A New Group Decision Making Method with Distributed Indeterminacy Form under Neutrosophic Environment: Neutrosophic Social Choice Theory

Selçuk Topal [1⋆] and Ahmet Çevik [2]
1⋆) Department of Mathematics, Bitlis Eren University, 13000, Bitlis, Turkey, s.topal@beu.edu.tr
2) Gendarmerie and Coast Guard Academy, 06805, Ankara, Turkey, a.cevik@hotmail.com

Abstract. This paper presents a novel social choice theory based multi-criteria decision method under neutrosophic environment. Our hybrid method consists of classical methods and an aggregation operator used in social choice theory. In addition to this, we also use distributed indeterminacy form function in our method to provide a more sensitive indeterminacy approach towards accuracy functions. We also consider reciprocal property for all individuals. This provides, as in intuitionistic fuzzy decision making theory, a consistent decision making for each individual. The solution approach presented in this paper in group decision making is treated under neutrosophic individual preference relations. These new approaches seem to be more consistent with natural human behaviour, hence should be more acceptable and implementable. Moreover, the use of a similar approach to develop some *deeper soft* degrees of consensus is outlined. Finally, we give a Python implementation of our work in the Appendix section.

Mathematics Subject Classification (2010). 91B06, 91B14, 91B10, 03B52.

Keywords. Neutrosophic logic, group decision making, neutrosophic preference relations, distributed indeterminacy form, social choice theory, neutrosophic social choice theory.

.

1. Introduction

In classical set theory, an element either belongs or does not belong to the set, as the membership of elements in a set is interpreted in binary terms according to a divalent case. In fuzzy set theory, introduced by Zadeh [1], the gradual assessment of the membership of elements in a set is permitted by a membership function valued in the real unit interval $[0,1]$. In fuzzy set theory, classical divalent sets are usually called crisp sets. Fuzzy set theory is a generalization of the classical set theory. Intuitionistic fuzzy sets are sets whose elements have degrees of membership and non-membership. Intuitionistic fuzzy sets have been introduced by Atanassov [2] as an extension of the notion of fuzzy set, which itself extends the classical notion of a set. Neutrosophic set theory is a generalization of the intuitionistic set, classical set, fuzzy set, paraconsistent set, dialetheist set, paradoxist set, tautological set based on Neutrosophy [3]. An element $x(T,I,F)$ belongs to the set in the following way: it is true in the set with a degree of $t \in [0,1]$, indeterminate with a degree of $i \in [0,1]$, and it is false with a degree of $f \in [0,1]$. Especially recently, neutrosophy found a significant degree of applications and attracted attention. Ye [6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19], Lui and his colleagues [20, 21, 22, 23, 25, 26, 27, 28], Biswas and his colleagues [29, 30, 31, 32, 33, 34, 35], Mondal and his colleagues [36, 38, 39, 40], Zang and his colleagues [41, 42, 43, 44, 45, 46, 47] published many papers containing significant and innovative methods on decision making under neutrosophic environment.

In our work, we determine rational social choice solely by the preferences of individuals in a society. A rational choice is possible only if every individual in the society is rational. Social choice theory investigates solutions to the problem of making a collective decision on a fair and democratic ground. The main purpose and subject area of social choice theory is to study the decision making problem for collectives to make a collective decision in a democratic manner. Of course our main concern will be to devise a method to make a cumulative decision rather than judging how fair the decisions of individuals are. The collective decision will manifest itself in neutrosophic values that the individuals assign for the preferences. Every individual is assumed to be able to assign to every preference some neutrosophic comparison value as pairs. We benefit from fuzzy and intuitionistic fuzzy social choice in solving the decision problems concerning neutrosophic social choice. Some of the most well known works in fuzzy social choice and fuzzy decision making can be found in [48, 49, 50, 51, 52, 53]. For the intuitionistic fuzzy choice, we refer the reader to [54, 55, 56]. Many of the computational social choice theories that have been studied are based on rational individuals and their consistent preferences. Knowing the fact that the consistency of these pairwise comparisons forms the main theme, such theories devise appropriate methods based on the winner of the consensus of the group or based on an ordering of preferences with respect to a priority as a result of the voting of each individual. In any social choice, the consensus winner is defined as the choice of the dominant individual or the collective decision of rational individuals. The goal is to determine the best preference picked by the group. For the fuzzy solutions of finding a consensus, we refer the reader to Kacprzyk and Nurmi [57]. We introduce a mathematical model for determining a consensus winner after a collective decision, in case there is any In case of otherwise, we present a model which orders the preferences with respect to their weights. We also give an example the last part of the paper to explain the model better.

2. Fundamental definitions

Here, we give some definitions of fundamental concepts related to our study scope such as fuzzy set, intuitionistic fuzzy set (IFS), neutrosophic set and single-valued neutrosophic set.

Definition 2.1. [1] Given a universal set U, with a generic element denoted by x, a fuzzy set X in U is a defined as a set of ordered pairs:

$X = \{(x, \mu_X(x))|x \in U\}$ where $\mu_X : U \longmapsto [0,1]$ is called the membership function of A and $\mu_X(x)$ represents the degree of membership of the element x in X.

Definition 2.2. [2] An intuitionistic fuzzy set X over a universe of discourse U is represented as:

$X = \{(x, \mu_X(x), \nu_X(x))|x \in U\}$ where $\mu_X : U \longmapsto [0,1]$ and $\nu_X : U \longmapsto [0,1]$ are called respectively the membership function of A and the non-membership function of A for x in X. The degree of non-membership of the element x in X is normally defined as $\mu_X(x) = 1 - \nu_X(x)$.

Definition 2.3. [3, 4] Let U be a universe of discourse, then a *neutrosophic set* is defined as:

$$N = \{(x, T(x), I(x), F(x)) : x \in U\}$$

which is identified by a truth-membership function $T_N : U \longmapsto]0^-, 1^+[$, indeterminacy-membership function $I_N : U \longmapsto]0^-, 1^+[$ and falsity-membership function $F_N : U \longmapsto]0^-, 1^+[$.

Definition 2.4. [3, 4] Let U be a universe of discourse, then a *single valued neutrosophic set* is defined as:

$$N = \{(x, T(x), I(x), F(x)) : x \in U\}$$

where, a truth-membership function $T_N : U \longmapsto [0,1]$, indeterminacy-membership function $I_N : U \longmapsto [0,1]$ and falsity-membership function $F_N : U \longmapsto [0,1]$ with $0 \leq T_N(x) + I_N(x) + F_N(x) \leq 3$. A single-valued neutrosophic number (SVNN) is denoted by a $a = (T, I, F)$.

Definition 2.5. [5] Let a be a single-valued neutrosophic number, an *accuracy function* H of a single-valued neutrosophic number can be represented as follows:

$$H(a) = \frac{1 + T_a - I_a(1 - T_a) - F_a(1 - I_a))}{2}$$

where for all a, $H(a) \in [0, 1]$. H is a order relation which gives an accuracy score of information of a. If $H(a_1) = H(a_2)$, then $a_1 = a_2$, that is, they have the same information. If $H(a_1) < H(a_2)$, then a_2 is larger than a_1, denoted by $a_1 < a_2$.

