</td>
<td>81.5272</td>
<td>89.1420</td>
<td>92.4812</td>
<td>93.4776</td>
<td>94.5528</td>
<td>94.9825</td>
<td>96.1639</td>
<td>96.1160</td>
<td>96.3331</td>
<td>96.5811</td>
</tr>
</tbody>
</table>

**Fig. 4.** The figure for Table 4

**Table 5.** The results for the parameters and the objects ranging from 10 to 100

<table>
<thead>
<tr>
<th></th>
<th>10</th>
<th>20</th>
<th>30</th>
<th>40</th>
<th>50</th>
<th>60</th>
<th>70</th>
<th>80</th>
<th>90</th>
<th>100</th>
</tr>
</thead>
<tbody>
<tr>
<td>CE10an</td>
<td>0.0213</td>
<td>0.0109</td>
<td>0.0078</td>
<td>0.0166</td>
<td>0.0378</td>
<td>0.0645</td>
<td>0.0863</td>
<td>0.1156</td>
<td>0.1665</td>
<td>0.2299</td>
</tr>
<tr>
<td>EMA18an</td>
<td>0.0093</td>
<td>0.0041</td>
<td>0.0009</td>
<td>0.0009</td>
<td>0.0048</td>
<td>0.0023</td>
<td>0.0011</td>
<td>0.0014</td>
<td>0.0014</td>
<td>0.0014</td>
</tr>
<tr>
<td>Difference</td>
<td>0.0121</td>
<td>0.0069</td>
<td>0.0069</td>
<td>0.0157</td>
<td>0.0330</td>
<td>0.0622</td>
<td>0.0851</td>
<td>0.1142</td>
<td>0.1651</td>
<td>0.2285</td>
</tr>
<tr>
<td>Advantage (%)</td>
<td>56.4639</td>
<td>62.7094</td>
<td>88.6511</td>
<td>94.5591</td>
<td>87.3928</td>
<td>96.4563</td>
<td>98.6770</td>
<td>98.8164</td>
<td>99.1380</td>
<td>99.3720</td>
</tr>
</tbody>
</table>
Fig. 5. The figure for Table 5

Table 6. The results for the parameters and the objects ranging from 100 to 1000

<table>
<thead>
<tr>
<th></th>
<th>100</th>
<th>200</th>
<th>300</th>
<th>400</th>
<th>500</th>
<th>600</th>
<th>700</th>
<th>800</th>
<th>900</th>
<th>1000</th>
</tr>
</thead>
<tbody>
<tr>
<td>CE10an</td>
<td>0.2739</td>
<td>3.2532</td>
<td>14.0127</td>
<td>40.1959</td>
<td>93.9178</td>
<td>184.5333</td>
<td>335.5700</td>
<td>568.7381</td>
<td>914.9916</td>
<td>1412.0988</td>
</tr>
<tr>
<td>EMA18an</td>
<td>0.0113</td>
<td>0.0069</td>
<td>0.0068</td>
<td>0.0101</td>
<td>0.0162</td>
<td>0.0200</td>
<td>0.0244</td>
<td>0.0287</td>
<td>0.0396</td>
<td>0.0506</td>
</tr>
<tr>
<td>Difference</td>
<td>0.2626</td>
<td>3.2463</td>
<td>14.0060</td>
<td>40.1858</td>
<td>93.9085</td>
<td>184.5134</td>
<td>335.5456</td>
<td>568.6794</td>
<td>914.9520</td>
<td>1412.0482</td>
</tr>
</tbody>
</table>

Fig. 6. The figure for Table 6

Secondly, we present the running times of CE10on and EMA18on in Table 7 and Fig. 7 for 10 objects and the parameters ranging from 10 to 100. We then give their running times in Table 8 and Fig. 8 for 10 objects and the parameters ranging from 1000 to 10000, in Table 9 and Fig. 9 for 10 parameters and the objects ranging from 10 to 100, in Table 10 and Fig. 10 for 10 parameters and the objects ranging from 1000 to 10000, in Table 11 and Fig. 11 for the parameters and the objects ranging from 10 to 100, and in Table 12 and Fig. 12 for the parameters and the objects ranging from 100 to 1000. The results show that EMA18on outperforms CE10on in any number of data under the specified condition.
Table 7. The results for 10 objects and the parameters ranging from 10 to 100

