On Ranking Algorithms for $SVNT's$ and $NVNT's$

Murad M. Arar[1,*]
[1]Mathematics Department, Prince Sattam Bin Abdulaziz University-Al Kharj, Saudi Arabia
Emails: m.arar@psau.edu.sa

Abstract

Ranking algorithms are very important tools in decision making. There are two ranking algorithms for n-valued neutrosophic tuplets (Single-Valued MultiNeutrosophic tuplets): The S-ranking algorithm of Single-Valued MultiNeutrosophic tuplets, which is introduced by F.Smarandache in 2023, and the N-ranking algorithm of n-valued neutrosophic tuplets, Which is introduced by V. L. Nayagam and Bharanidharan R. in 2023. In this paper we show (by examples) that these two ranking algorithms are not a total ordering for the set of n-valued neutrosophic tuplets. These algorithms do not taking into account the number of sources, which is a very important factor in neutrosophic n-valued refined sets theory. We introduce two ranking algorithms: The integrated S-ranking algorithm of Single-Valued MultiNeutrosophic tuplets, and the integrated N-ranking algorithm of n-valued neutrosophic tuplets. These algorithms are improvements of the S-ranking algorithm of Single-Valued MultiNeutrosophic tuplets, and the N-ranking algorithm of n-valued neutrosophic tuplets, respectively, and taking the number of sources into account. We construct different examples to show that each step in the integrated ranking algorithms is necessary to make them a total ordering for the set of all n-valued neutrosophic tuplets.

Keywords: Single-Valued MultiNeutrosophic set; n-valued neutrosophic tuplets; S-ranking algorithm; score function; accuracy function; certainty function; membership score;non-membership score; average score

1 introduction

Logic is developed from classical logic to n-valued refined neutrosophic logic. In Boolean logic we have only two truth values: T and F. The main assumption in Boolean logic is that: Every statement is either T (true) or F (false). In the numerical two-valued logic, with every statement we have two numerical values: T and F, can be called degrees of membership and non-membership. If $T, F \in \{0, 1\}$, then we have the numerical classical (Boolean) logic. And if $T, F \in [0, 1]$ with $F = 1 - T$, then we have the fuzzy logic. Fuzzy logic introduced by L. Zadeh in 1965,[9] where with each element we have a degree of membership T such that $T \in [0, 1]$. The degree of non-membership F in fuzzy sets satisfies $T + F = 1$. Intuitionstic fuzzy sets are introduced by K. Atanassov (see[1]) in 1983 as a generalization of fuzzy sets. In intuitionistic fuzzy sets with each element we have two degrees; A degree of membership T and a degree of non-membership F such that $T, F \in [0, 1]$ and $T + F \leq 1$. Neutrosophic sets are introduced as a generalization of intuitionistic fuzzy sets by Smarandache (see[5]). In Neutrosophic sets theory with each element we have three degrees; The degree of membership (T), the degree of non-membership (F) and the degree of indeterminacy I such that $T, I, F \in [0, 1]$. In intuitionistic fuzzy sets theory we have $I = 1 - (T + F)$.

In n-valued refined logic truth value T splits into many types of truths: T_1, T_2, ..., T_p and I into many types of indeterminacies: I_1, I_2, ...,I_r and F into many types of falsities: F_1, F_2, ..., F_s, where p, r and s are integers greater than 1, and $p + r + s = n$ see.[6] Importance of n-valued refined logic and sets appeared in different applications specially in medical diagnosis see.[4]

Ordering n-valued neutrosophic tuplets is significant in multi-criteria decision making (MCDM). If we are looking for a solution of an MCDM problem, we need an ordering for the neutrosophic triplets (or n-valued neutrosophic tuplets) to help us make a decide which triplet (or n-valued neutrosophic tuplets) is doing better than the other. In fuzzy MCDM problems there are many orderings on fuzzy numbers and intuitionistic fuzzy numbers. In neutrosophic set theory there are several orderings for neutrosophic triplets and n-valued neutrosophic tuplets. In[7] F. Smarandache used the single-valued score, accuracy, and certainty functions to construct a total ordering for neutrosophic triplets. In[2] V. L. Nayagam and Bharanidharan R. used the membership, non-membership, and average score functions to construct a total ordering for the neutrosophic triplets. In 2023 different orderings for n-valued neutrosophic tuplets are introduced by V. L. Nayagam and Bharanidharan R, see.[3] In the same year, F. Smarandache introduced two methods to order Single-Valued MultiNeutrosophic sets (SVMNS's). One method for single-valued MultiNeutrosophic sets with the same (p, q, r)-forms and the other for single-valued MultiNeutrosophic sets with different (p, q, r)-forms, see.[8]

Definition 1.1. :[5] We say that A is *neutrosophic set* on X if
$A = \{\langle x, \mu_A(x), \sigma_A(x), \nu_A(x)\rangle; x \in X\}; \mu, \sigma, \nu : X \to]^-0, 1^+[$ and $^-0 \leq \mu(x) + \sigma(x) + \nu(x) \leq 3^+$.

The class of all neutrosophic sets on the universe X will be denoted by $\mathcal{N}(X)$.

In neutrosophic n-valued refined logic (see[6]) the membership degree refined (split) into r values $\mu_1, \mu_2, ..., \mu_r$, the indetermancy refined into s values $\sigma_1, \sigma_2, ..., \sigma_s$ and the nonmebership refined into t values $\nu_1, \nu_2, ..., \nu_t$ such that $n = r + s + t$ and

$$^-0 \leq \sum_{i=1}^{r} \mu_i + \sum_{i=1}^{s} \sigma_i + \sum_{i=1}^{t} \nu_i \leq n^+$$

The components ν_i, σ_j and ν_k may be partially or totally pairwise independent or dependent. This depends on applications or situations such as the same expert provided us with many components, or different experts communicated and shared information. Some authors assumes that $r = s = t$ see for example[?] and others assumes that $r = t$ to make the new logic functional and applicable.

Definition 1.2. :[6] A is called a *neutrosophic n-valued refined set* on a universe X if

$$A = \{\langle x, \mu_A^1(x), \mu_A^2(x), ..., \mu_A^r(x); \sigma_A^1(x), \sigma_A^2(x), ..., \sigma_A^s(x); \nu_A^1(x), \nu_A^2(x), ..., \nu_A^t(x)\rangle; x \in X\}$$

such that $\mu_A^i, \sigma_A^j, \nu_A^k : X \to]^-0, 1^+[$ for every $i = 1, ..., r$, $j = 1, ..., s$, $k = 1, ..., t$, $r + s + t = n$ and

$$^-0 \leq \sum_{i=1}^{r} \mu_A^i(x) + \sum_{j=1}^{s} \sigma_A^j + \sum_{k=1}^{t} \nu_A^k \leq n^+.$$

The class of all neutrosophic n-valued refined sets on the universe X will be denoted by $\mathcal{R}_n(X)$.

Definition 1.3. Let $\mathfrak{M} = \{(T, I, F); T, I, F \in [0, 1] \quad \text{and} \quad 0 \leq T, I, F \leq 1\}$ be the set of all $SVNT's$.

1. The SVNT score function $s : \mathfrak{M} \to [0, 1]$ is defined by:

$$s(T, I, F) = \frac{2 + T - I - F}{3}$$

.

2. The SVNT accuracy function $a : \mathfrak{M} \to [0, 1]$ is defined by:

$$a(T, I, F) = T - F$$

.

3. The SVNT certainty function $c : \mathfrak{M} \to [0, 1]$ is defined by:

$$c(T, I, F) = T$$

.

Definition 1.4. [2] Let $\mathfrak{M} = \{(T, I, F); T, I, F \in [0,1] \quad \text{and} \quad 0 \leq T, I, F \leq 1\}$ be the set of all single valued neutrosophic triplet (SVNT).