3. Accuracy function and distributed indeterminacy form

We would like to discuss the accuracy function H used so far in almost all papers in neutrosophic studies. For a neutrosophic value, the accuracy function H is calculated by the values T, I and F. However, in a process of making a decision, such independent values may not yield results consistent with the decision-making process on objects. Suppose, one has truth, falsity and indeterminacy applied on a concept (or idea). We can not speak about these truth values by eliminating the indeterminacy. The reason is that we make a decision on the basis of including indeterminacy and the truth-maker gives the values by taking into account the indeterminacy. Sorensen [58, 59, 60], who published some papers on truth-maker theory, buries the theory of indeterminacy in the truth-maker theory. In a similar approach, we desire to calculate the the accuracy function by considering the indeterminacy as dependent on T and F. The direct application of this idea to neutrosophic decision making helps us to approximate the outcomes with a better precision by distributing the indeterminacy on neutrosophic values. Let H be an accuracy function. This time we reflect the indeterminacy value on the truth and falsity values by the following way: let $a = (T_a, I_a, F_a)$ be a single valued neutrosophic number with truth value T_a, indeterminacy value I_a, and falsity value F_a. *Distributed Indeterminacy Form* (DIF) of a is $a_{DIF} = (T_a - T_a I_a, 0, F_a - F_a I_a)$. Here, we distribute indeterminacy effect on truth and falsity. In other words, we decrease the power of truth and falsity in proportion to the magnitude of indeterminacy. Our aim here is to determine how the value of truth and falsity is affected by the growth of indeterminacy. We give an example to an accuracy function H. Despite that $H(0.5, 0.5, 0.6) = 0.475$, we have that $H(0.5, 0.6, 0.6) = 0.48$. In other words, even though the precision should have been decreased when the indeterminacy increases, we observe here the opposite. This at first might seem contradictory but the situation will become clear in a moment. Therefore, DIF gives us a method to keep a neutrosophic number as least as possible in the ordering of the preferences in proportional to the increment of the indeterminacy value, provided that the truth or falsity values are fixed.

Self Comparison. Any comparisons on the same alternative should be assigned as an balanced value by rational individuals. The values 0.5, (0.5, 0.5), and (0.5, 0.5, 0.5) are assigned respectively for self-comparison by individuals in fuzzy set, intuitionistic fuzzy set and neutrosophic set. Assigned self comparison of a neutrosophic value a is $(0.5, 0.5, 0.5)$ and outcome of this number under H function is naturally $H(a) = 0.5$. The DIF of this value is $a_{DIF} = (0.25, 0, 0.25)$ and $H(0.25, 0, 0.25) = 0.375$. This in turn gives us a result quite different from self-comparison. One of the most important reasons that we introduce the distributed indeterminacy concept is the effect of indeterminacy over the other two values, i.e truth and falsity. Moreover, we would like to see this effect as a rational assignment in the self-comparison process, so we would like to use the triplet $(0.5, 0, 0.5)$ instead of $(0, 5, 0.5, 0.5)$. As it can be seen, we pull the indeterminacy factor down to zero. Moreover, the DIF of $(0.5, 0, 0.5)$ is equal to itself, that is $(0.5, 0, 0.5)$. Furthermore, the image of $(0.5, 0, 0.5)$ under the function H takes the value 0.5, which is just the appropriate value for the self-comparison process.

4. Reciprocal property and hesitation function

4.1. Reciprocal property in fuzzy theory

[61] A fuzzy preference relation $R = (r_{ij})$ on a finite set of alternatives X is a relation in $X \times X$ that is characterised by a membership function $\mu_R : X \times X \longmapsto [0, 1]$. Pairwise comparisons concentrate simply on two alternatives at a time which enable individuals when expressing their preferences. If an individual prefers an alternative x_i to another alternative x_j, then she/he should not simultaneously prefers x_j to x_i. Then, the comparison outcomes preference of an alternative numerical representation into a reciprocal preference relation R is:

$$\begin{aligned} r_{ij} = 1 &\quad \Leftrightarrow \quad x_i \succ x_j \\ r_{ij} = 0 &\quad \Leftrightarrow \quad x_j \succ x_i \\ r_{ij} = 0.5 &\quad \Leftrightarrow \quad x_j \sim x_i \end{aligned}$$

Binary crisp preference relations or $[0, 1]$-valued (fuzzy) preference relations are used in fuzzy social choice theory. $x_{ij} = 1$ shows the absolute degree of preference for x_i over x_j. A definite preference for x_i over x_j is $r_{ij} \in (0.5, 1)$. Indifference between x_i and x_j is $r_{ij} = 0.5$. Reciprocal $[0, 1]$-valued relations $(R = (r_{ij}; \forall i, j : 0 \leq r_{ij} \leq 1, r_{ij} + r_{ji} = 1)$ are constantly used in fuzzy set theory for representing preferences.

4.2. Reciprocal property and hesitation function in intuitionistic fuzzy theory

[62] An intuitionistic fuzzy preference relation P on a finite set of alternatives $X = \{x_1, ..., x_n\}$ is characterised by a membership function $\mu_P : X \times X \longrightarrow [0, 1]$ and a non-membership function $\nu_P : X \times X \longrightarrow$ such that $0 \leq \mu_P(x_i, x_j) + \nu_P(x_i, x_j) \leq 1, \forall (x_i, x_j) \in X \times X$. As is in a fuzzy preference relation, an intuitionistic fuzzy preference relation is represented by a matrix $P = (p_{ij})$ with $p_{ij} =< \mu_{ij}, \nu_{ij} >, \forall i, j = 1, 2, ..., n$. Obviously, when the hesitancy function is the null function we have that $\mu_{ij} + \nu_{ij} = 1$ $(\forall i, j)$ and the

intuitionistic fuzzy preference relation $P = (p_{ij})$ is mathematically equivalent to the reciprocal fuzzy preference relation $R = (r_{ij})$, with $r_{ij} = \mu_{ij}$. An intuitionistic fuzzy preference relation is referred to as reciprocal when the following additional conditions are imposed:

(i) $\mu_{ii} = \nu_{ii} = 0.5$, $\forall i \in \{1, ..., n\}$.

(ii) $\mu_{ij} = \nu_{ji}$, $\forall i, j \in \{1, ..., n\}$.

In intuitionistic fuzzy studies, the relations do not have to have reciprocity but must satisfy $r_{ij} \leq 1 - r_{ji}$ due to intuitionistic index. In other words, for an IFS A, $\pi_A(x)$ determined by the following expression: $\pi_A(x) = 1 - \mu_A(x) - \nu_A(x)$ is called the hesitancy degree of the element $x \in X$ to the set A, and $\pi_A(x) \in [0, 1]$, $\forall x \in X$.

