Data Envelopment Analysis for Simplified Neutrosophic Sets

S. A. Edalatpanah¹* and F. Smarandache²

¹ Department of Applied Mathematics, Ayandegan Institute of Higher Education, Tonekabon, Iran; E-mail: saedalatpanah@gmail.com
² University of New Mexico, 705 Gurley Ave., Gallup, New Mexico 87301, USA; E-mail: smarand@unm.edu
* Correspondence: saedalatpanah@aihe.ac.ir; Tel.: +981154310428.

Abstract: In recent years, there has been a growing interest in neutrosophic theory, and there are several methods for solving various problems under neutrosophic environment. However, a few papers have discussed the Data envelopment analysis (DEA) with neutrosophic sets. So, in this paper, we propose an input-oriented DEA model with simplified neutrosophic numbers and present a new strategy to solve it. The proposed method is based on the weighted arithmetic average operator and has a simple structure. Finally, the new approach is illustrated with the help of a numerical example.

Keywords: Data envelopment analysis; Neutrosophic set; Simplified neutrosophic sets (SNSs); Aggregation operator.

1. Introduction

With the advent of technology and the complexity and volume of information, senior executives have required themselves to apply scientific methods to determine and increase the productivity of the organization under their jurisdiction. Data envelopment analysis (DEA) is a mathematical technique to evaluate the relative efficiency of a set of some homogeneous units called decision-making units (DMUs) that use multiple inputs to produce multiple outputs. DMUs are called homogeneous because they all employ the same inputs to produce the same outputs. DEA by constructing an efficiency frontier measures the relative efficiency of decision making units (DMUs). Charnes et al. [1] developed a DEA model (CCR) based on the seminal work of Farrell [2] under the assumption of constant returns to scale (CRS). Banker et al. [3] extended the pioneering work Charnes et al. [1] and proposed a model conventionally called BCC to measure the relative efficiency under the assumption of variable returns to scale (VRS). DEA technique has just been effectively connected in various cases such as broadcasting companies [4], banking institutions [5-8], R&D organizations [9-10], health care services [11-12], manufacturing [13-14], telecommunication [15], and supply chain management [16-19]. However, data in the standard models are certain, but there are numerous circumstances in real life where we have to face uncertain parameters. Zadeh [20] first proposed the theory of fuzzy sets (FSs) against certain logic where the membership degree is a real number between zero and one. After this work, many researchers studied on this topic; details of some researches can be observed in [21-30]. Several researchers also proposed some models of DEA under fuzzy environment [31-42]. However, Zadeh’s fuzzy sets cannot deal with certain cases in which it is difficult to define the membership degree using one specific value. To overcome this lack of knowledge, Atanassov [43] introduced an extension of the FSs that called the intuitionistic fuzzy sets (IFs). Although the theory of IFs can handle incomplete information in various real-world issues, it cannot address all types of uncertainty such as indeterminate and inconsistent information.
Therefore, Smarandache [44-45], proposed the neutrosophic set (NS) as a strong general framework that generalizes the classical set concept, fuzzy set [20], interval-valued fuzzy set [46], intuitionistic fuzzy set [43], and interval-valued intuitionistic fuzzy set [47]. Neutrosophic set (NS) can deal with uncertain, indeterminate and incongruous information where the indeterminacy is quantified explicitly and truth membership, indeterminacy membership and falsity membership are completely independent. It can effectively describe uncertain, incomplete and inconsistent information and overcomes some limitations of the existing methods in depicting uncertain decision information. Moreover, some extensions of NSs, including interval neutrosophic set [48-51], bipolar neutrosophic set [52-54], single-valued neutrosophic set [55-59], simplified neutrosophic sets [60-64], multi-valued neutrosophic set [65-67], and neutrosophic linguistic set [68-70] have been presented and applied to solve various problems; see [71-80].

Although there are several approaches to solving various problems under neutrosophic environment, to the best of our knowledge, there are few investigations regarding DEA with neutrosophic sets. The first attempt has been proposed by Edalatpanah in [81] and further research has been presented in [82]. So, in this paper, we design a model of DEA with simplified neutrosophic numbers (SNNs) and establish a new strategy to solve it. The proposed method is based on the weighted arithmetic average operator and has a simple structure.