1. The SVNT membership score $S^+ : \mathfrak{M} \to [0,1]$ is defined by:
$$S^+(T, I, F) = \frac{2 + (T - F)(2 - I) - I}{4}$$
2. The SVNT non-membership score $S^- : \mathfrak{M} \to [0,1]$ is defined by:
$$S^-(T, I, F) = \frac{2 + (F - I)(2 - I) - I}{4}$$
3. The SVNT average score $C : \mathfrak{M} \to [0,1]$ is defined by:
$$C(T, I, F) = \frac{T + F}{2}$$

$\mathfrak{M}$ can be ordered using two ranking algorithms.

- **The S-Ranking Algorithm of SVNT's:** This algorithm is introduced by F. Smaradache in[7] using the score (s), accuracy (a),and certainty (c) functions (see Definition 1.3). It works as follows:
  Let $A = (T, I, F)$ and $B = (T', I', F')$ be two $SVMNT's$, Then (The symbol $>_S$ will be used to refer this ordering).
  1. **Step 1:** If $s(A) > s(B)$ ($s(A) < s(B)$), then $A >_S B$ ($A <_S B$). Otherwise, go to Step 2.
  2. **Step 2:** If $a(A) > a(B)$ ($a(A) < a(B)$), then $A >_S B$ ($A <_S B$). Otherwise, go to Step 3.
  3. **Step 3:** If $c(A) > c(B)$ ($c(A) < c(B)$), then $A >_S B$ ($A <_S B$). Otherwise, $A = B$.
- **The N-Ranking Algorithm for SVNT's:** This algorithm introduced by[2] using the membership score (S^+), the non-membership score (S^-) and the average functions (C) (see Definition 1.4). And it works as follows:

  Let $A = (T, I, F)$ and $B = (T', I', F')$ be two SVNT's. Then (The symbol $>_N$ will be used to refer this ordering):
  1. **Step 1:** If $S^+(A) > S^+(B)$ ($S^+(A) < S^+(B)$), then $A >_N B$ ($A <_N B$). Otherwise, go to Step 2.
  2. **Step 2:** If $S^-(A) > S^-(B)$ ($S^-(A) < S^-(B)$), then $A <_N B$ ($A >_N B$). Otherwise, go to Step 3.
  3. **Step 3:** If $C(A) > C(B)$ ($C(A) < C(B)$), then $A >_N B$ ($A <_N B$). Otherwise, $A = B$.

Single-Valued MutliNeutrosophic Tuplets ($SVMNT's$) or n-Valued Neutrosophic Tuplets ($NVNT's$) can be ordered using one of the following algorithms:

1. **The S-Ranking Algorithm for SVMNT's with same (p, q, r)-forms:**[8] This algorithm is introduced by F. Smarandache to order SVMNT's and it works as follows:
   For any SVMNT $A = (T_1, ..., T_p; I_1, ..., I_q; F_1, ..., F_r)$ we compute the following:

   (a) The Average Positivity:
$$Avg^+(A) = \frac{\sum_{j=1}^{p} T_j + \sum_{k=1}^{q}(1 - I_k) + \sum_{l=1}^{r}(1 - F_l)}{p + q + r}$$

(b) The Average Truth-Falsehood:

$$Avg^{TF}(A) = \frac{\sum_{j=1}^{p} T_j - \sum_{l=1}^{r} F_l}{p+r}$$

(c) The Average Truth:

$$Avg^{T}(A) = \frac{\sum_{j=1}^{p} T_j}{p}$$

We compare any two SVMNT's
$A = (T_1, ..., T_p; I_1, ..., I_q; F_1, ..., F_r)$ and $B = (T'_1, ..., T'_p; I'_1, ..., I'_q; F'_1, ..., F'_r)$ with $p + q + r = n$ as follows:

(a) If $Avg^+(A) > Avg^+(B)$, then $A >_S B$. If $Avg^+(A) < Avg^+(B)$, then $A <_S B$. If $Avg^+(A) = Avg^+(B)$, then we go to the next step.

(b) If $Avg^{TF}(A) > Avg^{TF}(B)$, then $A >_S B$. If $Avg^{TF}(A) < Avg^{TF}(B)$, then $A <_S B$. If $Avg^{TF}(A) = Avg^{TF}(B)$, then we go to the next step.

(c) If $Avg^T(A) > Avg^T(B)$, then $A >_S B$. If $Avg^T(A) < Avg^T(B)$, then $A <_S B$. If $Avg^T(A) = Avg^T(B)$, then we have

i. $T = \frac{\sum_{j=1}^{p} T_j}{p} = \frac{\sum_{j=1}^{p} T'_j}{p} = T'$.

ii. $I = \frac{\sum_{k=1}^{q} I_k}{q} = \frac{\sum_{k=1}^{q} T'_k}{q} = I'$.

iii. $F = \frac{\sum_{l=1}^{r} F_j}{r} = \frac{\sum_{l=1}^{r} F'_j}{r} = F'$.

2. **The S-Ranking Algorithm for $SVMNT's$ with different (p, q, r)-forms:**[8] This algorithm is introduced by F. Smarandache to order SVMNT's with different (p, q, r)-forms, and it works as follows:
Let $A = (T_1, ..., T_{p_1}; I_1, ..., I_{q_1}; F_1, ..., F_{r_1})$ and $B = (T'_1, ..., T'_{p_2}; I'_1, ..., I'_{q_2}; F'_1, ..., F'_{r_2})$ be two $SVMNT's$ with $p_1 + q_1 + r_1 = n_1$ and $p_2 + q_2 + r_2 = n_2$. And let T_a, I_a, F_a, T'_a, I'_a and F'_a be the classical averages of the sets:
$\{T_1, ..., T_{p_1}\}, \{I_1, ..., I_{q_1}\}, \{F_1, ..., F_{r_1}\}, \{T'_1, ..., T'_{p_2}\}, \{I'_1, ..., I'_{q_2}\}$ and $\{F'_1, ..., F'_{r_2}\}$, respectively. Set $A_a = (T_a, I_a.F_a)$ and $B_a = (T'_a, I'_a, F'_a)$. We order A_a and B_a using the S-Ranking Algorithm for SVNT's:

$$\text{If } A_a >_S B_a \ (A_a <_S B_a), \text{ then } A >_S B \ (A <_S B).$$

3. **The N-Ranking algorithm for $NVNT's$:**[3] used the **N-Ranking Algorithm for SVNT's** to rank NVNT's as follows:

Let $A = (T_1, ..., T_{p_1}; I_1, ..., I_{q_1}; F_1, ..., F_{r_1})$ and $B = (T'_1, ..., T'_{p_2}; I'_1, ..., I'_{q_2}; F'_1, ..., F'_{r_2})$ be two $NVNT's$ with $p_1 + q_1 + r_1 = p_2 + q_2 + r_2 = n$ and the true degrees, indeterminacy degrees and false degrees are in ascending order. Let $k = lcm\{p_1, q_1, r_1, p_2, q_2, r_2\}$. Rewrite A and B (actually we define new sets so we will use the symbols A^* and B^* instead of A and B) as follows:

$$A^* = (\underbrace{\overbrace{T_1, ..., T_1}^{x_1-times}, \overbrace{T_2, ..., T_2}^{x_1-times}, ..., \overbrace{T_{p_1}, ..., T_{p_1}}^{x_1-times}}_{k-times}; \underbrace{\overbrace{I_1, ..., I_1}^{y_1-times}, \overbrace{I_2, ..., I_2}^{y_1-times}, ..., \overbrace{I_{q_1}, ..., I_{q_1}}^{y_1-times}}_{k-times}; \underbrace{\overbrace{F_1, ..., F_1}^{z_1-times}, \overbrace{F_2, ..., F_2}^{z_1-times}, ..., \overbrace{F_{r_1}, ..., F_{r_1}}^{z_1-times}}_{k-times})$$

$$B^* = (\underbrace{\overbrace{T'_1, ..., T'_1}^{x_2-times}, \overbrace{T'_2, ..., T'_2}^{x_2-times}, ..., \overbrace{T'_{p_2}, ..., T'_{p_2}}^{x_2-times}}_{k-times}; \underbrace{\overbrace{I'_1, ..., I'_1}^{y_2-times}, \overbrace{I'_2, ..., I'_2}^{y_2-times}, ..., \overbrace{I'_{q_2}, ..., I'_{q_2}}^{y_2-times}}_{k-times}; \underbrace{\overbrace{F'_1, ..., F'_1}^{z_2-times}, \overbrace{F'_2, ..., F'_2}^{z_2-times}, ..., \overbrace{F'_{r_2}, ..., F'_{r_2}}^{z_2-times}}_{k-times})$$