4.3. Reciprocal property and hesitation function in neutrosophy theory

Let $S = \{s_1, s_2, s_3, ..., s_n\}$ be a set of alternatives (or options) and m be set of individuals. Each individual declares his or her own preferences over S which are represented by an individual neutrosophic preference relation R_k such that:

$$N_{R_k} : S \times S \longmapsto [0, 1] \times [0, 1] \times [0, 1]$$

which is traditionally represented by a matrix $R_k = [r^k_{ij} = N_{R_k}(r^k_i, r^k_j)], i, j = 1, 2, 3, .., n; k = 1, 2, 3, ..., m$.

$$R_k = \begin{bmatrix} (0.5, 0.5, 0.5) & r^k_{12} & r^k_{13} & r^k_{14} \\ r^k_{21} & (0.5, 0.5, 0.5) & r^k_{23} & r^k_{24} \\ r^k_{31} & r^k_{32} & (0.5, 0.5, 0.5) & r^k_{33} \\ r^k_{41} & r^k_{42} & r^k_{43} & (0.5, 0.5, 0.5) \end{bmatrix}$$

The matrix above shows that neutrosophic preferences of an individual k among s_1, s_2, s_3, s_4. $N_{R_k}(s_1, s_1) = N_{R_k}(s_2, s_2) = N_{R_k}(s_3, s_3) = N_{R_k}(s_4, s_4) = (0.5, 0.5, 0.5)$, $N_{R_k}(s_1, s_2) = r^k_{12}$, $N_{R_k}(s_3, s_4) = r^k_{34}$, ... etc. Our assumption requires that there is no largeness when an alternative compares to itself. Almost all the research on decision making do not assign any value or assign zero degree in their underlying discourse for self-comparisons. We follow a completely computational approach in this study. On the other hand, zeros given in previous studies may lead us to a false perception to compare any s_i. If a neutrosophic preference function mu, $mu(s_i, s_j) = 0$, then s_i is definitely larger than s_j. If we had a rational individual, $mu(s_i, s_i)$ would have been 0.5, because if we do self-comparison, an alternative can not have any advantage or different information over itself. We use H function in Definition 2.5 for both accuracy and to be a neutrosophic index of SVNNs. If $i = j$, then we take $N_{R_k}(s_i, s_j)$ to be $(0.5, 0.5, 0.5)$ without DIF, and $(0.5, 0, 0.5)$ with DIF. So, we have the following matrix:

$$R_k = \begin{bmatrix} (0.5, 0, 0.5) & r^k_{12} & r^k_{13} & r^k_{14} \\ r^k_{21} & (0.5, 0, 0.5) & r^k_{23} & r^k_{24} \\ r^k_{31} & r^k_{32} & (0.5, 0, 0.5) & r^k_{33} \\ r^k_{41} & r^k_{42} & r^k_{43} & (0.5, 0, 0.5) \end{bmatrix}$$

The function H (called neutrosophic index or neutrosophic hesitation function) assigns each a_{ij} neutrosophic value to a number in $[0, 1]$.

$$H(a_{ij}) = \frac{1 + T(a_{ij}) - I(a_{ij})(1 - T(a_{ij})) - F(a_{ij})(1 - I(a_{ij}))}{2}$$

Now, we have a new matrix, $R^H_k = [H(r^k_{ij}) = H^k(N_{R_k}(s_i, s_j))]$ where $i, j = 1, 2, 3, .., n$; $k = 1, 2, 3, ..., m$. With a more explicit expression,

$$R^H_k = \begin{bmatrix} H((0.5, 0, 0.5)) & H(r^k_{12}) & H(r^k_{13}) & H(r^k_{14}) \\ H(r^k_{21}) & H((0.5, 0, 0.5)) & H(r^k_{23}) & H(r^k_{24}) \\ H(r^k_{31}) & H(r^k_{32}) & H((0.5, 0, 0.5)) & H(r^k_{34}) \\ H(r^k_{41}) & H(r^k_{42}) & H(r^k_{43}) & H((0.5, 0, 0.5)) \end{bmatrix}$$

We find more appropriate to use the notion of *hesitation* in order to have consistency between the choosers (individuals) and the chooser's preference. Here, we benefit from the intuitionistic fuzzy sets (IFS). In utilizing IFS, we provide a hybrid account of the neutrosophic accuracy function by hesitation. Here, we will adapt intuitionistic index to our study since we will be using the function H as a solid index throughout the paper. Not every $H^k(r_{ij})$ need to be reciprocal, i.e. $H^k(r_{ij}) \neq 1 - H^k(r_{ji})$ but should be quasi-reciprocal. That is, $H(r^k_{ij}) \leq 1 - H(r^k_{ji})$, for each $i, j = 1, ..., n$. If k is not quasi-reciprocal, we call k is not a rational individual. That is, k is an *irrational individual.* If $i = j$, then we just take $N_{Rk}(a_i, a_j) = (0.5, 0.5, 0.5)$ because $H((0.5, 0.5, 0.5)) = 0.5$ irrespective of DIF of neutrosophic numbers. Furthermore, when we consider DIF, for information for a rational individual on the same preference is $(0.5, 0, 0.5)$ from now on, and $H((0.5, 0, 0.5)) = 0.5$

as desired.

$$DIF(R_k) = \begin{bmatrix} (0.5, 0, 0.5) & DIF(r^k_{12}) & DIF(r^k_{13}) & DIF(r^k_{14}) \\ DIF(r^k_{21}) & (0.5, 0, 0.5) & DIF(r^k_{23}) & DIF(r^k_{24}) \\ DIF(r^k_{31}) & DIF(r^k_{32}) & (0.5, 0, 0.5) & DIF(r^k_{33}) \\ DIF(r^k_{41}) & DIF(r^k_{42}) & DIF(r^k_{43}) & (0.5, 0, 0.5) \end{bmatrix}$$

R_i : preference matrix of ith individual,

$DIF(R_i)$: DIF of preference matrix of ith individual,

R_i^H : range of preference matrix of ith individual under H function,

$r_k^H(ij)$: represents the element at the row i and column j of R_i^H for individual k

$h^k(ij)$: distribution of individual k's votes for each pairwise comparison of alternative's value is determined by the size from 0.5 obtained from R_i^H,

$[[h^k]]$: the matrix obtained by each element of $h^k(ij)$,

$[[H_{ij}]]$: matrix of the group vote,

A_k : the degree for preference k assigned by the group ,

a^k_{ij} : majority determination value for preference k from the group (the element at the row i and column j of $[[h^k]]$),

H^k_{ij} : majority determination value for preference k from the group under H function,

$H_{\pi_{ij}}$: average majority determination value of the group under H function,

H_π : consensus winner determination matrix,

$C(s_i)$: social aggregation function for alternative (preference) s_i,

Practical Example 4.1. *Suppose that there are three experts* m_1, m_2, m_3 *and four facilities* s_1, s_2, s_3, s_4 *in the same business industry. We assume that all experts are rational and so we consider all neutrosophic values satisfy quasi-reciprocal property. We also assume the self-comparison value to be* $(0.5, 0, 0.5)$. *Each expert assigns his/her neutrosophic opinion value by comparing the facilities in pairs as follows:*

R_{m_i} *is the set of assigned values (preferences of) by* m_i *to pairs in the facilities where* $1 \leq i \leq 3$.