<table>
<thead>
<tr>
<th></th>
<th>10</th>
<th>20</th>
<th>30</th>
<th>40</th>
<th>50</th>
<th>60</th>
<th>70</th>
<th>80</th>
<th>90</th>
<th>100</th>
</tr>
</thead>
<tbody>
<tr>
<td>CE10on</td>
<td>0.0273</td>
<td>0.0107</td>
<td>0.0037</td>
<td>0.0051</td>
<td>0.0116</td>
<td>0.0165</td>
<td>0.0141</td>
<td>0.0167</td>
<td>0.0245</td>
<td>0.0197</td>
</tr>
<tr>
<td>EMA18on</td>
<td>0.0136</td>
<td>0.0056</td>
<td>0.0007</td>
<td>0.0008</td>
<td>0.0027</td>
<td>0.0020</td>
<td>0.0006</td>
<td>0.0006</td>
<td>0.0004</td>
<td>0.0004</td>
</tr>
<tr>
<td>Difference</td>
<td>0.0137</td>
<td>0.0050</td>
<td>0.0029</td>
<td>0.0044</td>
<td>0.0089</td>
<td>0.0145</td>
<td>0.0135</td>
<td>0.0162</td>
<td>0.0241</td>
<td>0.0193</td>
</tr>
<tr>
<td>Advantage (%)</td>
<td>50.1693</td>
<td>47.3241</td>
<td>80.2441</td>
<td>85.2661</td>
<td>76.7704</td>
<td>88.1753</td>
<td>95.9501</td>
<td>96.4667</td>
<td>98.3700</td>
<td>98.0986</td>
</tr>
</tbody>
</table>

Fig. 7. The figure for Table 7

Table 8. The results for 10 objects and the parameters ranging from 1000 to 10000

<table>
<thead>
<tr>
<th></th>
<th>1000</th>
<th>2000</th>
<th>3000</th>
<th>4000</th>
<th>5000</th>
<th>6000</th>
<th>7000</th>
<th>8000</th>
<th>9000</th>
<th>10000</th>
</tr>
</thead>
<tbody>
<tr>
<td>CE10on</td>
<td>1.7399</td>
<td>6.0070</td>
<td>12.6272</td>
<td>21.8605</td>
<td>34.3464</td>
<td>47.4500</td>
<td>68.2726</td>
<td>90.2301</td>
<td>112.5468</td>
<td>145.8467</td>
</tr>
<tr>
<td>EMA18on</td>
<td>0.0111</td>
<td>0.0061</td>
<td>0.0024</td>
<td>0.0028</td>
<td>0.0053</td>
<td>0.0052</td>
<td>0.0040</td>
<td>0.0042</td>
<td>0.0051</td>
<td>0.0053</td>
</tr>
<tr>
<td>Difference</td>
<td>1.7287</td>
<td>6.0009</td>
<td>12.6249</td>
<td>21.8577</td>
<td>34.3411</td>
<td>47.4448</td>
<td>68.2687</td>
<td>90.2259</td>
<td>112.5417</td>
<td>145.8414</td>
</tr>
</tbody>
</table>

Fig. 8. The figure for Table 8

Table 9. The results for 10 parameters and the objects ranging from 10 to 100

<table>
<thead>
<tr>
<th></th>
<th>10</th>
<th>20</th>
<th>30</th>
<th>40</th>
<th>50</th>
<th>60</th>
<th>70</th>
<th>80</th>
<th>90</th>
<th>100</th>
</tr>
</thead>
<tbody>
<tr>
<td>CE10on</td>
<td>0.0225</td>
<td>0.0087</td>
<td>0.0022</td>
<td>0.0023</td>
<td>0.0066</td>
<td>0.0084</td>
<td>0.0044</td>
<td>0.0036</td>
<td>0.0043</td>
<td>0.0054</td>
</tr>
<tr>
<td>EMA18on</td>
<td>0.0101</td>
<td>0.0048</td>
<td>0.0007</td>
<td>0.0008</td>
<td>0.0030</td>
<td>0.0023</td>
<td>0.0010</td>
<td>0.0009</td>
<td>0.0012</td>
<td>0.0013</td>
</tr>
<tr>
<td>Difference</td>
<td>0.0124</td>
<td>0.0039</td>
<td>0.0014</td>
<td>0.0015</td>
<td>0.0036</td>
<td>0.0062</td>
<td>0.0034</td>
<td>0.0027</td>
<td>0.0031</td>
<td>0.0041</td>
</tr>
<tr>
<td>Advantage (%)</td>
<td>55.1341</td>
<td>44.7449</td>
<td>66.3800</td>
<td>66.7695</td>
<td>54.5929</td>
<td>73.0843</td>
<td>76.7851</td>
<td>74.1328</td>
<td>72.7223</td>
<td>75.7520</td>
</tr>
</tbody>
</table>
Table 10. The results for 10 parameters and the objects ranging from 1000 to 10000