Where $k = x_1 \times p_1 = y_1 \times q_1 = z_1 \times r_1 = x_2 \times p_2 = y_2 \times q_2 = z_2 \times r_2$. Define

$$(T_0, I_0, F_0) = \left(\frac{\sum_{t=1}^{p_1} x_1 T_t}{k}, \frac{\sum_{t=1}^{q_1} y_1 I_t}{k}, \frac{\sum_{t=1}^{r_1} z_1 F_t}{k} \right)$$

$$(T'_0, I'_0, F'_0) = \left(\frac{\sum_{t=1}^{p_2} x_2 T'_t}{k}, \frac{\sum_{t=1}^{q_2} y_2 I'_t}{k}, \frac{\sum_{t=1}^{r_2} z_2 F'_t}{k} \right)$$

The N-Ranking algorithm for $NVNT's$ works as follows:
Step 1: If $S^+(T_0,I_0,F_0) > S^+(T'_0,I'_0,F'_0)$, then $A >_N B$. And if $S^+(T'_0,I'_0,F'_0) < S^+(T_0,I_0,F_0)$, then $B <_N A$. If $S^+(T_0,I_0,F_0) = S^+(T'_0,I'_0,F'_0)$, then we go to step 2.
Step 2: If $S^-(T_0,I_0,F_0) > S^-(T'_0,I'_0,F'_0)$, then $A <_N B$. And if $S^-(T'_0,I'_0,F'_0) < S^-(T_0,I_0,F_0)$, then $B >_N A$. If $S^-(T_0,I_0,F_0) = S^-(T'_0,I'_0,F'_0)$, then we go to step 3.
Step 3: If $C(T_0,I_0,F_0) > C(T'_0,I'_0,F'_0)$, then $A >_N B$. And if $C(T'_0,I'_0,F'_0) < C(T_0,I_0,F_0)$, then $B <_N A$. If $C(T_0,I_0,F_0) = C(T'_0,I'_0,F'_0)$, then we go to step 4.
Step 4: We compare (T_m,I_m,F_m) and (T'_m,I'_m,F'_m) for $m = k, k-1, ..., 1$ by steps 1-3. We have to get a decision, otherwise we have $A = B$.

Theorem 1.5. [3] *] The N-ranking algorithm inherits a total order on the set of all n-valued neutrosophic tuplets.*

In the next section we will show that Theorem 1.5 is incorrect; more precisely, we will show that there are tow different $NVNT's$ A and B, with neither $A <_N B$ nor $B <_N A$.

2 Remarks on the Existing Ranking Algorithms for $SVNT's$ and $NVNT's$

Remark 2.1. S-algorithm and N-Ranking Algorithm for $SVNT's$ are two different orderings. That is there are two different single valued neutrosophic triplets $A = (T,I,F)$ and $B = (T',I',F')$ with $A <_S B$ and $A >_N B$.

See the following example.

Example 2.2. In this example we will find two single valued neutrosophic triplets $A = (T,I,F)$ and $B = (T',I',F')$ with $A >_S B$ and $A <_N B$. Finding such triplets is not straightforward. Let
$A = (0.6, 0.9, 0.2)$ and $B = (0.5, a, 0.3)$ where a is any real number in $[0,1]$. To find the parameter a such that: $S^+(A) < S^+(B)$ and $s(B) < s(A)$ (which implies $A <_N B$ and $A >_S B$).
If $S^+(A) < S^+(B)$, then
$S^+(A) = \frac{2+(0.6-0.2)(2-0.9)-0.9}{4} = 0.385 < S^+(B) = \frac{2+(0.5-0.3)(2-a)-a}{4} = 0.6-0.3a$. Which give us the first inequality $0.385 < 0.6-0.3a$. We solve this inequality for a to get $a < 0.7166....$
On the other hand $s(B) < s(A)$ implies: $s(B) = \frac{2+0.5-0.3-a}{3} < s(A) = \frac{2+0.6-0.2-0.9}{3} = 0.5$. Which give us the inequality $\frac{2.2-a}{3} < 0.5$. Solving it for a implies $a > 0.7$.
So $a \in [0.7, 0.71666...]$. That is $A <_S B$ and $A >_N B$ for any $a \in (0.7, 0.71666...)$. For example if $a = 0.71 \in (0.7, 0.71666...)$ we have:

$$S^+(B) = \frac{2+(0.5-0.3)(2-0.71)-0.71}{4} = 0.387 > S^+(A) = 0.385. \quad \text{So } B >_N A.$$

On the other hand,

$$s(B) = \frac{2+0.5-0.3-0.71}{3} = 0.49666... < s(A) = 0.5. \text{ So } B <_S A.$$

Example 2.2 shows that $<_S$ and $<_N$ are two different orderings (rankings) for SVNT's. Hence S-algorithm and N-Algorithm give us two different decisions in applications. Therefore we need to decide which one is more suitable for applications than the other.

Remark 2.3. There are two different $NVNT's$ A and B with the same ranking using the N-Ranking and the S-Ranking algorithms for $NVNT's$ ($SVMNT's$). See the following example.

Example 2.4. Consider the following two NVNT's:

$$A = (0.9, 0.9, 0.9, 0.9\ ;\ 0.2\ ;\ 0.1, 0.1) \text{ and } B = (0.9\ ;\ 0.2, 0.2\ ;\ 0.1, 0.1, 0.1, 0.1)$$

It is clear that A and B are two different NVNT's with $p_1 + q_1 + r_1 = p_2 + q_2 + r_2 = 6$. We will rank A and B by the N-Ranking and the S-Ranking algorithms:

1. Using the N-Ranking algorithm for $NVNT's$:
   First of all, we compute $k = lcm\{p_1, q_1, r_1, p_2, q_2, r_2\} = \{4, 1, 2, 1, 2, 4\} = 4$. Thus we rewrite A and B as follows:

$$A^* = (0.9, 0.9, 0.9, 0.9 \ ; \ 0.2, 0.2, 0.2, 0.2 \ ; \ 0.1, 0.1, 0.1, 0.1) \quad \text{and}$$
$$B^* = (0.9, 0.9, 0.9, 0.9 \ ; \ 0.2, 0.2, 0.2, 0.2 \ ; \ 0.1, 0.1, 0.1, 0.1).$$

   Note that $A^* = B^*$, thus

$$(T_0, I_0, F_0) = \left(\frac{\sum_{t=1}^{p_1} x_1 T_t}{k}, \frac{\sum_{t=1}^{q_1} y_1 I_t}{k}, \frac{\sum_{t=1}^{r_1} z_1 F_t}{k} \right) = (0.9, 0.2, 0.1) \quad \text{and}$$

$$(T'_0, I'_0, F'_0) = \left(\frac{\sum_{t=1}^{p_2} x_2 T'_t}{k}, \frac{\sum_{t=1}^{q_2} y_2 I'_t}{k}, \frac{\sum_{t=1}^{r_2} z_2 F'_t}{k} \right) = (0.9, 0.2, 0.1)$$

   Since $(T_0, I_0, F_0) = (T'_0, I'_0, F'_0)$, we have

$$S^+(T_0, I_0, F_0) = S^+(T'_0, I'_0, F'_0)$$
$$S^-(T_0, I_0, F_0) = S^-(T'_0, I'_0, F'_0)$$
$$C(T_0, I_0, F_0) = C(T'_0, I'_0, F'_0)$$

   So we do not have a decision for A and B using Step 1, Step 2 and Step 3 of the N-Ranking algorithm. Thus we have to go to Step 4. But $T_t = T'_t$, $I_t = I'_t$ and $F_t = F'_t$ for every $t = k, k-1, ..., 1$, so Step 4, also, fails to give us a decision. Therefore (As proved in[3]) we have $A = B$. Which is not true, since A and B are two different $NVNT's$.