$R_{m_1} = \{(s_1, s_1) = (0.5, 0, 0.5), (s_1, s_2) = (0.45, 0.24, 0.27), (s_1, s_3) = (0.31, 0.14, 0.66), (s_1, s_4) = (0.8, 0.3, 0), (s_2, s_1) = (0.1, 0.45, 0.52), (s_2, s_2) = (0.5, 0, 0.5), (s_2, s_3) = (0.48, 0.26, 0.37), (s_2, s_4) = (0.2, 0.7, 0.8), (s_3, s_1) = (0.61, 0.43, 0.71), (s_3, s_2) = (0.31, 0, 0.71), (s_3, s_3) = (0.5, 0, 0.5), (s_3, s_4) = (0.76, 0.23, 0.27), (s_4, s_1) = (0.1, 0.6, 0.9), (s_4, s_2) = (0.81, 0.55, 0.33), (s_4, s_3) = (0.11, 0.32, 0.59), (s_4, s_4) = (0.5, 0, 0.5)\}$

$R_{m_2} = \{(s_1, s_1) = (0.5, 0, 0.5), (s_1, s_2) = (0.2, 0.4, 0.7), (s_1, s_3) = (0.21, 0.55, 0.95), (s_1, s_4) = (0.4, 0.5, 0.3), (s_2, s_1) = (0.29, 0.53, 0.38), (0.29, 0.53, 0.38), (s_2, s_2) = (0.5, 0, 0.5), (s_2, s_3) = (0.62, 0.45, 0.16), (s_2, s_4) = (0.2, 0.7, 0.8), (s_3, s_1) = (0.72, 0.15, 0.18) (0.11, 0.13, 0.79), (s_3, s_3) = (0.5, 0, 0.5), (s_3, s_4) = (0.51, 0.45, 0.53), (s_4, s_1) = (0.15, 0.35, 0.23), (s_4, s_2) = (0.81, 0.55, 0.33), (s_4, s_3) = (0.17, 0.57, 0.36), (s_4, s_4) = (0.5, 0, 0.5)\}$

$R_{m_3} = \{(s_1, s_1) = (0.5, 0, 0.5), (s_1, s_2) = (0.3, 0.45, 0.7), (s_1, s_3) = (0.1, 0.85, 0.78), (s_1, s_4) = (0.4, 0.5, 0.3), (s_2, s_1) = (0.36, 0.51, 0.39), (s_2, s_2) = (0.5, 0, 0.5), (s_2, s_3) = (0.62, 0.45, 0.16), (s_2, s_4) = (0.1, 0.8, 0.21), (s_3, s_1) = (0.92, 0.1, 0.16), (s_3, s_2) = (0.11, 0.13, 0.79), (s_3, s_3) = (0.5, 0, 0.5), (s_3, s_4) = (0.23, 0.45, 0.74), (s_4, s_1) = (0.15, 0.35, 0.23), (s_4, s_2) = (0.6, 0.2, 0.1), (s_4, s_3) = (0.57, 0.57, 0.36), (s_4, s_4) = (0.5, 0, 0.5)\}$

$R_{m_4} = \{(s_1, s_1) = (0.5, 0, 0.5), (s_1, s_2) = (0.2, 0.4, 0.7), (s_1, s_3) = (0.25, 0.87, 0.38), (s_1, s_4) = (0.4, 0.5, 0.3), (s_2, s_1) = (0.29, 0.53, 0.38), (s_2, s_2) = (0.5, 0, 0.5), (s_2, s_3) = (0.62, 0.45, 0.16), (s_2, s_4) = (0.34, 0.66, 0.21), (s_3, s_1) = (0.73, 0.87, 0.56), (s_3, s_2) = (0.14, 0.19, 0.79), (s_3, s_3) = (0.5, 0, 0.5), (s_3, s_4) = (0.21, 0.45, 0.66), (s_4, s_1) = (0.16, 0.35, 0.23), (s_4, s_2) = (0.6, 0.4, 0.8), (s_4, s_3) = (0.68, 0.57, 0.36), (s_4, s_4) = (0.5, 0, 0.5)\}$

We now represent each R_{m_i} *in matrix form and then calculate their distributed indeterminacy forms* $DIF(R_{m_i})$.

$$R_{m_1} = \begin{bmatrix} (0.5, 0, 0.5) & (0.45, 0.24, 0.27) & (0.31, 0.14, 0.66) & (0.8, 0.3, 0) \\ (0.1, 0.45, 0.52) & (0.5, 0, 0.5) & (0.48, 0.26, 0.37) & (0.2, 0.7, 0.8) \\ (0.61, 0.43, 0.71) & (0.31, 0, 0.71) & (0.5, 0, 0.5) & (0.76, 0.23, 0.27) \\ (0.1, 0.6, 0.9) & (0.81, 0.55, 0.33) & (0.11, 0.32, 0.59) & (0.5, 0, 0.5) \end{bmatrix}$$

$$DIF(R_{m_1}) = \begin{bmatrix} (0.5,0.5,0.5) & (0.342,0,0.2052) & (0.2666,0,0.5676) & (0.56,0,0) \\ (0.055,0,0.286) & (0.5,0,0.5) & (0.3552,0,0.2738) & (0.06,0,0.24) \\ (0.3477,0,0.4047) & (0.31,0,0.71) & (0.5,0,0.5) & (0.5852,0,0.2079) \\ (0.04,0,0.36) & (0.3645,0,0.1485) & (0.0748,0,0.4012) & (0.5,0,0.5) \end{bmatrix}$$

$$R_{m_2} = \begin{bmatrix} (0.5,0,0.5) & (0.2,0.4,0.7) & (0.21,0.55,0.95) & (0.4,0.5,0.3) \\ (0.29,0.53,0.38) & (0.5,0,0.5) & (0.62,0.45,0.16) & (0.2,0.7,0.8) \\ (0.72,0.15,0.18) & (0.11,0.13,0.79) & (0.5,0,0.5) & (0.51,0.45,0.53) \\ (0.15,0.35,0.23) & (0.81,0.55,0.33) & (0.17,0.57,0.36) & (0.5,0,0.5) \end{bmatrix}$$

$$DIF(R_{m_2}) = \begin{bmatrix} (0.5,0.5,0.5) & (0.12,0,0.42) & (0.0945,0,0.4275) & (0.2,0,0.15) \\ (0.1363,0,0.1786) & (0.5,0,0.5) & (0.341,0,0.088) & (0.06,0,0.24) \\ (0.612,0,0.153) & (0.0957,0,0.6873) & (0.5,0,0.5) & (0.2805,0,0.2915) \\ (0.0975,0,0.1495) & (0.3645,0,0.1485) & (0.0731,0,0.1548) & (0.5,0,0.5) \end{bmatrix}$$

$$R_{m_3} = \begin{bmatrix} (0.5,0,0.5),(0.3,0.45,0.7),(0.76,0.35,0.38),(0.4,0.5,0.3) \\ (0.36,0.51,0.39),(0.5,0,0.5),(0.62,0.45,0.16),(0.46,0.46,0.21) \\ (0.92,0.86,0.35),(0.11,0.13,0.79),(0.5,0,0.5),(0.23,0.45,0.74) \\ (0.15,0.35,0.23),(0.6,0.4,0.8),(0.57,0.57,0.36),(0.5,0,0.5) \end{bmatrix}$$

$$DIF(R_{m_3}) = \begin{bmatrix} (0.5,0,0.5) & (0.165,0,0.385) & (0.494,0,0.247) & (0.2,0,0.15) \\ (0.1764,0,0.1911) & (0.5,0,0.5) & (0.341,0,0.088) & (0.2484,0,0.1134) \\ (0.1288,0,0.049) & (0.0957,0,0.6873) & (0.5,0,0.5) & (0.1265,0,0.407) \\ (0.0975,0,0.1495) & (0.36,0,0.48) & (0.2451,0,0.1548) & (0.5,0,0.5) \end{bmatrix}$$