<table>
<thead>
<tr>
<th></th>
<th>1000</th>
<th>2000</th>
<th>3000</th>
<th>4000</th>
<th>5000</th>
<th>6000</th>
<th>7000</th>
<th>8000</th>
<th>9000</th>
<th>10000</th>
</tr>
</thead>
<tbody>
<tr>
<td>CE10on</td>
<td>0.1106</td>
<td>0.2317</td>
<td>0.4371</td>
<td>0.6954</td>
<td>1.0671</td>
<td>1.3376</td>
<td>1.9939</td>
<td>2.5698</td>
<td>3.2157</td>
<td>4.0413</td>
</tr>
<tr>
<td>EMA18on</td>
<td>0.0220</td>
<td>0.0241</td>
<td>0.0299</td>
<td>0.0409</td>
<td>0.0550</td>
<td>0.0676</td>
<td>0.0779</td>
<td>0.0908</td>
<td>0.1062</td>
<td>0.1203</td>
</tr>
<tr>
<td>Difference</td>
<td>0.0886</td>
<td>0.2076</td>
<td>0.4072</td>
<td>0.6545</td>
<td>1.0121</td>
<td>1.3691</td>
<td>1.9160</td>
<td>2.4790</td>
<td>3.1094</td>
<td>3.9210</td>
</tr>
<tr>
<td>Advantage (%)</td>
<td>80.1058</td>
<td>89.5835</td>
<td>93.1572</td>
<td>94.1181</td>
<td>94.8425</td>
<td>95.5995</td>
<td>96.0947</td>
<td>96.4049</td>
<td>96.6968</td>
<td>97.0232</td>
</tr>
</tbody>
</table>

Table 11. The results for the parameters and the objects ranging from 10 to 100

<table>
<thead>
<tr>
<th></th>
<th>10</th>
<th>20</th>
<th>30</th>
<th>40</th>
<th>50</th>
<th>60</th>
<th>70</th>
<th>80</th>
<th>90</th>
<th>100</th>
</tr>
</thead>
<tbody>
<tr>
<td>CE10on</td>
<td>0.0207</td>
<td>0.0108</td>
<td>0.0084</td>
<td>0.0170</td>
<td>0.0343</td>
<td>0.0629</td>
<td>0.0872</td>
<td>0.1145</td>
<td>0.1688</td>
<td>0.2283</td>
</tr>
<tr>
<td>EMA18on</td>
<td>0.0108</td>
<td>0.0045</td>
<td>0.0016</td>
<td>0.0011</td>
<td>0.0031</td>
<td>0.0024</td>
<td>0.0012</td>
<td>0.0013</td>
<td>0.0017</td>
<td>0.0016</td>
</tr>
<tr>
<td>Difference</td>
<td>0.0099</td>
<td>0.0063</td>
<td>0.0068</td>
<td>0.0159</td>
<td>0.0112</td>
<td>0.0605</td>
<td>0.0860</td>
<td>0.1132</td>
<td>0.1670</td>
<td>0.2267</td>
</tr>
<tr>
<td>Advantage (%)</td>
<td>47.7823</td>
<td>58.1857</td>
<td>81.1154</td>
<td>93.7711</td>
<td>90.8651</td>
<td>96.1574</td>
<td>98.6538</td>
<td>98.8066</td>
<td>98.9715</td>
<td>99.3208</td>
</tr>
</tbody>
</table>
Fig. 11. The figure for Table 11