2. Using the S-Ranking algorithm for SVMNT's with different (p, r, s)-forms:
   It is clear that A and B have different forms, since A and B have the forms $(4, 1, 2)$ and $(1, 2, 4)$ respectively. We need to compute the triplets (T_a, I_a, F_a) and (T'_a, I'_a, F'_a) where T_a, I_a, F_a, T'_a, I'_a and F'_a are the classical averages of the sets:
   $\{T_1, T_2, T_3, T_4\}$, $\{I_1\}$, $\{F_1, F_2\}$, $\{T'_1\}$, $\{I'_1, I'_2\}$ and $\{F'_1, F'_2, F'_3, F'_4\}$, respectively.
   Obviously! $(T_a, I_a, F_a) = (0.9, 0.2, 0.1) = (T'_a, I'_a, F'_a)$. Which means this algorithm, also, failed to give us a decision for these two different $NVNT's$.

Example 2.4 shows the following:

Corollary 2.5. *Neither the S-ranking nor the N-ranking algorithm for $NVNT's$ ($SVMNT's$) is a total ordering for $\mathfrak{M}$.*

The above corollary says that we may fail to have a decision about two different $NVNT's$ using he S-ranking and the N-ranking algorithms. In the following section we will improve these two ranking algorithms to make them more powerful tools to rank $NVNT's$.

3 Improvement for the S-ranking and the N-ranking algorithms

Remark 2.1 and Remark 2.3 in the previous section show that the existing ranking algorithms for NVNT's may fail to give us a decision in some applications (seen Example 2.4). In this section we will introduce improvements for the S-ranking and N-ranking algorithms. Before doing this we need to agree when two different $NVNT's$ can be considered the same? Consider the following two $NVNT's$:

$$A = (0.3, 0.9, 0.4 \ ; \ 0.5, 0.8 \ ; \ 0.7, 0.1, 0.2)$$

$$B = (0.9, 0.4, 0.3 \ ; \ 0.8, 0.5 \ ; \ 0.1, 0.2, 0.7)$$

It is intuitive to assume that A and B doing the same (having the same ranking), since they have the same degrees of membership, indeterminacy and non-membership, but with different orders. This push us to define the concept of equivalent $NVNT's$.

Definition 3.1. Let $A = (T_1, ..., T_{p_1}; I_1, ..., I_{q_1}; F_1, ..., F_{r_1})$ and $B = (T'_1, ..., T'_{p_2}; I'_1, ..., I'_{q_2}; F'_1, ..., F'_{r_2})$ be two $NVNT's$ with $p_1 + q_1 + r_1 = n_1$ and $p_2 + q_2 + r_2 = n_2$. A and B are said *equivalent* (in Symbols $A \approx B$) if an only if there are three bijections:

$$f : \{T_1, ..., T_{p_1}\} \longrightarrow \{T'_1, ..., T'_{p_2}\}$$

$$g : \{I_1, ..., I_{q_1}\} \longrightarrow \{I'_1, ..., I'_{q_2}\}$$

$$h : \{F_1, ..., F_{r_1}\} \longrightarrow \{F'_1, ..., F'_{r_2}\}$$

Such that $f(T) = T$ for every $T \in \{T_1, ..., T_{p_1}\}$, $g(I) = I$ for every $I \in \{I_1, ..., I_{q_1}\}$ and $h(F) = F$ for every $F \in \{F_1, ..., F_{r_1}\}$.

Roughly saying, $A \approx B$ if and only if A and B have the same degrees of membership, indeterminacy and non-membership but maybe with different order.

Proposition 3.2. *The relation $\approx$ in Definition 3.1 is an equivalence relation on $\mathfrak{N}$: The set of all $NVNT's$.*

The equivalence classes of $\approx$ are:

$$\mathfrak{N}/\approx \; = \{[A]; A \in \mathfrak{N}\} \text{ where } [A] = \{B \in \mathfrak{N}; A \approx B\}.$$

The following assumption is necessary.

Assumption 3.3. [Axiom of Ranking] If the available sources have equal weights, then for any ranking algorithm $\mathcal{L}$ on $\mathfrak{N}$ and any $A, B \in \mathfrak{N}$ with $A \approx B$, we have $\mathcal{L}(A) = \mathcal{L}(B)$.

The axiom or ranking says that equivalent $NVNT's$ will be awarded the same rank in any ranking algorithm. That is; equivalent $NVNT's$ concept is like the concept of homeomorphic spaces in topology, or isomorphic groups in group theory, we do not see them different in any ranking algorithms.

Remark 3.4. Henceforth, we will consider the equivalent class $[A] = \{B \in \mathfrak{N}; A \approx B\}$ as one n-valued neutrosophic tuplet and it will be simply written as A (where $A \in [A]$ is a representative element of $[A]$). And the rank of A is a rank for every $B \in [A]$.

We go back to the improvements of the existing ranking algorithms of $NVNT's$. Our work built on the following assumption about the number of sources by F. Smarandache:“ The more sources evaluate a subject, the better accurate result”. This means that if we have two different $NVNT's$ with the same ranking using any ranking algorithm but they have different (p, q, r)-forms, then we have to consider the number of sources p, q, r and $n = p + q + r$ in our ranking to decide which one is doing better. To that end we need the following new concept.

Definition 3.5. Let $A = (T_1, ..., T_p; I_1, ..., I_q; F_1, ..., F_r)$ be an n-valued neutrosophic tuplets with $p+q+r = n$ and let

$$(T, I, F) = \left(\frac{\sum_{t=1}^{p} T_t}{p}, \frac{\sum_{t=1}^{q} I_t}{q}, \frac{\sum_{t=1}^{r} F_t}{r} \right)$$

The weighted triplet of A is denoted by $W(A)$ and is defined by:

$$W(A) = (\frac{T \times p}{n}, \frac{I \times q}{n}, \frac{F \times r}{n}) = \left(\frac{\sum_{t=1}^{p} T_t}{n}, \frac{\sum_{t=1}^{q} I_t}{n}, \frac{\sum_{t=1}^{r} F_t}{n} \right)$$

Now, we are ready for improve the existing algorithms for $NVNT's$.