$$R_{m_4} = \begin{bmatrix} (0.5,0,0.5),(0.2,0.4,0.7),(0.51,0.35,0.38),(0.4,0.5,0.3) \\ (0.29,0.53,0.38),(0.5,0,0.5),(0.62,0.45,0.16),(0.34,0.66,0.21) \\ (0.73,0.87,0.56),(0.14,0.19,0.79),(0.5,0,0.5),(0.21,0.45,0.66) \\ (0.16,0.35,0.23),(0.6,0.4,0.8),(0.68,0.57,0.36),(0.5,0,0.5) \end{bmatrix}$$

$$DIF(R_{m_4}) = \begin{bmatrix} (0.5,0,0.5) & (0.12,0,0.42) & (0.3315,0,0.247) & (0.2,0,0.15) \\ (0.1363,0,0.1786) & (0.5,0,0.5) & (0.341,0,0.088) & (0.1156,0,0.0714) \\ (0.0949,0,0.0728) & (0.1134,0,0.6399) & (0.5,0,0.5) & (0.1155,0,0.363) \\ (0.104,0,0.1495) & (0.36,0,0.48) & (0.2924,0,0.1548) & (0.5,0,0.5) \end{bmatrix}$$

Now we apply H function to $DIF(R_i)$ and then obtain R_i^H.

$$R_{m_1}^H = \begin{bmatrix} 0.5 & 0.5684 & 0.3495 & 0.78 \\ 0.3844 & 0.5 & 0.5407 & 0.41 \\ 0.4715 & 0.3 & 0.5 & 0.6886 \\ 0.34 & 0.608 & 0.3368 & 0.5 \end{bmatrix}$$

$$h^{m_1}(ij) = \begin{cases} 1, & r_{m_1}^H(ij) > 0.5 \\ 0, & otherwise \end{cases}$$

$$[[h^{m_1}]] = \begin{bmatrix} 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

$$R^H_{m_2} = \begin{bmatrix} 0.5 & 0.35 & 0.3335 & 0.525 \\ 0.4788 & 0.5 & 0.6265 & 0.41 \\ 0.7295 & 0.2041 & 0.5 & 0.4945 \\ 0.474 & 0.474 & 0.4591 & 0.5 \end{bmatrix}$$

$$h^{m_2}(ij) = \begin{cases} 1, & r^H_{m_2}(ij) > 0.5 \\ 0, & otherwise \end{cases}$$

$$[[h^{m_2}]] = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$R^H_{m_3} = \begin{bmatrix} 0.5 & 0.39 & 0.6234 & 0.525 \\ 0.4926 & 0.5 & 0.6265 & 0.5675 \\ 0.5399 & 0.2041 & 0.5 & 0.35975 \\ 0.474 & 0.4399 & 0.54515 & 0.5 \end{bmatrix}$$

$$h^{m_3}(ij) = \begin{cases} 1, & r^H_{m_3}(ij) > 0.5 \\ 0, & otherwise \end{cases}$$

$$[[h^{m_3}]] = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

$$R^H_{m_4} = \begin{bmatrix} 0.5 & 0.35 & 0.5422 & 0.525 \\ 0.4788 & 0.5 & 0.6265 & 0.5221 \\ 0.511 & 0.2367 & 0.5 & 0.3762 \\ 0.477 & 0.439 & 0.5688 & 0.5 \end{bmatrix}$$

$$h^{m_4}(ij) = \begin{cases} 1, & r^H_{m_4}(ij) > 0.5 \\ 0, & otherwise \end{cases}$$

$$[[h^{m_4}]] = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

The next step is to collect and compare the preferences. To do this, we sum the columns of $[[H_{ij}]]$ and divide it to number of the alternatives.

$$A_k = \frac{1}{m}\sum[[H_{ik}]]$$

where $1 \leq k \leq m$

$$H_{\pi_{ij}} = \begin{cases} \frac{1}{m}\sum_{k=1}^{m} a_{ij}^{k}, & i \neq j \\ 0, & i = j \end{cases}$$

where $i, j = 1, 2, ...n$ and $k = 1, 2, ..., m$.

$$H_{\pi_{12}} = \frac{a_{12}^{m_1} + a_{12}^{m_2} + a_{12}^{m_3} + a_{12}^{m_4}}{4} = \frac{1+0+0+0}{4} = \frac{1}{4},$$

$H_{\pi_{13}} = \frac{1}{2}$, $H_{\pi_{14}} = 1$, $H_{\pi_{21}} = 0$, $H_{\pi_{23}} = 1$, $H_{\pi_{24}} = \frac{1}{2}$, $H_{\pi_{31}} = \frac{3}{4}$, $H_{\pi_{32}} = 0$, $H_{\pi_{34}} = \frac{1}{2}$, $H_{\pi_{41}} = 0$, $H_{\pi_{42}} = \frac{1}{4}$, $H_{\pi_{43}} = \frac{1}{2}$

$$H_{\pi} = \begin{bmatrix} - & \frac{1}{4} & \frac{1}{2} & 1 \\ 0 & - & 1 & \frac{1}{2} \\ \frac{3}{4} & 0 & - & \frac{1}{2} \\ 0 & \frac{1}{4} & \frac{1}{2} & - \end{bmatrix}$$

We now define the notion of a consensus winner.

Definition 4.2. [63] $s_i \in W$ is called a *consensus winner* if and only if $\forall s_j \neq s_i : r_{ij} > 0.5$, where $r_{ij} \in H_{\pi}$.

In our example above, there is no winner. Of course, it is easy to define that $\alpha-$consensus winner and others. So, we propose a *social aggregation average function* to calculate the order of s_i in the group is the extent to which individuals is not against option s_i.

$C(s_i) = \dfrac{1}{m-1} \sum_{i \neq j} r_{ij}$, where $i, j = 1, 2, ..., m$.

$C(s_1) = \dfrac{3}{16}$, $C(s_2) = \dfrac{1}{16}$, $C(s_3) = \dfrac{9}{16}$, $C(s_4) = \dfrac{6}{16}$.

Then, $C(s_3) > C(s_4) > C(s_1) > C(s_2)$.

5. Conclusion

The main aim of this paper is to bring into attention the interplay between neutrosophy and social choice theory. Within the framework of this intention, we have taken inheritance from the studies of fuzzy and intuitionistic fuzzy social choice theory and developed *the neutrosophic based social choice theory*. First we defined the DIF, which was used in Sorensen's truth-maker theory to distribute the indeterminacy on truth and falsity for neutrosophic calculations. We believe that the notion of DIF will give a new breath and different perspectives for neutrosophic studies. Through DIF, we emphasize both hesitation and reciprocal characteristics in self-comparisons and other pairwise comparisons to make a consistent decision maker. We determine a consensus winner if any. In case of otherwise, we obtain orders of the given alternatives by defining a social aggregation average function. The implementation in Python, given in the Appendix, is an algorithm computing the output in the order of $\dfrac{n}{11}$ seconds, where n is the input size (the number of matrices), when executed in a mid-end computer.

References

[1] Zadeh, L. A. (1965). Fuzzy sets. Information and control, 8(3), 338-353.