Table 12. The results for the parameters and the objects ranging from 100 to 1000

<table>
<thead>
<tr>
<th></th>
<th>100</th>
<th>200</th>
<th>300</th>
<th>400</th>
<th>500</th>
<th>600</th>
<th>700</th>
<th>800</th>
<th>900</th>
<th>1000</th>
</tr>
</thead>
<tbody>
<tr>
<td>CE10on</td>
<td>0.2714</td>
<td>3.2456</td>
<td>14.0665</td>
<td>40.5834</td>
<td>93.4571</td>
<td>182.9835</td>
<td>325.6018</td>
<td>545.3415</td>
<td>870.2537</td>
<td>1329.9690</td>
</tr>
<tr>
<td>EMA18on</td>
<td>0.0116</td>
<td>0.0079</td>
<td>0.0062</td>
<td>0.0094</td>
<td>0.0163</td>
<td>0.0187</td>
<td>0.0236</td>
<td>0.0295</td>
<td>0.0381</td>
<td>0.0463</td>
</tr>
<tr>
<td>Difference</td>
<td>0.2598</td>
<td>3.2377</td>
<td>14.0573</td>
<td>40.5741</td>
<td>93.4408</td>
<td>182.9648</td>
<td>325.5782</td>
<td>545.3119</td>
<td>870.2156</td>
<td>1329.9228</td>
</tr>
</tbody>
</table>

Fig. 12. The figure for Table 12

6 An Application of EMA18on

Being one of the most important topics in image processing, the noise removal directly affects the success rate of the procedures such as segmentation and classification. For this reason, the determining of the methods which perform better than the others is worthwhile to study.

In this section, in Table 13, we present the mean results of some well-known salt-and-pepper noise (SPN) removal methods Decision Based Algorithm (DBA) [36], Modified Decision Based Unsymmetrical Trimmed Median Filter (MDBUTMF) [37], Noise Adaptive Fuzzy Switching Median Filter (NAFSMF) [38]), Different Applied Median Filter (DAMF) [39], and Adaptive Weighted Mean Filter (AWMF) [40] by using 15 traditional images (Cameraman, Lena, Peppers, Baboon, Plane, Bridge, Pirate, Elaine, Boat, Lake, Flintstones, Living Room, House, Parrot, and Hill) with
and the decision set are as follows:

$$\mathbf{fpfs}$$

to the methods used in the noise removal, so that we can order them in terms of performance. Finally, we then apply EMA18on to a performance-based value assignment to the methods used in the noise removal, so that we can order them in terms of performance.

**Table 13.** The mean-SSIM results of the algorithms for the 15 Traditional Images

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>10%</th>
<th>20%</th>
<th>30%</th>
<th>40%</th>
<th>50%</th>
<th>60%</th>
<th>70%</th>
<th>80%</th>
<th>90%</th>
</tr>
</thead>
<tbody>
<tr>
<td>DBA</td>
<td>0.9655</td>
<td>0.9211</td>
<td>0.8605</td>
<td>0.7837</td>
<td>0.6915</td>
<td>0.5895</td>
<td>0.4846</td>
<td>0.3864</td>
<td>0.3138</td>
</tr>
<tr>
<td>MDBUTMF</td>
<td>0.9428</td>
<td>0.7961</td>
<td>0.8380</td>
<td>0.8391</td>
<td>0.7830</td>
<td>0.6322</td>
<td>0.3228</td>
<td>0.0969</td>
<td>0.0213</td>
</tr>
<tr>
<td>NAFSM</td>
<td>0.9753</td>
<td>0.9506</td>
<td>0.9244</td>
<td>0.8968</td>
<td>0.8660</td>
<td>0.8312</td>
<td>0.7888</td>
<td>0.7308</td>
<td>0.6094</td>
</tr>
<tr>
<td>DAMF</td>
<td>0.9865</td>
<td>0.9715</td>
<td>0.9538</td>
<td>0.9330</td>
<td>0.9083</td>
<td>0.8788</td>
<td>0.8412</td>
<td>0.7883</td>
<td>0.6975</td>
</tr>
<tr>
<td>AWMF</td>
<td>0.9738</td>
<td>0.9639</td>
<td>0.9507</td>
<td>0.9343</td>
<td>0.9133</td>
<td>0.8857</td>
<td>0.8481</td>
<td>0.7943</td>
<td>0.7044</td>
</tr>
</tbody>
</table>