3.1 The Integrated N-Ranking Algorithm for $NVNT's$

The integrated N-ranking algorithm for $NVNT's$ is an improvement for the N-Ranking Algorithm for $NVNT's$ and it works as follows:
Let $A = (T_1, ..., T_{p_1}; I_1, ..., I_{q_1}; F_1, ..., F_{r_1})$ and $B = (T'_1, ..., T'_{p_2}; I'_1, ..., I'_{q_2}; F'_1, ..., F'_{r_2})$ be two $NVNT's$ with $p_1 + q_1 + r_1 = n_1$ and $p_2 + q_2 + r_2 = n_2$, and the true degrees, indeterminacy degrees and false degrees are in ascending order. Let $k = lcm\{p_1, q_1, r_1, p_2, q_2, r_2\}$ and

$$(T_0, I_0, F_0) = \left(\frac{\sum_{t=1}^{p_1} x_1 T_t}{k}, \frac{\sum_{t=1}^{q_1} y_1 I_t}{k}, \frac{\sum_{t=1}^{r_1} z_1 F_t}{k}\right)$$

$$(T'_0, I'_0, F'_0) = \left(\frac{\sum_{t=1}^{p_2} x_2 T'_t}{k}, \frac{\sum_{t=1}^{q_2} y_2 I'_t}{k}, \frac{\sum_{t=1}^{r_2} z_2 F'_t}{k}\right)$$

A and B will be ranked as follows:
Apply **Step 1**, **Step 2**, **Step 3** and **Step 4** in the N-Ranking Algorithm for $NVNT's$ in sequence till you have a decision ([3]). If you do not, then compute the weighted triplets of A and B and go to Step 5:

$$W(A) = (\frac{T_0 \times p_1}{n_1}, \frac{I_0 \times q_1}{n_1}, \frac{F_0 \times r_1}{n_1}) \quad \text{and} \quad W(B) = (\frac{T'_0 \times p_2}{n_2}, \frac{I'_0 \times q_2}{n_2}, \frac{F'_0 \times r_2}{n_2}).$$

Step 5: Compute $S^+(W(A))$ and $S^+(W(B))$. If $S^+(W(A)) < S^+(W(B))$, then $A <_N B$. And if $S^+(W(A)) > S^+(W(B))$, then $A >_N B$. If $S^+(W(A)) = S^+(W(B))$, then go to step 6.
Step 6: Compute $S^-(W(A))$ and $S^-(W(B))$. If $S^-(W(A)) < S^-(W(B))$, then $A >_N B$. And if $S^-(W(A)) > S^+(W(B))$, then $A <_N B$. If $S^-(W(A)) = S^-(W(B))$, then go to step 7.
Step 7: Compute $C(W(A))$ and $C(W(B))$. If $C(W(A)) < C(W(B))$, then $A <_N B$. And if $C(W(A)) > C(W(B))$, then $A >_N B$. If $C(W(A)) = C(W(B))$, then go to step 8.
Step 8: If $S^+(T_0, I_0, F_0) > 0.5$ ($S^+(T_0, I_0, F_0) < 0.5$) and $n_1 > n_2$, then $A >_N B$ ($A <_N B$). And if $S^+(T_0, I_0, F_0) > 0.5$ ($S^+(T_0, I_0, F_0) < 0.5$) and $n_1 < n_2$, then $A <_N B$ ($A >_N B$). If $S^+(A) = 0.5$, then go to step 9.
Step 9: If $S^-(T_0, I_0, F_0) > 0.5$ ($S^-(T_0, I_0, F_0) < 0.5$) and $n_1 > n_2$, then $A <_N B$ ($A >_N B$). And if $S^-(T_0, I_0, F_0) > 0.5$ ($S^-(T_0, I_0, F_0) < 0.5$) and $n_1 < n_2$, then $A >_N B$ ($A <_N B$). If $S^-(A) = 0.5$, then go to step 10.
Step 10: If $C(T_0, I_0, F_0) > 0.5$ ($C(T_0, I_0, F_0) < 0.5$) and $n_1 > n_2$, then $A >_N B$ ($A <_N B$). And if $C(T_0, I_0, F_0) > 0.5$ ($C(T_0, I_0, F_0) < 0.5$) and $n_1 < n_2$, then $A <_N B$ ($A >_N B$). If $C(A) = 0.5$, then go to step 11.
Step 11: If $n_1 > n_2$ ($n_1 < n_2$), then $A >_N B$ ($A <_N B$).

It is important to note that we did not assume any thing about the number of sources, that is we may have $n_1 \neq n_2$. The first thing we will do is to show that $<_N$ is a total ordering of the set of all n-valued neutrosophic tuplets.

Theorem 3.6. *$<_N$ is a total ordering on the set of all n-valued neutrosophic tuplets.*

Proof. Let A and B be two none equivalent $NVNT's$ (see Proposition 3.2, Assumption 3.3 and Remark 3.4). It is sufficient to show that either $A <_N B$ or $B <_N A$. Or equivalently, if neither $A <_N B$ nor $B <_N A$, then $A \approx B$. Suppose that neither $A <_N B$ nor $B <_N A$. Then we do not have a decision using Steps 1-11. Since we do not have a decision using steps 1-4, we have $(T_0, I_0, F_0) = (T'_0, I'_0, F'_0)$ and $A^* = B^*$ (see the N-ranking of $SVNT's$). Having $A^* = B^*$ implies $A = B$ only if $n_1 = n_2$, $p_1 = p_2$, $q_1 = q_2$ and $r_1 = r_2$. But we do not have a decision using steps 5-7 which are a ranking of $W(A)$ and $W(B)$ using the N-ranking of $SVNT's$, so $W(A) = W(B)$. Or, equivalently,

$$(\frac{T_0 \times p_1}{n_1}, \frac{I_0 \times q_1}{n_1}, \frac{F_0 \times r_1}{n_1}) = (\frac{T'_0 \times p_2}{n_2}, \frac{I'_0 \times q_2}{n_2}, \frac{F'_0 \times r_2}{n_2}).$$

But $(T_0, I_0, F_0) = (T'_0, I'_0, F'_0)$, so

$$\frac{p_1}{n_1} = \frac{p_2}{n_2} \quad (1)$$

$$\frac{q_1}{n_1} = \frac{q_2}{n_2} \quad (2)$$

$$\frac{r_1}{n_1} = \frac{r_2}{n_2} \quad (3)$$

Since steps 8-11 did not give us a decision, we have $n_1 = n_2$ and $p_1 = p_2$, $q_1 = q_2$ and $r_1 = r_2$. Thus A and B are of the same (p, q, r)-forms. Since $A^* = B^*$, we have $T_t = T'_t$, $I_t = I'_t$ and $F_t = F'_t$ for every $t = k, k-1, ..., 1$ (See the N-ranking algorithm above). That is $A \approx B$ an we done. □

Before going for examples note that in Steps 8-11 we use the comparison $n_1 < n_2$ or $n_1 > n_1$, which raises the question, what will happen in the Integrated N-ranking algorithm for $SVNT's$ if we have $n_1 = n_2$? The following theorem has an answer.

Theorem 3.7. *Let $A = (T_1, ..., T_{p_1}; I_1, ..., I_{q_1}; F_1, ..., F_{r_1})$ and $B = (T'_1, ..., T'_{p_2}; I'_1, ..., I'_{q_2}; F'_1, ..., F'_{r_2})$ be two n-valued neutrosophic tuplets with $n_1 = n_2$. If we do not have a decision using the Steps 1-7 in the integrated N-ranking algorithm, then $A \approx B$.*

Proof. Obvious! see the proof of Theorem 3.6. □

The above theorem says the following: We need Steps 8-11 in the integrated N-ranking algorithm only if $n_1 \neq n_2$ and Steps 1-7 did not make a decision.

Our first example is ranking the $NVNT's$ A and B mentioned in Example 2.4 (note that we do not have a decision about A and B using the N-ranking algorithm).