[2] Atanassov, K. T. (1999). Intuitionistic fuzzy sets. In Intuitionistic fuzzy sets (pp. 1-137). Physica, Heidelberg.

[3] Smarandache, F. (1998). Neutrosophy, Neutrosophic probability, set, and logic. Amer. Res. Press, Rehoboth, pp.104-106.

[4] H.Wang, F. Smarandache, Y.Q. Zhang, R. Sunderraman, Single valued neutrosophic sets. Multispace Multistructure , 4(2010):410413.

[5] Sahin, R. (2014). Multi-criteria neutrosophic decision making method based on score and accuracy functions under neutrosophic environment. arXiv preprint arXiv:1412.5202.

[6] Ye, J. (2013). Another form of correlation coefficient between single valued neutrosophic sets and its multiple attribute decision-making method. Neutrosophic Sets and systems, 1(1), 8-12.

[7] Ye, J. (2013). Multicriteria decision-making method using the correlation coefficient under single-valued neutrosophic environment. International Journal of General Systems, 42(4), 386-394.

[8] Ye, J. (2014). Some aggregation operators of interval neutrosophic linguistic numbers for multiple attribute decision making. Journal of Intelligent & Fuzzy Systems, 27(5), 2231-2241.

[9] Ye, J. (2014). A multicriteria decision-making method using aggregation operators for simplified neutrosophic sets. Journal of Intelligent & Fuzzy Systems, 26(5), 2459-2466.

[10] Ye, J. (2014). Similarity measures between interval neutrosophic sets and their applications in multicriteria decision-making. Journal of Intelligent & Fuzzy Systems, 26(1), 165-172.

[11] Ye, J. (2014). Single valued neutrosophic cross-entropy for multicriteria decision making problems. Applied Mathematical Modelling, 38(3), 1170-1175.

[12] Ye, J. (2014). Multiple attribute group decision-making method with completely unknown weights based on similarity measures under single valued neutrosophic environment. Journal of Intelligent & Fuzzy Systems, 27(6), 2927-2935.

[13] Ye, J. (2014). Vector similarity measures of simplified neutrosophic sets and their application in multicriteria decision making. Infinite Study.

[14] Ye, J. (2014). Improved correlation coefficients of single valued neutrosophic sets and interval neutrosophic sets for multiple attribute decision making. Journal of Intelligent & Fuzzy Systems, 27(5), 2453-2462.

[15] Ye, J. (2015). Trapezoidal neutrosophic set and its application to multiple attribute decision-making. Neural Computing and Applications, 26(5), 1157-1166.

[16] Ye, J. (2015). An extended TOPSIS method for multiple attribute group decision making based on single valued neutrosophic linguistic numbers. Journal of Intelligent & Fuzzy Systems, 28(1), 247-255.

[17] Ye, J. (2015). Multiple-attribute decision-making method under a single-valued neutrosophic hesitant fuzzy environment. Journal of Intelligent Systems, 24(1), 23-36.

[18] Ye, J. (2017). Projection and bidirectional projection measures of single-valued neutrosophic sets and their decision-making method for mechanical design schemes. Journal of Experimental & theoretical artificial intelligence, 29(4), 731-740.

[19] Ye, J. (2017). Bidirectional projection method for multiple attribute group decision making with neutrosophic numbers. Neural Computing and Applications, 28(5), 1021-1029.

[20] Liu, P., & Wang, Y. (2014). Multiple attribute decision-making method based on single-valued neutrosophic normalized weighted Bonferroni mean. Neural Computing and Applications, 25(7-8), 2001-2010.

[21] Liu, P., Chu, Y., Li, Y., & Chen, Y. (2014). Some Generalized Neutrosophic Number Hamacher Aggregation Operators and Their Application to Group Decision Making. International Journal of fuzzy systems, 16(2).

[22] Liu, P., & Li, H. (2017). Multiple attribute decision-making method based on some normal neutrosophic Bonferroni mean operators. Neural Computing and Applications, 28(1), 179-194.

[23] Liu, P., & Shi, L. (2017). Some neutrosophic uncertain linguistic number Heronian mean operators and their application to multi-attribute group decision making. Neural Computing and Applications, 28(5), 1079-1093.

[24] Liu, P., & Shi, L. (2015). The generalized hybrid weighted average operator based on interval neutrosophic hesitant set and its application to multiple attribute decision making. Neural Computing and Applications, 26(2), 457-471.

[25] Liu, P., & Tang, G. (2016). Some power generalized aggregation operators based on the interval neutrosophic sets and their application to decision making. Journal of Intelligent & Fuzzy Systems, 30(5), 2517-2528.

[26] Sahin, R., & Liu, P. (2016). Maximizing deviation method for neutrosophic multiple attribute decision making with incomplete weight information. Neural Computing and Applications, 27(7), 2017-2029.

[27] Chi, P., & Liu, P. (2013). An extended TOPSIS method for the multiple attribute decision making problems based on interval neutrosophic set. Neutrosophic Sets and Systems, 1(1), 63-70.

[28] Sahin, R., Liu P.(2016). Maximizing deviation method for neutrosophic multiple attribute decision making with incomplete weight information. Neural Computing and Applications 27(7) 2017-2029.

[29] Biswas, P., Pramanik, S., & Giri, B. C. (2014). A new methodology for neutrosophic multi-attribute decision making with unknown weight information. Neutrosophic Sets and Systems, 3, 42-52.

[30] Biswas, P., Pramanik, S., & Giri, B. C. (2014). Entropy based grey relational analysis method for multi-attribute decision making under single valued neutrosophic assessments. Neutrosophic Sets and Systems, 2, 102-110.

[31] Biswas, P., Pramanik, S., & Giri, B. C. (2015). Cosine similarity measure based multi-attribute decision-making with trapezoidal fuzzy neutrosophic numbers. Neutrosophic sets and systems, 8(unknown), 46-56.

[32] Biswas, P., Pramanik, S., & Giri, B. C. (2016). Value and ambiguity index based ranking method of single-valued trapezoidal neutrosophic numbers and its application to multi-attribute decision making. Neutrosophic Sets and Systems, 12, 127-138.

[33] Biswas, P., Pramanik, S., & Giri, B. C. (2016). Aggregation of triangular fuzzy neutrosophic set information and its application to multi-attribute decision making. Neutrosophic sets and systems, 12, 20-40.

[34] Biswas, P., Pramanik, S., & Giri, B. C. (2016). TOPSIS method for multi-attribute group decision-making under single-valued neutrosophic environment. Neural computing and Applications, 27(3), 727-737.

[35] Pramanik, S., Biswas, P., & Giri, B. C. (2017). Hybrid vector similarity measures and their applications to multi-attribute decision making under neutrosophic environment. Neural computing and Applications, 28(5), 1163-1176.

[36] Mondal, K., & Pramanik, S. (2014). Multi-criteria group decision making approach for teacher recruitment in higher education under simplified neutrosophic environment. Neutrosophic Sets and Systems, 6, 28-34.

[37] Mondal, K., & Pramanik, S. (2015). Neutrosophic tangent similarity measure and its application to multiple attribute decision making. Neutrosophic sets and systems, 9, 80-87.

[38] Mondal, K., & Pramanik, S. (2015). Neutrosophic decision making model of school choice. Neutrosophic Sets and Systems, 7, 62-68.