**Table 14.** The mean running-time results of the algorithms for the 15 Traditional Images

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>10%</th>
<th>20%</th>
<th>30%</th>
<th>40%</th>
<th>50%</th>
<th>60%</th>
<th>70%</th>
<th>80%</th>
<th>90%</th>
</tr>
</thead>
<tbody>
<tr>
<td>DAMF</td>
<td>0.1567</td>
<td>0.3008</td>
<td>0.4478</td>
<td>0.5929</td>
<td>0.7399</td>
<td>0.8903</td>
<td>1.0464</td>
<td>1.2319</td>
<td>1.5205</td>
</tr>
<tr>
<td>AWMF</td>
<td>3.9340</td>
<td>3.2274</td>
<td>2.9008</td>
<td>2.7226</td>
<td>2.6228</td>
<td>2.5688</td>
<td>2.5946</td>
<td>2.7314</td>
<td>3.1366</td>
</tr>
</tbody>
</table>

Let’s suppose that the success in low or high-noise density is more important than in the others. Furthermore, the long running time is a drawback. In that case, the values in Table 13 can be represented as an $fpfs$-matrices as follows:

$$[a_{ij}] := \begin{bmatrix}
0.9 & 0.7 & 0.5 & 0.3 & 0.1 & 0.3 & 0.5 & 0.7 & 0.9 \\
0.9655 & 0.9211 & 0.8605 & 0.7837 & 0.6915 & 0.5895 & 0.4846 & 0.3864 & 0.3138 \\
0.9428 & 0.7961 & 0.8380 & 0.8391 & 0.7830 & 0.6322 & 0.3228 & 0.0969 & 0.0213 \\
0.9753 & 0.9506 & 0.9244 & 0.8968 & 0.8660 & 0.8312 & 0.7888 & 0.7308 & 0.6094 \\
0.9865 & 0.9715 & 0.9538 & 0.9330 & 0.9083 & 0.8788 & 0.8412 & 0.7883 & 0.6975 \\
0.9738 & 0.9639 & 0.9507 & 0.9343 & 0.9133 & 0.8857 & 0.8481 & 0.7943 & 0.7044
\end{bmatrix}$$

Similarly, the values given in Table 14 can be represented as an $fpfs$-matrices via the function $f : [0, 15] \to [0, 1]$ defined by $f(x) = 1 - x/15$, as follows:

$$[b_{ij}] := \begin{bmatrix}
0.1 & 0.3 & 0.5 & 0.7 & 0.9 & 0.7 & 0.5 & 0.3 & 0.1 \\
0.7498 & 0.7485 & 0.7478 & 0.7487 & 0.7484 & 0.7473 & 0.7470 & 0.7478 & 0.7476 \\
0.8336 & 0.7393 & 0.6275 & 0.5566 & 0.5138 & 0.4854 & 0.4676 & 0.4579 & 0.4551 \\
0.9165 & 0.8356 & 0.7540 & 0.6746 & 0.5942 & 0.5132 & 0.4342 & 0.3585 & 0.2839 \\
0.9896 & 0.9799 & 0.9701 & 0.9605 & 0.9507 & 0.9406 & 0.9302 & 0.9179 & 0.8866 \\
0.7377 & 0.7848 & 0.8066 & 0.8185 & 0.8251 & 0.8287 & 0.8270 & 0.8179 & 0.7909
\end{bmatrix}$$

If we apply EMA18on to the $fpfs$-matrices $[a_{ij}]$ and $[b_{ij}]$, then the score matrix and the decision set are as follows:

$$[s_{i1}] = \begin{bmatrix}0.8689 & 0.8485 & 0.8778 & 0.8879 & 0.8764\end{bmatrix}^T$$
and 
\[ \{0.9786 \text{DBA}, 0.9556 \text{MDBUTMF}, 0.9886 \text{NAFSM}, 1 \text{DAMF}, 0.9871 \text{AWMF}\} \]

The scores show that DAMF outperforms the other methods and the order DAMF, NAFSM, AWMF, DBA, and MDBUTMF is valid.