This paper organized as follows: some basic knowledge, concepts and arithmetic operations on SNNs are introduced in Section 2. In Section 3, we review some concepts of DEA and the input-oriented BCC model. In Section 4, we introduce the mentioned model of DEA under the simplified neutrosophic environment and propose a method to solve it. In Section 5, an example demonstrates the application of the proposed model. Finally, some conclusions and future research are offered in Section 6.

2. Simplified neutrosophic sets

Smarandache [44-45] has provided a variety of real-life examples for possible applications of his neutrosophic sets; however, it is difficult to apply neutrosophic sets to practical problems. Therefore, Ye [60] reduced neutrosophic sets of non-standard intervals into a kind of simplified neutrosophic sets (SNSs) of standard intervals that will preserve the operations of the neutrosophic sets. In this section, we will review the concept of SNSs, which are a subclass of neutrosophic sets briefly.

**Definition 1** [60]. Let X be a space of points (objects), with a generic element in X denoted by x. A neutrosophic set A in X is characterized by a truth-membership function \( T_A(x) \), an indeterminacy membership function \( I_A(x) \) and a falsity-membership function \( F_A(x) \). If the functions \( T_A(x) \), \( I_A(x) \) and \( F_A(x) \) are singleton subintervals/subsets in the real standard [0, 1], that is \( T_A(x) : X \rightarrow [0,1], I_A(x): x \rightarrow [0,1], F_A(x): X \rightarrow [0,1] \), then, a simplification of the neutrosophic set A is denoted by \( A = \{(x, T_A(x), I_A(x), F_A(x))| x \in X \} \), which is called a SNS. Also, SNS satisfies the condition \( 0 \leq T_A(x) + I_A(x) + F_A(x) \leq 3 \).

**Definition 2** [60]. For SNSs A and B, A \( \leq B \) if and only if \( T_A(x) \leq T_B(x) \), \( I_A(x) \geq I_B(x) \), and \( F_A(x) \geq F_B(x) \) for every \( x \) in X.

**Definition 3** [63]. Let A, B be two SNSs. Then the arithmetic relations are defined as:

\[
\begin{align*}
(i) \quad A \oplus B & = T_A(x) + T_B(x) - T_A(x) T_B(x), I_A(x) I_B(x), F_A(x) F_B(x) >, \\
(ii) \quad A \otimes B & = T_A(x) T_B(x), I_A(x) + I_B(x) - I_A(x) I_B(x), F_A(x) + F_B(x) - F_A(x) F_B(x) >, \\
(iii) \quad A^\lambda & = 1 - (1 - T_A(x))^\lambda, (I_A(x))^\lambda, (F_A(x))^\lambda, \lambda > 0, \\
(iv) \quad A^{-1} & = T_A(x), 1 - (1 - I_A(x))^\lambda, 1 - (1 - F_A(x))^\lambda, \lambda > 0. 
\end{align*}
\]

**Definition 4** [60]. Let \( A_j \) (\( j = 1, 2, \ldots, n \)) be a SNS. The simplified neutrosophic weighted arithmetic average operator is defined as:

\[
F_a(A_1, \ldots, A_n) = \sum_{j=1}^{n} \omega_j A_j 
\]
where \( W = (\omega_1, \omega_2, \ldots, \omega_n) \) is the weight vector of \( A_{ij} \), \( \omega_j \in [0, 1] \) and \( \sum_{j=1}^{n} \omega_j = 1 \).

**Theorem 1** [63]. For the simplified neutrosophic weighted arithmetic average operator, the aggregated result is as follows:

\[
F_n (A_1, \ldots, A_n) = \left\{ 1 - \prod_{j=1}^{n} (1 - T_{A_{ij}} (x))^{\nu_j} \prod_{j=1}^{n} (U_{A_{ij}} (x))^{\nu_j} \prod_{j=1}^{n} (F_{A_{ij}} (x))^{\nu_j} \right\}
\]  

(6)

3. The input-oriented BCC model of DEA

Data envelopment analysis (DEA) is a linear programming method for assessing the efficiency and productivity of decision-making units (DMUs). In the traditional DEA literature, various well-known DEA approaches can be found such as CCR and BCC models [1, 3]. The efficiency of a DMU is established as the ratio of sum weighted output to sum weighted input, subjected to happen between one and zero. Let DMU\( O \) is under consideration, then input-oriented BCC model for the relative efficiency is as follows [3]:

\[
\begin{align*}
\text{Min} & \quad \theta^* \\
\text{s.t.} & \quad \sum_{j=1}^{n} \lambda_j x_{ij} \leq \theta^* x_{io}, \quad i = 1, 2, \ldots, m \\
& \quad \sum_{j=1}^{n} \lambda_j y_{rj} \geq y_{ro}, \quad r = 1, 2, \ldots, s \\
& \quad \sum_{j=1}^{n} \lambda_j = 1 \\
& \quad \lambda_j \geq 0, \quad j = 1, 2, \ldots, n
\end{align*}
\]  