[39] Mondal, K., & Pramanik, S. (2015). Rough neutrosophic multi-attribute decision-making based on grey relational analysis. Neutrosophic Sets and Systems, 7, 8-17.

[40] Mondal, K., & Pramanik, S. (2015). Neutrosophic refined similarity measure based on cotangent function and its application to multi-attribute decision making. Infinite Study.

[41] Peng, J. J., Wang, J. Q., Zhang, H. Y., & Chen, X. H. (2014). An outranking approach for multi-criteria decision-making problems with simplified neutrosophic sets. Applied Soft Computing, 25, 336-346.

[42] Zhang, H. Y., Wang, J. Q., & Chen, X. H. (2014). Interval neutrosophic sets and their application in multicriteria decision making problems. The Scientific World Journal, 2014.

[43] Zhang, H. Y., Ji, P., Wang, J. Q., & Chen, X. H. (2015). An improved weighted correlation coefficient based on integrated weight for interval neutrosophic sets and its application in multi-criteria decision-making problems. International Journal of Computational Intelligence Systems, 8(6), 1027-1043.

[44] Peng, J. J., Wang, J. Q., Wang, J., Zhang, H. Y., & Chen, X. H. (2016). Simplified neutrosophic sets and their applications in multi-criteria group decision-making problems. International journal of systems science, 47(10), 2342-2358.

[45] Zhang, H., Wang, J., & Chen, X. (2016). An outranking approach for multi-criteria decision-making problems with interval-valued neutrosophic sets. Neural Computing and Applications, 27(3), 615-627.

[46] Zhang, H. Y., Ji, P., Wang, J. Q., & Chen, X. H. (2016). A neutrosophic normal cloud and its application in decision-making. Cognitive Computation, 8(4), 649-669.

[47] Tian, Z. P., Zhang, H. Y., Wang, J., Wang, J. Q., & Chen, X. H. (2016). Multi-criteria decision-making method based on a cross-entropy with interval neutrosophic sets. International Journal of Systems Science, 47(15), 3598-3608.

[48] Blin, J. M., & Whinston, A. B. (1973). Fuzzy sets and social choice. Journal of Cybernetics Volume 3, Issue 4.

[49] Dubois, D. & Koning, J. L. (1991). Social choice axioms for fuzzy set aggregation. Fuzzy sets and systems, 43(3), 257-274.

[50] Dutta, B. (1987). Fuzzy preferences and social choice. Mathematical Social Sciences, 13(3), 215-229.

[51] Richardson, G. (1998). The structure of fuzzy preferences: social choice implications. Social Choice and Welfare, 15(3), 359-369.

[52] Nurmi, H. (1981). Approaches to collective decision making with fuzzy preference relations. Fuzzy Sets and systems, 6(3), 249-259.

[53] Richard, B., & Maurice, S. (2011). Social choice with fuzzy preferences. In Handbook of social choice and welfare (Vol. 2, pp. 367-389). Elsevier.

[54] Szmidt, E.& Kacprzyk, J. (1996). Intuitionistic fuzzy sets in group decision making. Notes on IFS, 2(1).

[55] Yue, Z. (2011). Deriving decision makers weights based on distance measure for interval-valued intuitionistic fuzzy group decision making. Expert Systems with Applications, 38(9), 11665-11670.

[56] Yue, Z. & Jia, Y. (2013). An application of soft computing technique in group decision making under interval-valued intuitionistic fuzzy environment. Applied Soft Computing, 13(5), 2490-2503.

[57] Kacprzyk, J., Nurmi, H., & Fedrizzi, M. (Eds.). (2012). Consensus under fuzziness (Vol. 10). Springer Science & Business Media.

[58] Sorensen, R. (2001). Vagueness and Contradiction, Oxford: Oxford University Press.

[59] Sorensen, R. (2005). Prĕcis of Vagueness and Contradiction, Philosophy and Phenomenological Research, 71, pp. 678-685.

[60] Sorensen, R. (2005). A Reply to Critics, Philosophy and Phenomenological Research, 71, pp. 712-728.

[61] Dasgupta, M. and R. Deb (1991) Fuzzy choice functions, Social Choice and Welfare, 8: 171- 182.

[62] Szmidt, E., & Kacprzyk, J. (1996). Intuitionistic fuzzy sets in group decision making. Notes on IFS, 2(1).

[63] Nurmi, H. (1981). Approaches to collective decision making with fuzzy preference relations, Fuzzy Sets and Systems. 6 249-259.

[64] Python Software Foundation. Python Language Reference, version 2.7. Available at http://www.python.org

[65] G. van Rossum, Python tutorial, Technical Report CS-R9526, Centrum voor Wiskunde en Informatica (CWI), Amsterdam, May 1995.