Let’s suppose that the success in medium-noise density is more important than in the others. Furthermore, the long running time is a drawback. In that case, the values in Table 13 can be represented as an \( fpfs\)-matrices as follows:

\[
[c_{ij}] := \begin{bmatrix}
0.1 & 0.3 & 0.5 & 0.7 & 0.9 & 0.7 & 0.5 & 0.3 & 0.1 \\
0.9655 & 0.9211 & 0.8605 & 0.7837 & 0.6915 & 0.5895 & 0.4846 & 0.3864 & 0.3138 \\
0.9428 & 0.7961 & 0.8380 & 0.8391 & 0.7830 & 0.6322 & 0.3228 & 0.0969 & 0.0213 \\
0.9753 & 0.9506 & 0.9244 & 0.8968 & 0.8660 & 0.8312 & 0.7888 & 0.7308 & 0.6094 \\
0.9865 & 0.9715 & 0.9538 & 0.9330 & 0.9083 & 0.8788 & 0.8412 & 0.7883 & 0.6975 \\
0.9738 & 0.9639 & 0.9507 & 0.9343 & 0.9133 & 0.8857 & 0.8481 & 0.7943 & 0.7044
\end{bmatrix}
\]

and

\[
[d_{ij}] := \begin{bmatrix}
0.9 & 0.7 & 0.5 & 0.3 & 0.1 & 0.3 & 0.5 & 0.7 & 0.9 \\
0.7498 & 0.7485 & 0.7478 & 0.7487 & 0.7484 & 0.7473 & 0.7470 & 0.7478 & 0.7476 \\
0.8336 & 0.7393 & 0.6275 & 0.5566 & 0.5138 & 0.4854 & 0.4676 & 0.4579 & 0.4551 \\
0.9165 & 0.8356 & 0.7540 & 0.6746 & 0.5942 & 0.5132 & 0.4342 & 0.3585 & 0.2839 \\
0.9896 & 0.9799 & 0.9701 & 0.9605 & 0.9507 & 0.9406 & 0.9302 & 0.9179 & 0.8986 \\
0.7377 & 0.7848 & 0.8066 & 0.8185 & 0.8251 & 0.8287 & 0.8270 & 0.8179 & 0.7909
\end{bmatrix}
\]

If we apply EMA18on to the \( fpfs\)-matrices \([c_{ij}]\) and \([d_{ij}]\), then the score matrix and the decision set are as follows:

\[
[s_{i1}] = [0.6748 \ 0.7502 \ 0.8248 \ 0.8906 \ 0.8220]^T
\]

and 
\[ \{0.7577 \text{DBA}, 0.8421 \text{MDBUTMF}, 0.9262 \text{NAFSM}, 1 \text{DAMF}, 0.9229 \text{AWMF}\} \]

The scores show that DAMF performs better than the other methods and the order DAMF, NAFSM, AWMF, MDBUTMF, and DBA is valid.

### 7 Conclusion

The uni-int decision-making method was defined in 2010 [5]. Afterwards, this method has been configured [34] via \( fpfs\)-matrices [11]. However, the method suffers from a drawback, i.e. its incapability of processing a large amount of parameters on a standard computer, e.g. with 2.6 GHz i5 Dual Core CPU and 4GB RAM. For this reason, simplification of such methods is significant for a wide range of scientific and industrial processes. In this study, firstly, we have proposed two fast and simple soft decision-making methods EMA18an and EMA18on. Moreover, we have proved that these two methods accept CE10 as a special case, under the condition that the first rows of the \( fpfs\)-matrices are binary. It is also possible to investigate the simplifications of the other products such as andnot-product and ornot-product (see Definition 2.5).
We then have compared the running times of these algorithms. In addition to the results in Section 4, the results in Table 15 and 16 too show that EMA18an and EMA18on outperform CE10an and CE10on, respectively, in any number of data under the specified condition. Furthermore, other decision-making methods constructed by a different decision function such as minimum-maximum (min-max), max-max, and min-min can also be studied by the similar way.