(7)

In this model, each DMU (suppose that we have \( n \) DMUs) uses \( m \) inputs \( x_{ij} \) \((i = 1, 2, \ldots, m)\), to obtains \( s \) outputs \( y_{rj} \) \((r = 1, 2, \ldots, s)\). Here \( u_r, r = 1, 2, \ldots, s \) and \( v_i, i = 1, 2, \ldots, m \), are the weights of the \( i \) th input and \( r \) th output. This model is calculated for every DMU to find out its best input and output weights. If \( \theta^* = 1 \), we say that the DMU\( O \) is efficient otherwise it is inefficient.

4. Simplified Neutrosophic Data Envelopment Analysis

In this section, we establish DEA under simplified neutrosophic environment. Consider the input and output for the \( j \) th DMU as \( x^N = (T_{x_{1j}}, T_{x_{2j}}, T_{x_{mj}}) \), \( y^N = (T_{y_{1j}}, T_{y_{2j}}, T_{y_{sj}}) \) which are the simplified neutrosophic numbers (SNN). Then the simplified neutrosophic BCC model that called SNBCC is defined as follows:

\[
\begin{align*}
\text{Min} & \quad \theta^* \\
\text{s.t.} & \quad \sum_{j=1}^{n} \lambda_j x_{ij}^N \leq \theta^* x_{io}^N, \quad i = 1, 2, \ldots, m \\
& \quad \sum_{j=1}^{n} \lambda_j y_{rj}^N \geq y_{ro}^N, \quad r = 1, 2, \ldots, s \\
& \quad \sum_{j=1}^{n} \lambda_j = 1 \\
& \quad \lambda_j \geq 0, \quad j = 1, 2, \ldots, n
\end{align*}
\]  

(8)

Next, to solve the model (8) we propose the following algorithm:

**Algorithm 1.**

Step 1. Consider the DEA model (8) that the inputs and outputs of each DMU are SNN.
Step 2. Using the Definition 3 and Theorem 1, the SNBCC model of Step 1 can be transformed into the following model:

\[ \text{Min } \theta_o \]

\[ \text{s.t. } \]

\[ \left\{ \begin{array}{l}
1 - \prod_{j=1}^{n} (1-T_{x_i})^{\lambda_j} \leq \left(1 - \prod_{j=1}^{n} (1-T_{x_i})^{\theta_o} \right)^{\theta_o}, \\
1 - \prod_{j=1}^{n} (1-J_{x_i})^{\lambda_j} \leq \left(1 - \prod_{j=1}^{n} (1-J_{x_i})^{\theta_o} \right)^{\theta_o}, \\
1 - \prod_{j=1}^{n} (1-F_{x_i})^{\lambda_j} \leq \left(1 - \prod_{j=1}^{n} (1-F_{x_i})^{\theta_o} \right)^{\theta_o}, \\
\sum_{j=1}^{n} \lambda_j = 1 \\
\lambda_j \geq 0, \quad j = 1, 2, \ldots, n.
\end{array} \right. \]  

(9)

Step 3. Using Definition 2, the SNBCC model of Step 2 can be transformed into the following model:

\[ \text{Min } \theta_o \]

\[ \text{s.t. } \]

\[ \left\{ \begin{array}{l}
\prod_{j=1}^{n} (1-T_{x_i})^{\lambda_j} \geq (1 - T_{x_i})^{\theta_o}, \quad i = 1, 2, \ldots, m \\
\prod_{j=1}^{n} (I_{x_i})^{\lambda_j} \geq (I_{x_i})^{\theta_o}, \quad i = 1, 2, \ldots, m \\
\prod_{j=1}^{n} (F_{x_i})^{\lambda_j} \geq (F_{x_i})^{\theta_o}, \quad i = 1, 2, \ldots, m \\
\prod_{j=1}^{n} (1-T_{y_i})^{\lambda_j} \leq (1 - T_{y_i})^{\theta_o}, \quad r = 1, 2, \ldots, s \\
\prod_{j=1}^{n} (I_{y_i})^{\lambda_j} \leq I_{y_i}, \quad r = 1, 2, \ldots, s \\
\prod_{j=1}^{n} (F_{y_i})^{\lambda_j} \leq F_{y_i}, \quad r = 1, 2, \ldots, s \\
\sum_{j=1}^{n} \lambda_j = 1 \\
\lambda_j \geq 0, \quad j = 1, 2, \ldots, n.
\end{array} \right. \]  