6. Appendix

A Python implementation [64, 65] of the group decision making method with distributed indeterminacy form under neutrosophic environment is as follows:

```
from __future__ import division
from collections import defaultdict
import math
import sys


R1=[ [ (0.5,0,0.5),(0.45,0.24,0.27), (0.31,0.14,0.66) , (0.8,0.3,0)],
     [(0.1,0.45,0.52) , (0.5,0,0.5), (0.48,0.26,0.37) , (0.2,0.7,0.8)],
     [(0.61,0.43,0.71) , (0.31,0,0.71) , (0.5,0,0.5)  , (0.76,0.23,0.27)],
     [(0.1,0.6,0.9) , (0.81,0.55,0.33) , (0.11,0.32,0.59) , (0.5,0,0.5)]]


R2=[ [ (0.5,0,0.5),(0.2,0.4,0.7), (0.21,0.55,0.95) , (0.4,0.5,0.3)],
     [(0.29,0.53,0.38) , (0.5,0,0.5), (0.62,0.45,0.16) , (0.2,0.7,0.8)],
     [(0.72,0.15,0.18) , (0.11,0.13,0.79)  , (0.5,0,0.5)  , (0.51,0.45,0.53)],
     [(0.15,0.35,0.23) , (0.81,0.55,0.33) , (0.17,0.57,0.36) , (0.5,0,0.5)] ]


R3=[ [ (0.5,0,0.5),(0.3,0.45,0.7), (0.1,0.85,0.78) , (0.4,0.5,0.3)],
     [(0.36,0.51,0.39) , (0.5,0,0.5), (0.62,0.45,0.16) , (0.1,0.8,0.21)],
     [(0.92,0.1,0.16) , (0.11,0.13,0.79)  , (0.5,0,0.5)  , (0.23,0.45,0.74)],
     [(0.15,0.35,0.23) , (0.6,0.2,0.1) , (0.57,0.57,0.36) , (0.5,0,0.5)] ]


R4=[ [ (0.5,0,0.5),(0.2,0.4,0.7), (0.25,0.87,0.38) , (0.4,0.5,0.3)],
     [(0.29,0.53,0.38) , (0.5,0,0.5), (0.62,0.45,0.16) , (0.34,0.66,0.21)],
     [(0.73,0.87,0.56) , (0.14,0.19,0.79)  , (0.5,0,0.5)  , (0.21,0.45,0.66)],
     [(0.16,0.35,0.23) , (0.6,0.4,0.8) , (0.68,0.57,0.36) , (0.5,0,0.5)] ]


AllTogether= {'R1': [[ (0.5,0,0.5),(0.45,0.24,0.27), (0.31,0.14,0.66) , (0.8,0.3,0)],
                     [(0.1,0.45,0.52) , (0.5,0,0.5), (0.48,0.26,0.37), (0.2,0.7,0.8)],
     [(0.61,0.43,0.71) , (0.31,0,0.71) , (0.5,0,0.5)  , (0.76,0.23,0.27)],
                     [(0.1,0.6,0.9) , (0.81,0.55,0.33) , (0.21,0.32,0.59) , (0.5,0,0.5)]],
    'R2': [[ (0.5,0,0.5),(0.2,0.4,0.7), (0.21,0.55,0.95) , (0.4,0.5,0.3)],
           [(0.29,0.53,0.38) , (0.5,0,0.5), (0.62,0.45,0.16) , (0.83,0.46,0.21)],
    [(0.72,0.15,0.18) , (0.11,0.13,0.79)  , (0.5,0,0.5)  , (0.51,0.45,0.53)],
           [(0.15,0.35,0.23) , (0.6,0.4,0.8) , (0.47,0.57,0.36) , (0.5,0,0.5)]],
    'R3': [[ (0.5,0,0.5),(0.3,0.45,0.7), (0.76,0.35,0.38) , (0.4,0.5,0.3)],
           [(0.36,0.51,0.39) , (0.5,0,0.5), (0.62,0.45,0.16) , (0.46,0.46,0.21)],
           [(0.92,0.86,0.35) , (0.11,0.13,0.79)  , (0.5,0,0.5)  , (0.23,0.45,0.74)],
           [(0.15,0.35,0.23) , (0.6,0.4,0.8) , (0.57,0.57,0.36) , (0.5,0,0.5)]],
    'R4': [[ (0.5,0,0.5),(0.2,0.4,0.7), (0.51,0.35,0.38) , (0.4,0.5,0.3)],
           [(0.29,0.53,0.38) , (0.5,0,0.5), (0.62,0.45,0.16) , (0.34,0.66,0.21)],
           [(0.73,0.87,0.56) , (0.14,0.19,0.79)  , (0.5,0,0.5)  , (0.21,0.45,0.66)],
           [(0.16,0.35,0.23) , (0.6,0.4,0.8) , (0.68,0.57,0.36) , (0.5,0,0.5)]]}


def AccuracyFunction(T,I,F):

    HV= (1+ T - I*(1-T)-F*(1-I))/2
    return HV



def DIF(T,I,F):
    T1=math.fabs(T-I*T)
    F1=math.fabs(F-I*F)

    DIFi='('+str(T1)+','+str(0)+','+str(F1)+')'
    return DIFi




def AccuracyIntedeteminacyDistubition(T,I,F):
    T1=math.fabs(T-I*T)
    F1=math.fabs(F-I*F)

    ID=AccuracyFunction(T1,I,F1)
    return ID
```

```
def RationalityChecker(R):
    columnR=len(R)
    idn=0

    rowR=len(R[0])
    for i in range(0,rowR-1):
        if R[i][i] != (0.5, 0, 0.5):
            print '(,',i,i,") is not (0.5, 0, 0.5), so, s ",i, ' is not rational agent'
            idn=1



    for i in range(0,rowR):
        for j in range(0,rowR):
            if i !=j:
                t1=R[i][j][0]
                i1=R[i][j][1]
                f1=R[i][j][2]
                A1=AccuracyIntedeteminacyDistubition(t1,i1,f1)

                t2=R[j][i][0]
                i2=R[j][i][1]
                f2=R[j][i][2]

                A2=AccuracyIntedeteminacyDistubition(t2,i2,f2)

                if A1 > 1-A2 : # A1 must be less than or equal to 1-A2
                    idn=1

                    print R[i][j], ' and ', R[j][i], ' does not satisfy hesitation property'

    return idn


def RHcreation(K):

    global RHtogether
    RHtogether= defaultdict()
    for i in K.keys():

            columnAll=len(K[i])

            rowAll1=len(K[i][0])
            rowAll2=len(K[i][0])

            for j in range(0,rowAll1):


                for k in range(0,rowAll2):

                    t1=K[i][j][k][0]
                    i1=K[i][j][k][1]
                    f1=K[i][j][k][2]
                    A=AccuracyIntedeteminacyDistubition(t1,i1,f1)



                    if i not in  RHtogether.keys():
                        RHtogether[i]=[A]

                    else:
                        RHtogether[i].extend([A])

            number= int(math.sqrt(len(RHtogether[i])))
            m=0
            new_list=[]
            while m<len(RHtogether[i]):
                new_list.append( RHtogether[i][m:m + number])
                m+= number

            RHtogether[i]=new_list


    return RHtogether
```

```
def OneZero(K):
    global H
    H=defaultdict()
    for i in K.keys():
            columnAllin=len(K[i])

            rowAl1=len(K[i][0])

            for j in range(0,columnAllin):

                for k in range(0,rowAl1):
                    if K[i][j][k]>0.5:
                        if i not in H:
                            H[i]=[1]
                        else:
                            H[i].append(1)
                    else:
                        if i not in H:
                            H[i]=[0]
                        else:
                            H[i].append(0)
            number= int(math.sqrt(len(H[i])))
            m=0
            new_list=[]
            while m<len(H[i]):
                new_list.append( H[i][m:m + number])
                m+= number

            H[i]=new_list

    return H

def H_pi_ij(K):
    global Hpij
    Hpij= defaultdict()
    columnAllin112=len(H)


    for i in range(0,columnAllin112):



        Topij=0

        for j in range(0,columnAllin112):
                Topij=0

                for k in H.keys():

                    if i != j:

                        Topij = Topij + H[k][i][j]

                    else:
                        Topij=0

                aij=str(i+1)+str(j+1)
                TopijAvarage= Topij/len(H)

                if aij not in Hpij.keys():
                    TopijAvarage= Topij/len(H)
                    Hpij[aij]=TopijAvarage
                else:
                    Hpij[aij]=TopijAvarage
    return Hpij


def Alternative_Ordinary(Hpij):
    global ORD
    ORD= defaultdict()

    Number_Of_Alternatives=int(math.sqrt(len(Hpij)))
    for i in range(1,Number_Of_Alternatives+1):
        istr=str(i)

        Top=0
        for k in Hpij.keys():
```

```
            if istr==k[1]:

                Top=Top+Hpij[k]

        TopJavarage= Top/Number_Of_Alternatives



        if istr not in ORD.keys():
            istA='Alternative '+istr
            ORD[istA]=TopJavarage

        else:
            ORD[istA]=TopJavarage

    return ORD





def GroupDecisionWithID(m):
    for i in AllTogether.keys():
        if RationalityChecker(AllTogether[i])==1:
            print 'inconsistent agent'



    Step1=RHcreation(m)
    Step2=OneZero(Step1)
    Step3=H_pi_ij(Step2)
    Step4=Alternative_Ordinary(Step3)

    return Step4
```

Selçuk Topal [1⋆] and Ahmet Çevik [2]
1⋆) Department of Mathematics, Bitlis Eren University, 13000, Bitlis, Turkey, s.topal@beu.edu.tr
2) Gendarmerie and Coast Guard Academy, 06805, Ankara, Turkey, a.cevik@hotmail.com