Example 3.8. Let $A = (0.9, 0.9, 0.9, 0.9\ ;\ 0.2\ ;\ 0.1, 0.1)$ and $B = (0.9\ ;\ 0.2, 0.2\ ;\ 0.1, 0.1, 0.1, 0.1)$. We will rank A and B by the integrated N-ranking algorithm for $NVNT's$. From Example 2.4 we have Steps 1-4 failed to give us a decision and $(T_0, I_0, F_0) = (T'_0, I'_0, F'_0) = (0.9, 0.2, 0.1)$. We compute the weighted triplet of A and B:

$$W(A) = (\frac{T_0 \times p_1}{n_1}, \frac{I_0 \times q_1}{n_1}, \frac{F_0 \times r_1}{n_1}) = (\frac{0.9 \times 4}{7}, \frac{0.2 \times 1}{7}, \frac{0.1 \times 2}{7}) = (\frac{3.6}{7}, \frac{0.2}{7}, \frac{0.2}{7}).$$

$$W(B) = (\frac{T'_0 \times p_2}{n_2}, \frac{I'_0 \times q_2}{n_2}, \frac{F'_0 \times r_2}{n_2}) = (\frac{0.9 \times 1}{7}, \frac{0.2 \times 2}{7}, \frac{0.1 \times 4}{7}) = (\frac{0.9}{7}, \frac{0.4}{7}, \frac{0.4}{7}).$$

We go to Step 5. First we compute $S^+(W(A))$ and $S^+(W(B))$.

$$S^+(W(A)) = \frac{2 + (T-F)(2-I) - I}{4} = \frac{2 + \left(\frac{3.6}{7} - \frac{0.2}{7}\right)\left(2 - \frac{0.2}{7}\right) - \frac{0.2}{7}}{4} \cong 0.73.$$

$$S^+(W(B)) = \frac{2 + (T-F)(2-I) - I}{4} = \frac{2 + \left(\frac{0.9}{7} - \frac{0.4}{7}\right)\left(2 - \frac{0.4}{7}\right) - \frac{0.4}{7}}{4} \cong 0.52.$$

Since $S^+(W(A)) \cong 0.73 > S^+(W(B)) \cong 0.52$, we have a decision using Step 5 which is: $A >_N B$.

In the following example steps 1-7 fail to make a decision, but Step 8 successes.

Example 3.9. Let $A = (0.3, 0.3\ ;\ 0.6, 0.6, 0.6\ ;\ 0.7)$ and $B = (0.3, 0.3, 0.3, 0.3\ ;\ 0.6, 0.6, 0.6, 0.6, 0.6, 0.6\ ;\ 0.7, 0.7)$. Since $(T_0, I_0, F_0) = (T'_0, I'_0, F'_0) = (0.3, 0.6, 0.7)$, we have Steps 1-4 fail to make a decision. So, we go to the next step by computing the weighted triplet of A and B:

$$W(A) = (\frac{T_0 \times p_1}{n_1}, \frac{I_0 \times q_1}{n_1}, \frac{F_0 \times r_1}{n_1}) = (\frac{0.3 \times 2}{6}, \frac{0.6 \times 3}{6}, \frac{0.7 \times 1}{6}) = (\frac{0.6}{6}, \frac{1.8}{6}, \frac{0.7}{6}).$$

$$W(B) = (\frac{T'_0 \times p_2}{n_2}, \frac{I'_0 \times q_2}{n_2}, \frac{F'_0 \times r_2}{n_2}) = (\frac{0.3 \times 4}{12}, \frac{0.6 \times 6}{12}, \frac{0.7 \times 2}{12}) = (\frac{0.6}{6}, \frac{1.8}{6}, \frac{0.7}{6}).$$

Since $W(A) = W(B)$, Steps 5-7 fail to make a decision, too. Thus, we go for the next step. First, we need to compute $S^+(T_0, I_0, F_0)$ (which equals $S^+(T'_0, I'_0, F'_0)$). $S^+(T_0, I_0, F_0) = \frac{2+(0.3-0.7)(2-0.6)-0.6}{4} = 0.21$. Since $S^+(T_0, I_0, F_0) = 0.21 < 0.5$ and $n_1 = 6 < n_2 = 12$, we have a decision using the second part of Step 8 which is; $A >_N B$.

In the following example Steps 1-8 fail and Step 9 successes to make a decision.

Example 3.10. Let $A = (0.55, 0.55, 0.55 \; ; \; 0.4, 0.4 \; ; \; 0.3, 0.3)$ and
$B = (0.55, 0.55, 0.55, 0.55, 0.55, 0.55 \; ; \; 0.4, 0.4, 0.4, 0.4 \; ; \; 0.3, 0.3, 0.3, 0.3)$.
Since $(T_0, I_0, F_0) = (T_0', I_0', F_0') = (0.55, 0.4, 0.3)$, Steps 1-4 fail to give us a decision. We go to the next step by computing the weighted triplet of A and B:

$$W(A) = (\frac{T_0 \times p_1}{n_1}, \frac{I_0 \times q_1}{n_1}, \frac{F_0 \times r_1}{n_1}) = (\frac{0.55 \times 3}{7}, \frac{0.4 \times 2}{7}, \frac{0.3 \times 2}{7}) = (\frac{1.65}{7}, \frac{0.8}{7}, \frac{0.6}{7}).$$
$$W(B) = (\frac{T_0' \times p_2}{n_2}, \frac{I_0' \times q_2}{n_2}, \frac{F_0' \times r_2}{n_2}) = (\frac{0.55 \times 6}{14}, \frac{0.4 \times 4}{14}, \frac{0.3 \times 4}{14}) = (\frac{1.65}{7}, \frac{0.8}{7}, \frac{0.6}{7}).$$

Since $W(A) = W(B)$, we have Steps 5-7 fail, also, to give us a decision. We go to Step 8 which is requiring $S^+(T_0, I_0, F_0) \neq 0.5$. But $S^+(T_0, I_0, F_0) = \frac{2+(0.55-0.3)(2-0.4)-0.4}{4} = 0.5$, so Steps 8 fails to give us a decision. We compute $S^-(T_0, I_0, F_0)$ and go to Step 9.
$S^-(T_0, I_0, F_0) = \frac{2+(F_0-T_0)(2-I_0)-I_0}{4} = \frac{2+(0.3-0.55)(2-0.4)-0.4}{4} = 0.3$. Since $n_1 = 7 < n_2 = 14$, we have $A >_N B$ and $S^-(T_0, I_0, F_0) < 0.5$, we have (by the second part of Step 9) $A >_N B$.

Before proceeding, we need the following Proposition.

Proposition 3.11. *Let (T, I, F) be a neutrosophic triplet.*

1. *If $S^+(T, I, F) = S^+(T, I, F) = 0.5$, then $I = 0.0$ and $T = F$.*
2. *If $S^+(T, I, F) = S^+(T, I, F) = C(T, F, I) = 0.5$, then $I = 0.0$ and $T = F = 0.5$.*

Proof. Obvious! we only need to solve simple linear equations. □

In the following example Steps 1-9 fail and Step 10 successes to give us a decision. From Proposition 3.11 we conclude that: If we are looking for an example where steps 8 and 9 fail, then we have to have $I_0 = 0$ and $T_0 = F_0$.

Example 3.12. Let $A = (0.4, 0.4, 0.4 \; ; \; 0, 0 \; ; \; 0.4, 0.4)$ and
$B = (0.4, 0.4, 0.4, 0.4, 0.4, 0.4 \; ; \; 0, 0, 0, 0 \; ; \; 0.4, 0.4, 0.4, 0.4)$.
Since $(T_0, I_0, F_0) = (T_0', I_0', F_0') = (0.4, 0, 0.4)$, Steps 1-4 fail to give us a decision. We go to the next step by computing the weighted triplet of A and B:

$$W(A) = (\frac{T_0 \times p_1}{n_1}, \frac{I_0 \times q_1}{n_1}, \frac{F_0 \times r_1}{n_1}) = (\frac{0.4 \times 3}{7}, \frac{0 \times 2}{7}, \frac{0.4 \times 2}{7}) = (\frac{1.2}{7}, 0, \frac{1.2}{7}) = W(B).$$

Since $W(A) = W(B)$, we have Steps 5-7 fail, also, to give us a decision. We go to Step 8 which is requiring $S^+(T_0, I_0, F_0) \neq 0.5$. But $S^+(T_0, I_0, F_0) = \frac{2+(0.4-0.4)(2-0)-0}{4} = 0.5$, so Steps 8 fails to give us a decision. We compute $S^-(T_0, I_0, F_0)$ and go to Step 9.
But $S^-(T_0, I_0, F_0) = \frac{2+(0.4-0.4)(2-0)-0}{4} = 0.5$. Thus Step 9 fails too. We go to Step 10. First, we compute $C(T_0, I_0, F_0)$. Since $C(T_0, I_0, F_0) = \frac{T_0+F_0}{2} = \frac{0.4+0.4}{2} = 0.4 < 0.5$ and $n_1 = 7 < n_2 = 14$, we have $A >_N B$ (by the second part of Step 10) $A >_N B$.