(10)

Step 4. Using the natural logarithm, transform the nonlinear model of (10) into the following linear model:

\[ \text{Min } \theta_o \]

\[ \text{s.t. } \]

\[ \left\{ \begin{array}{l}
\sum_{j=1}^{n} \lambda_j \ln(1-T_{x_i}) \geq \theta_o \ln(1-T_{x_i}), \quad i = 1, 2, \ldots, m \\
\sum_{j=1}^{n} \lambda_j \ln(I_{x_i}) \geq \theta_o \ln(I_{x_i}), \quad i = 1, 2, \ldots, m \\
\sum_{j=1}^{n} \lambda_j \ln(F_{x_i}) \geq \theta_o \ln(F_{x_i}), \quad i = 1, 2, \ldots, m \\
\sum_{j=1}^{n} \lambda_j \ln(1-T_{y_i}) \leq \ln(1-T_{y_i}), \quad r = 1, 2, \ldots, s \\
\sum_{j=1}^{n} \lambda_j \ln(I_{y_i}) \leq \ln(I_{y_i}), \quad r = 1, 2, \ldots, s \\
\sum_{j=1}^{n} \lambda_j \ln(F_{y_i}) \leq \ln(F_{y_i}), \quad r = 1, 2, \ldots, s
\end{array} \right. \]  

(11)

(12)

(13)

(14)

(15)

(16)
\[
\sum_{j=1}^{n} \lambda_j \ln(F_{y_j}) \leq \ln(F_{y_r}), \quad r = 1, 2, ..., s
\]  
\[
\sum_{j=1}^{n} \lambda_j = 1,
\]
\[
\lambda_j \geq 0, \quad j = 1, 2, ..., n.
\]

Step 5. Run model (11) and obtain the optimal solution.

5. Numerical example

In this section, an example of DEA problem under simplified neutrosophic environment is used to demonstrate the validity and effectiveness of the proposed model.

Example 5.1. Consider 10 DMUs with three inputs and outputs where all the input and output data are designed as SNN (see tables 1 and 2).