Table 15. The mean/max advantage and max difference values of EMA18an over CE10an

<table>
<thead>
<tr>
<th>Location</th>
<th>Objects</th>
<th>Parameters</th>
<th>Mean Advantage%</th>
<th>Max Advantage%</th>
<th>Max Difference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table 1</td>
<td>10</td>
<td>10 - 100</td>
<td>83.4395</td>
<td>98.7574</td>
<td>0.0189</td>
</tr>
<tr>
<td>Table 2</td>
<td>10</td>
<td>1000 - 10000</td>
<td>99.9036</td>
<td>99.9966</td>
<td>143.4230</td>
</tr>
<tr>
<td>Table 3</td>
<td>10</td>
<td>10</td>
<td>70.2632</td>
<td>81.6589</td>
<td>0.0136</td>
</tr>
<tr>
<td>Table 4</td>
<td>1000 - 10000</td>
<td>10</td>
<td>93.1357</td>
<td>96.5811</td>
<td>3.7102</td>
</tr>
<tr>
<td>Table 5</td>
<td>10 - 100</td>
<td>10 - 100</td>
<td>88.2236</td>
<td>99.3720</td>
<td>0.2285</td>
</tr>
<tr>
<td>Table 6</td>
<td>100 - 1000</td>
<td>100 - 10000</td>
<td>99.5547</td>
<td>99.9964</td>
<td>1412.0482</td>
</tr>
</tbody>
</table>

Table 16. The mean/max advantage and max difference values of EMA18on over CE10on

<table>
<thead>
<tr>
<th>Location</th>
<th>Objects</th>
<th>Parameters</th>
<th>Mean Advantage%</th>
<th>Max Advantage%</th>
<th>Max Difference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table 1</td>
<td>10</td>
<td>10 - 100</td>
<td>81.6835</td>
<td>98.3700</td>
<td>0.0241</td>
</tr>
<tr>
<td>Table 2</td>
<td>10</td>
<td>1000 - 10000</td>
<td>99.9181</td>
<td>99.9964</td>
<td>145.8414</td>
</tr>
<tr>
<td>Table 3</td>
<td>10 - 100</td>
<td>10</td>
<td>66.0098</td>
<td>76.7851</td>
<td>0.0124</td>
</tr>
<tr>
<td>Table 4</td>
<td>1000 - 10000</td>
<td>10</td>
<td>93.3686</td>
<td>97.0232</td>
<td>3.9210</td>
</tr>
<tr>
<td>Table 5</td>
<td>10 - 100</td>
<td>10 - 100</td>
<td>86.3720</td>
<td>99.3208</td>
<td>0.2267</td>
</tr>
<tr>
<td>Table 6</td>
<td>100 - 1000</td>
<td>100 - 10000</td>
<td>99.5361</td>
<td>99.9965</td>
<td>1329.9228</td>
</tr>
</tbody>
</table>

Finally, we have applied EMA18on to the determination of the performance of the known methods. It is clear that EMA18on, which is a fast and simple method, can be successfully applied to the decision-making problems in various areas such as machine learning and image enhancement.

Although we have no proof about the accuracy of the results of such methods, the results are in compliance with our observations. In order to help in checking the accuracy of the comparison made by a soft decision-making method, we give, in Fig. 13-16, the Cameraman image with different SPN ratios and show the denoised images via the above-mentioned filters. It must be noted that these images has no information of their running times. Whereas, the use of a filter in a software depends on its running time is short. In other words, the running time of a filter is so significant that it can not be ignored. As a result, it is understood that \( fpfs \)-matrices are an effective mathematical tool to deal with the situations in which more than one parameter or objects are used.
Fig. 13. (a) Original image “Cameraman” (b) Noisy image with SPN ratio of 10%, (c) Noisy image with SPN ratio of 50%, and (d) Noisy image with SPN ratio of 90%.

Fig. 14. The images having with SPN ratio of 10% before denoising.

Fig. 15. The images having with SPN ratio of 50% before denoising.

Fig. 16. The images having with SPN ratio of 90% before denoising.

Acknowledgements

We thank Dr. Uğur Erkan for technical support. This work was supported by the Office of Scientific Research Projects Coordination at Çanakkale Onsekiz Mart University, Grant number: FBA-2018-1367.
References