In the following example only Step 11 successes to give us a decision. From Proposition 3.11, in such examples we must have $T_0 = F_0 = 0.5$ and $I_0 = 0$.

Example 3.13. Let $A = (0.5, 0.5, 0.5 \; ; \; 0, 0 \; ; \; 0.5, 0.5)$ and $B = (0.5, 0.5, 0.5, 0.5, 0.5, 0.5 \; ; \; 0, 0, 0, 0 \; ; \; 0.5, 0.5, 0.5, 0.5)$. Since $(T_0, I_0, F_0) = (T_0', I_0', F_0') = (0.5, 0, 0.5)$, Steps 1-4 fail to give us a decision. We go to the next step by computing the weighted triplet of A and B:

$$W(A) = (\frac{T_0 \times p_1}{n_1}, \frac{I_0 \times q_1}{n_1}, \frac{F_0 \times r_1}{n_1}) = (\frac{0.5 \times 3}{7}, \frac{0 \times 2}{7}, \frac{0.5 \times 2}{7}) = (\frac{1.5}{7}, 0, \frac{1.5}{7}) = W(B).$$

Since $W(A) = W(B)$, we have Steps 5-7 fail, also, to give us a decision. We go to Steps 8-10. But $S^{+}(T_0, I_0, F_0) = \frac{2+(0.5-0.5)(2-0)-0}{4} = 0.5$, $S^{-}(T_0, I_0, F_0) = \frac{2+(0.5-0.5)(2-0)-0}{4} = 0.5$ and $C(T_0, I_0, F_0) = \frac{T+F}{2} = \frac{0.5+0.5}{2} = 0.5$. Thus Steps 9-10 fail to give us a decision. We go to Step 11. Since $n_1 = 7 < n_2 = 14$, we have (by Step 11) $A <_N B$.

The above discussion shows that every step in Integrated N-Ranking Algorithm of $NVNT's$ is necessary.

In the following section we will improve the S-ranking algorithm for $SVMNT's$.

3.2 Integrated S-Ranking Algorithm for $SVMNT's$ with Different (p, q, r)-forms

The integrated S-Ranking Algorithm for $SVMNT's$ with different (p, q, r)-forms is an improvement for the S-Ranking Algorithm for $SVMNT's$ with different (p, q, r)-forms and it works as follows:
Let $A = (T_1, ..., T_{p_1}; I_1, ..., I_{q_1}; F_1, ..., F_{r_1})$ and $B = (T'_1, ..., T'_{p_2}; I'_1, ..., I'_{q_2}; F'_1, ..., F'_{r_2})$ be two $SVMNT's$ with $p_1 + q_1 + r_1 = n_1$, $p_2 + q_2 + r_2 = n_2$ and $(p_1, r_1, s_1) \neq (p_2, r_2, s_2)$ (we may have $n_1 = n_2$). And let T_a, I_a, F_a, T'_a, I'_a and F'_a be the classical averages of the sets:
$\{T_1, ..., T_{p_1}\}$, $\{I_1, ..., I_{q_1}\}$, $\{F_1, ..., F_{r_1}\}$, $\{T'_1, ..., T'_{p_2}\}$, $\{I'_1, ..., I'_{q_2}\}$ and $\{F'_1, ..., F'_{r_2}\}$, respectively. Set $A_a = (T_a, I_a, F_a)$ and $B_a = (T'_a, I'_a, F'_a)$. We rank A and B as follows:
Apply **Step 1**, **Step 2**, **Step 3** in the S-Ranking Algorithm in sequence till you have a decision. If you do not have a decision, then compute the weighted triplets of A and B:

$$W(A) = (\frac{T_a \times p_1}{n_1}, \frac{I_a \times q_1}{n_1}, \frac{F_a \times r_1}{n_1}) \quad \text{and} \quad W(B) = (\frac{T'_a \times p_2}{n_2}, \frac{I'_a \times q_2}{n_2}, \frac{F'_a \times r_2}{n_2}).$$

Step 4: Compute $s(W(A))$ and $s(W(B))$. If $s(W(A)) < s(W(B))$, then $A <_S B$. And if $s(W(A)) > s(W(B))$, then $A >_S B$. If $s(W(A)) = s(W(B))$, then go to Step 5.
Step 5: Compute $a(W(A))$ and $a(W(B))$. If $a(W(A)) < a(W(B))$, then $A <_S B$. And if $a(W(A)) > a(W(B))$, then $A >_S B$. If $a(W(A)) = a(W(B))$, then go to Step 6.
Step 6: Compute $c(W(A))$ and $c(W(B))$. If $c(W(A)) < c(W(B))$, then $A <_S B$. And if $c(W(A)) > c(W(B))$, then $A >_S B$. If $c(W(A)) = c(W(B))$, then go to Step 7.

If at this stage you do not have a decision, the you have $s(A_a) = s(B_a)$, $a(A_a) = a(B_a)$, $c(A_a) = c(B_a)$, $s(W(A)) = s(W(B))$, $a(W(A)) = a(W(B))$ and $c(W(A))$ and $c(W(B))$.

Step 7: If $s(A_a) > 0.5$ ($s(A_a) < 0.5$) and $n_1 > n_2$, then $A >_S B$ ($A <_S B$). And if $s(A_a) > 0.5$ ($s(A_a) < 0.5$) and $n_1 < n_2$, then $A <_S B$ ($A >_S B$). If $S(A_a) = 0.5$, then we go to step 8.
Step 8: If $a(A_a) > 0$ ($a(A_a) < 0$) and $n_1 > n_2$, then $A >_S B$ ($A <_S B$). And if $a(A_a) > 0$ ($a(A_a) < 0$) and $n_1 < n_2$, then $A <_S B$ ($A >_S B$). If $a(A_a) = 0$, then we go to step 9.
Step 9: If $c(A_a) > 0.5$ ($c(A_a) < 0.5$) and $n_1 > n_2$, then $A >_S B$ ($A <_S B$). And if $c(A_a) > 0.5$ ($c(A_a) < 0.5$) and $n_1 < n_2$, then $A <_S B$ ($A >_S B$). If $s(A_a) = 0.5$, then we go to step 10.
Step 10: If $n_1 > n_2$ ($n_1 < n_2$), then $A >_S B$ ($A <_S B$).

The first thing we will do is to show the following.

Theorem 3.14. *If A and B are two $SVMNT's$ with different (p, q, r)-forms, then $A <_S B$ or $A >_S B$.*

Proof. Suppose that A and B have different (p, q, r)-forms. To show that: $A <_S B$ or $A >_S B$. By contrapositive, suppose that: neither $A <_S B$ nor $A >_S B$. Which means we do not have a decision for A and B by steps 1 to 10 in the Integrated S-Ranking Algorithm. A fortiori, steps 4-6 did not give us a decision for $W(A)$ and $W(B)$, which implies (from the S-Ranking algorithm for $SVNT's$) $W(A) = W(B)$. And since we do not have a decision using steps 7-10, we have $n_1 = n_2$. If we solve the equations $W(A) = W(B)$ and $n_1 = n_2$, then we get $p_1 = p_2$, $q_1 = q_2$ and $r_1 = r_2$. That is A and B have the same (p, q, r)-forms, a contradiction. Thus $A <_S B$ or $A >_S B$. □

Note that in the above proof we have used "not having a decision using steps 7-10" to conclude that $n_1 = n_2$. Therefore; if we have $n_1 = n_2$, then we expect that a decision will be made using steps 1-6. See the following theorem.