<table>
<thead>
<tr>
<th>DMUS</th>
<th>Inputs 1</th>
<th>Inputs 2</th>
<th>Inputs 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>DMU1</td>
<td>&lt;0.75, 0.1, 0.15&gt;</td>
<td>&lt;0.75, 0.1, 0.15&gt;</td>
<td>&lt;0.8, 0.05, 0.1&gt;</td>
</tr>
<tr>
<td>DMU2</td>
<td>&lt;0.85, 0.2, 0.15&gt;</td>
<td>&lt;0.6, 0.05, 0.05&gt;</td>
<td>&lt;0.9, 0.1, 0.2&gt;</td>
</tr>
<tr>
<td>DMU3</td>
<td>&lt;0.9, 0.01, 0.05&gt;</td>
<td>&lt;0.95, 0.01, 0.01&gt;</td>
<td>&lt;0.98, 0.01, 0.01&gt;</td>
</tr>
<tr>
<td>DMU4</td>
<td>&lt;0.7, 0.2, 0.15&gt;</td>
<td>&lt;0.65, 0.2, 0.15&gt;</td>
<td>&lt;0.8, 0.05, 0.2&gt;</td>
</tr>
<tr>
<td>DMU5</td>
<td>&lt;0.9, 0.05, 0.1&gt;</td>
<td>&lt;0.95, 0.05, 0.05&gt;</td>
<td>&lt;0.7, 0.2, 0.4&gt;</td>
</tr>
<tr>
<td>DMU6</td>
<td>&lt;0.85, 0.2, 0.1&gt;</td>
<td>&lt;0.7, 0.05, 0.1&gt;</td>
<td>&lt;0.6, 0.2, 0.3&gt;</td>
</tr>
<tr>
<td>DMU7</td>
<td>&lt;0.8, 0.3, 0.1&gt;</td>
<td>&lt;0.9, 0.5, 0.1&gt;</td>
<td>&lt;0.8, 0.1, 0.3&gt;</td>
</tr>
<tr>
<td>DMU8</td>
<td>&lt;0.55, 0.3, 0.35&gt;</td>
<td>&lt;0.8, 0.2, 0.25&gt;</td>
<td>&lt;0.65, 0.2, 0.15&gt;</td>
</tr>
<tr>
<td>DMU9</td>
<td>&lt;0.65, 0.2, 0.25&gt;</td>
<td>&lt;0.7, 0.15, 0.2&gt;</td>
<td>&lt;0.85, 0.2, 0.2&gt;</td>
</tr>
<tr>
<td>DMU10</td>
<td>&lt;0.6, 0.1, 0.3&gt;</td>
<td>&lt;0.8, 0.3, 0.1&gt;</td>
<td>&lt;0.65, 0.2, 0.1&gt;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DMUS</th>
<th>Outputs 1</th>
<th>Outputs 2</th>
<th>Outputs 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>DMU1</td>
<td>&lt;0.7, 0.15, 0.2&gt;</td>
<td>&lt;0.7, 0.15, 0.2&gt;</td>
<td>&lt;0.65, 0.2, 0.25&gt;</td>
</tr>
<tr>
<td>DMU2</td>
<td>&lt;0.15, 0.2, 0.25&gt;</td>
<td>&lt;0.15, 0.2, 0.25&gt;</td>
<td>&lt;0.25, 0.15, 0.05&gt;</td>
</tr>
<tr>
<td>DMU3</td>
<td>&lt;0.75, 0.1, 0.15&gt;</td>
<td>&lt;0.7, 0.15, 0.2&gt;</td>
<td>&lt;0.8, 0.05, 0.1&gt;</td>
</tr>
<tr>
<td>DMU4</td>
<td>&lt;0.5, 0.35, 0.4&gt;</td>
<td>&lt;0.6, 0.25, 0.3&gt;</td>
<td>&lt;0.55, 0.3, 0.35&gt;</td>
</tr>
<tr>
<td>DMU5</td>
<td>&lt;0.6, 0.2, 0.25&gt;</td>
<td>&lt;0.6, 0.15, 0.4&gt;</td>
<td>&lt;0.3, 0.5, 0.5&gt;</td>
</tr>
<tr>
<td>DMU6</td>
<td>&lt;0.55, 0.3, 0.35&gt;</td>
<td>&lt;0.5, 0.5, 0.5&gt;</td>
<td>&lt;0.65, 0.2, 0.3&gt;</td>
</tr>
<tr>
<td>DMU7</td>
<td>&lt;0.8, 0.1, 0.2&gt;</td>
<td>&lt;0.3, 0.01, 0.05&gt;</td>
<td>&lt;0.9, 0.05, 0.05&gt;</td>
</tr>
<tr>
<td>DMU8</td>
<td>&lt;0.8, 0.1, 0.3&gt;</td>
<td>&lt;0.8, 0.25, 0.3&gt;</td>
<td>&lt;0.85, 0.2, 0.2&gt;</td>
</tr>
<tr>
<td>DMU9</td>
<td>&lt;0.65, 0.2, 0.25&gt;</td>
<td>&lt;0.7, 0.15, 0.2&gt;</td>
<td>&lt;0.75, 0.1, 0.15&gt;</td>
</tr>
<tr>
<td>DMU10</td>
<td>&lt;0.6, 0.1, 0.5&gt;</td>
<td>&lt;0.75, 0.1, 0.3&gt;</td>
<td>&lt;0.8, 0.3, 0.5&gt;</td>
</tr>
</tbody>
</table>

Next, we use Algorithm.1 to solve the mentioned performance assessment problem. For example, The Algorithm.1 for DMU1 can be used as follows:

Step 1. Obtain the SNBCC model (8):
Using the Step 4 of Algorithm 1, we have:

**Step 2.** Using the Step 4 of Algorithm 1, we have:

\[
\begin{align*}
\text{Min } & \quad \theta_i \\
\text{s.t. } & \quad \begin{cases} \\
\lambda_1 < 0.75, 0.1, 0.15 > \ominus \lambda_2 < 0.85, 0.2, 0.15 > \ominus \lambda_3 < 0.9, 0.01, 0.05 > \ominus \\
\lambda_4 < 0.7, 0.2, 0.1 > \ominus \lambda_5 < 0.9, 0.05, 0.1 > \ominus \lambda_6 < 0.85, 0.2, 0.1 > \ominus \\
\lambda_7 < 0.8, 0.3, 0.35 > \ominus \lambda_8 < 0.8, 0.05, 0.1 > \ominus \lambda_9 < 0.6, 0.1, 0.3 > \ominus \\
\lambda_{10} < 0.6, 0.1, 0.3 > \\
\end{cases} \\
\leq (\theta_1 < 0.75, 0.1, 0.15 >), \\
\end{align*}
\]

\[
\begin{align*}
\text{s.t. } & \quad \begin{cases} \\
\lambda_1 < 0.7, 0.1, 0.2 > \ominus \lambda_2 < 0.6, 0.05, 0.05 > \ominus \lambda_3 < 0.95, 0.01, 0.01 > \ominus \\
\lambda_4 < 0.65, 0.2, 0.15 > \ominus \lambda_5 < 0.95, 0.05, 0.05 > \ominus \lambda_6 < 0.7, 0.05, 0.1 > \ominus \\
\lambda_7 < 0.9, 0.5, 0.1 > \ominus \lambda_8 < 0.65, 0.2, 0.25 > \ominus \lambda_9 < 0.9, 0.01, 0.05 > \ominus \\
\lambda_{10} < 0.8, 0.3, 0.1 > \\
\end{cases} \\
\leq (\theta_1 < 0.7, 0.1, 0.2 >), \\
\end{align*}
\]

\[
\begin{align*}
\text{s.t. } & \quad \begin{cases} \\
\lambda_1 < 0.8, 0.05, 0.1 > \ominus \lambda_2 < 0.9, 0.1, 0.2 > \ominus \lambda_3 < 0.98, 0.01, 0.01 > \ominus \\
\lambda_4 < 0.8, 0.05, 0.2 > \ominus \lambda_5 < 0.7, 0.2, 0.4 > \ominus \lambda_6 < 0.6, 0.2, 0.3 > \ominus \\
\lambda_7 < 0.8, 0.1, 0.3 > \ominus \lambda_8 < 0.5, 0.35, 0.4 > \ominus \lambda_9 < 0.7, 0.05, 0.1 > \ominus \\
\lambda_{10} < 0.65, 0.2, 0.1 > \\
\end{cases} \\
\leq (\theta_1 < 0.8, 0.05, 0.1 >), \\
\end{align*}
\]

\[
\begin{align*}
\text{s.t. } & \quad \begin{cases} \\
\lambda_1 < 0.7, 0.15, 0.2 > \ominus \lambda_2 < 0.15, 0.2, 0.25 > \ominus \lambda_3 < 0.75, 0.1, 0.15 > \ominus \\
\lambda_4 < 0.5, 0.35, 0.4 > \ominus \lambda_5 < 0.6, 0.2, 0.25 > \ominus \lambda_6 < 0.55, 0.3, 0.35 > \ominus \\
\lambda_7 < 0.8, 0.1, 0.2 > \ominus \lambda_8 < 0.8, 0.1, 0.3 > \ominus \lambda_9 < 0.65, 0.2, 0.25 > \ominus \\
\lambda_{10} < 0.6, 0.1, 0.5 > \\
\end{cases} \\
\geq (0.7, 0.15, 0.2 >), \\
\end{align*}
\]

\[
\begin{align*}
\text{s.t. } & \quad \begin{cases} \\
\lambda_1 < 0.6, 0.1, 0.3 > \ominus \lambda_2 < 0.2, 0.1, 0.3 > \ominus \lambda_3 < 0.7, 0.15, 0.2 > \ominus \\
\lambda_4 < 0.6, 0.25, 0.3 > \ominus \lambda_5 < 0.6, 0.15, 0.4 > \ominus \lambda_6 < 0.5, 0.5, 0.5 > \ominus \\
\lambda_7 < 0.3, 0.01, 0.05 > \ominus \lambda_8 < 0.8, 0.25, 0.3 > \ominus \lambda_9 < 0.7, 0.15, 0.2 > \ominus \\
\lambda_{10} < 0.75, 0.1, 0.3 > \\
\end{cases} \\
\geq (0.6, 0.1, 0.3 >), \\
\end{align*}
\]

\[
\begin{align*}
\text{s.t. } & \quad \begin{cases} \\
\lambda_1 < 0.65, 0.2, 0.25 > \ominus \lambda_2 < 0.25, 0.15, 0.05 > \ominus \lambda_3 < 0.8, 0.05, 0.1 > \ominus \\
\lambda_4 < 0.55, 0.3, 0.35 > \ominus \lambda_5 < 0.3, 0.5, 0.5 > \ominus \lambda_6 < 0.6, 0.25, 0.3 > \ominus \\
\lambda_7 < 0.9, 0.05, 0.05 > \ominus \lambda_8 < 0.85, 0.2, 0.2 > \ominus \lambda_9 < 0.75, 0.1, 0.15 > \ominus \\
\lambda_{10} < 0.8, 0.3, 0.5 > \\
\end{cases} \\
\geq (0.65, 0.2, 0.25 >), \\
\end{align*}
\]

\[
\lambda_1 + \lambda_2 + \lambda_3 + \lambda_4 + \lambda_5 + \lambda_6 + \lambda_7 + \lambda_8 + \lambda_{10} = 1,
\]
\[
\lambda_j \geq 0, \quad j = 1, 2, ..., 10.
\]
\[ \lambda_1 \ln(0.3) + \lambda_2 \ln(0.4) + \lambda_3 \ln(0.05) + \lambda_4 \ln(0.35) + \lambda_5 \ln(0.05) + \lambda_6 \ln(0.3) + \lambda_7 \ln(0.1) + \lambda_8 \ln(0.35) + \lambda_9 \ln(0.1) + \lambda_{10} \ln(0.2) \geq \theta_1 \ln(0.3) \]

\[ \lambda_1 \ln(0.2) + \lambda_2 \ln(0.1) + \lambda_3 \ln(0.02) + \lambda_4 \ln(0.2) + \lambda_5 \ln(0.3) + \lambda_6 \ln(0.4) + \lambda_7 \ln(0.2) + \lambda_8 \ln(0.5) + \lambda_9 \ln(0.3) + \lambda_{10} \ln(0.35) \geq \theta_1 \ln(0.2) \]

(Using Eq. (13))

\[ \lambda_1 \ln(0.1) + \lambda_2 \ln(0.2) + \lambda_3 \ln(0.01) + \lambda_4 \ln(0.2) + \lambda_5 \ln(0.05) + \lambda_6 \ln(0.2) + \lambda_7 \ln(0.1) + \lambda_8 \ln(0.05) + \lambda_9 \ln(0.1) + \lambda_{10} \ln(0.1) \geq \theta_1 \ln(0.1) \]

(Using Eq. (14))

\[ \lambda_1 \ln(0.15) + \lambda_2 \ln(0.15) + \lambda_3 \ln(0.05) + \lambda_4 \ln(0.1) + \lambda_5 \ln(0.1) + \lambda_6 \ln(0.1) + \lambda_7 \ln(0.3) + \lambda_8 \ln(0.05) + \lambda_9 \ln(0.3) + \lambda_{10} \ln(0.3) \geq \theta_1 \ln(0.15) \]

(Using Eq. (15))

\[ \lambda_1 \ln(0.3) + \lambda_2 \ln(0.25) + \lambda_3 \ln(0.5) + \lambda_4 \ln(0.4) + \lambda_5 \ln(0.45) + \lambda_6 \ln(0.2) + \lambda_7 \ln(0.2) + \lambda_8 \ln(0.35) + \lambda_9 \ln(0.4) + \lambda_{10} \ln(0.4) \leq \ln(0.3), \]

\[ \lambda_1 \ln(0.4) + \lambda_2 \ln(0.8) + \lambda_3 \ln(0.3) + \lambda_4 \ln(0.4) + \lambda_5 \ln(0.4) + \lambda_6 \ln(0.5) + \lambda_7 \ln(0.7) + \lambda_8 \ln(0.2) + \lambda_9 \ln(0.3) + \lambda_{10} \ln(0.25) \leq \ln(0.4), \]

(Using Eq. (16))

\[ \lambda_1 \ln(0.35) + \lambda_2 \ln(0.75) + \lambda_3 \ln(0.2) + \lambda_4 \ln(0.45) + \lambda_5 \ln(0.7) + \lambda_6 \ln(0.4) + \lambda_7 \ln(0.1) + \lambda_8 \ln(0.15) + \lambda_9 \ln(0.25) + \lambda_{10} \ln(0.2) \leq \ln(0.35), \]

(Using Eq. (16))
\[
\lambda_1 \ln(0.2) + \lambda_2 \ln(0.15) + \lambda_3 \ln(0.05) + \lambda_4 \ln(0.3) + \lambda_5 \ln(0.5) + \\
\lambda_6 \ln(0.25) + \lambda_7 \ln(0.05) + \lambda_8 \ln(0.2) + \lambda_9 \ln(0.1) + \lambda_{10} \ln(0.3) \leq \ln(0.2),
\]

(Using Eq. (17))
\[
\lambda_1 \ln(0.2) + \lambda_2 \ln(0.25) + \lambda_3 \ln(0.15) + \lambda_4 \ln(0.4) + \lambda_5 \ln(0.25) + \\
\lambda_6 \ln(0.35) + \lambda_7 \ln(0.2) + \lambda_8 \ln(0.3) + \lambda_9 \ln(0.25) + \lambda_{10} \ln(0.5) \leq \ln(0.2),
\]
\[
\lambda_1 \ln(0.3) + \lambda_2 \ln(0.3) + \lambda_3 \ln(0.2) + \lambda_4 \ln(0.3) + \lambda_5 \ln(0.4) + \\
\lambda_6 \ln(0.5) + \lambda_7 \ln(0.05) + \lambda_8 \ln(0.3) + \lambda_9 \ln(0.2) + \lambda_{10} \ln(0.3) \leq \ln(0.3),
\]
\[
\lambda_1 \ln(0.25) + \lambda_2 \ln(0.05) + \lambda_3 \ln(0.1) + \lambda_4 \ln(0.35) + \lambda_5 \ln(0.5) + \\
\lambda_6 \ln(0.3) + \lambda_7 \ln(0.05) + \lambda_8 \ln(0.2) + \lambda_9 \ln(0.15) + \lambda_{10} \ln(0.5) \leq \ln(0.25),
\]

(Using Eq. (18))
\[
\lambda_1 + \lambda_2 + \lambda_3 + \lambda_4 + \lambda_5 + \lambda_7 + \lambda_8 + \lambda_9 + \lambda_{10} = 1,
\]
\[
\lambda_j \geq 0, \quad j = 1, 2, \ldots, 10.
\]

**Step 3.** After computations with Lingo, we obtain \( \theta^* = 0.9068 \) for DMU1.

Similarly, for the other DMUs, we report the results in Table 3.

**Table 3.** The efficiencies of the other DMUs

<table>
<thead>
<tr>
<th>DMUs</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \theta^* )</td>
<td>0.9068</td>
<td>0.9993</td>
<td>0.5153</td>
<td>0.9973</td>
<td>0.6382</td>
<td>0.6116</td>
<td>1</td>
<td>1</td>
<td>0.6325</td>
<td>1</td>
</tr>
<tr>
<td>Rank</td>
<td>4</td>
<td>2</td>
<td>8</td>
<td>3</td>
<td>5</td>
<td>7</td>
<td>1</td>
<td>1</td>
<td>6</td>
<td>1</td>
</tr>
</tbody>
</table>

By these results, we can see that DMUs 7, 8, and 10 are efficient and others are inefficient.

6. Conclusions and future work

There are several approaches to solving various problems under neutrosophic environment. However, to the best of our knowledge, the Data Envelopment Analysis (DEA) has not been discussed with neutrosophic sets until now. This paper, therefore, plans to fill this gap and a new method has been designed to solve an input-oriented DEA model with simplified neutrosophic numbers. A numerical example has been illustrated to show the efficiency of the proposed method. The proposed approach has produced promising results from computing efficiency and performance aspects. Moreover, although the model, arithmetic operations and results presented here demonstrate the effectiveness of our approach, it could also be considered in other DEA models and their applications to banks, police stations, hospitals, tax offices, prisons, schools and universities. As future researches, we intend to study these problems.

**Acknowledgments:** The authors would like to thank the editor and anonymous reviewers to improve the quality of this manuscript.

**Conflicts of Interest:** The authors declare no conflict of interest.

**References**


Received: June 10, 2019. Accepted: October 18, 2019