Theorem 3.15. *If A and B are two $SVMNT's$ with different (p,q,r)-forms and $n_1 = n_2$, then we have a decision: $A <_S B$ or $A >_S B$ using steps 1-6 of the integrated S-Ranking algorithm for $SVNT's$ with different (p,q,r)-forms.*

Proof. If we do not have a decision using steps 1-6, then we have $(T_a, I_a.F_a) = (T'_a, I'_a, F'_a)$ and $W(A) = W(B)$. These two equalities give us the following equations:

$$\frac{p_1}{n_1} = \frac{p_2}{n_2} \quad (4)$$

$$\frac{q_1}{n_1} = \frac{q_2}{n_2} \quad (5)$$

$$\frac{r_1}{n_1} = \frac{r_2}{n_2} \quad (6)$$

But $n_1 = n_2$, so $p_1 = p_2$, $q_1 = q_2$ and $r_1 = r_2$. Which means that A and B of the same (p,q,r)-forms. A contradiction, hence we have a decision using Steps 1-6. □

In the following example we will show that the two $NVNT's$ A and B in Example 2.4 has different ranks using the integrated S-Ranking Algorithm for $SVMNT's$ with different (p,q,r)-forms.

Example 3.16. Let $A = (0.9, 0.9, 0.9, 0.9 \ ; \ 0.2 \ ; \ 0.1, 0.1)$ and $B = (0.9 \ ; \ 0.2, 0.2 \ ; \ 0.1, 0.1, 0.1, 0.1)$. We will rank A and B by the integrated S-Ranking Algorithm for $SVMNT's$. From (2) of Example 2.4 we have Steps 1-3 failed to give us a decision. So we compute the weighted triplet of A and B:

$$W(A) = (\frac{T_a \times p_1}{n_1}, \frac{I_a \times q_1}{n_1}, \frac{F_a \times r_1}{n_1}) = (\frac{0.9 \times 4}{7}, \frac{0.2 \times 1}{7}, \frac{0.1 \times 2}{7}) = (\frac{3.6}{7}, \frac{0.2}{7}, \frac{0.2}{7}).$$

$$W(B) = (\frac{T'_a \times p_2}{n_2}, \frac{I'_a \times q_2}{n_2}, \frac{F'_a \times r_2}{n_2}) = (\frac{0.9 \times 1}{7}, \frac{0.2 \times 2}{7}, \frac{0.1 \times 4}{7}) = (\frac{0.9}{7}, \frac{0.4}{7}, \frac{0.4}{7}).$$

Compute $s(W(A))$ and $s(W(B))$.

$$s(W(A)) = \frac{2 + T - I - F}{3} = \frac{2 + \frac{3.6}{7} - \frac{0.2}{7} - \frac{0.2}{7}}{3} \cong 0.82.$$

$$s(W(B)) = \frac{2 + T - I - F}{3} = \frac{2 + \frac{0.9}{7} - \frac{0.4}{7} - \frac{0.4}{7}}{3} \cong 0.67.$$

Since $s(W(A)) \cong 0.82 > s(W(B)) = 0.67$, we have a decision (Using Step 4): $A >_S B$. Note that in Example 3.8 we get a similar decision using the integrated N-Ranking Algorithm which is $A >_N B$.

In the following example steps 1-6 fail, but Step 7 success to give us a decision.

Example 3.17. Let $A = (0.3, 0.3 \ ; \ 0.6, 0.6, 0.6 \ ; \ 0.7)$ and
$B = (0.3, 0.3, 0.3, 0.3 \ ; \ 0.6, 0.6, 0.6, 0.6, 0.6, 0.6 \ ; \ 0.7, 0.7)$. Since $A_a = (T_a, I_a, F_a) = (0.3, 0.6, 0.7)$ $= (T'_a, I'_a, F'_a) = B_a$, Steps 1-3 fail to give us a decision. We go to the next step by computing the weighted triplet of A and B. But $W(A) = W(B) = (\frac{0.6}{6}, \frac{1.8}{6}, \frac{0.7}{6})$ (see Example 3.9), so Steps 4-6 fail, also, to give us a decision. Thus we go for the next step. Since

$$s(A_a) = \frac{2 + 0.3 - 0.6 - 0.7}{3} = 0.33... < 0.5$$

and $n_1 = 6 < n_2 = 12$, we have (using the second part of Step 7) $A >_S B$. We get the same decision $(A >_N B)$ using the Integrated N-Ranking Algorithm (see Example 3.9).

In the following example Steps 1-7 fail and Step 8 success to give us a decision.

Example 3.18. Let $A = (0.5, 0.5, 0.5 \ ; \ 0.4, 0.4 \ ; \ 0.6, 0.6)$ and
$B = (0.5, 0.5, 0.5, 0.5, 0.5, 0.5 \ ; \ 0.4, 0.4, 0.4, 0.4 \ ; \ 0.6, 0.6, 0.6, 0.6)$.
Since $A_a = (T_a, I_a, F_a) = (0.5, 0.4, 0.6) = (T'_a, I'_a, F'_a) = B_a$, Steps 1-3 fail to give us a decision. We go to the next step by computing the weighted triplet of A and B. But

$$W(A) = (\frac{T_a \times p_1}{n_1}, \frac{I_a \times q_1}{n_1}, \frac{F_a \times r_1}{n_1}) = (\frac{0.5 \times 3}{7}, \frac{0.4 \times 2}{7}, \frac{0.6 \times 2}{7}) = (\frac{1.5}{7}, \frac{0.8}{7}, \frac{0.6}{7}).$$
$$W(B) = (\frac{T'_a \times p_2}{n_2}, \frac{I'_a \times q_2}{n_2}, \frac{F'_a \times r_2}{n_2}) = (\frac{0.5 \times 6}{14}, \frac{0.4 \times 4}{14}, \frac{0.6 \times 4}{14}) = (\frac{1.5}{7}, \frac{0.8}{7}, \frac{0.6}{7}).$$

Since $W(B) = W(A)$, we have Steps 4-6 fail to give us a decision. We go for the next step. But $s(A_a) = \frac{2+T_a-F_a-I_a}{3} = \frac{2+0.5-0.6-0.4}{3} = \frac{1.5}{3} = 0.5$, so Steps 7 fails to give a decision. Therefore we go to Step 8. Since $a(A_a) = T - F = 0.5 - 0.6 = -1 < 0$ and $n_1 = 7 < 14 = n_2$, we have (Using Step 8) $A >_N B$. That is A is doing better than B.

Before we proceed we need the following proposition:

Proposition 3.19. *Let (T, I, F) be a neutrosophic triplet.*

1. *If $s(T, I, F) = 0.5$ and $a(T, I, F) = 0$, then $I = 0.5$ and $T = F$.*
2. *If $s(T, I, F) = 0.5$, $c(T, I, F) = 0.5$ and $a(T, I, F) = 0$, then $T = I = F = 0.5$.*

Proof. Obvious! we only need to solve simple linear equations. □

From Proposition 3.19 we conclude that: If we are looking for an example where steps 7 and 8 fail, then we have to have $I_a = 0.5$ and $T_a = F_a$. See the following example.

Example 3.20. Let $A = (0.4, 0.4, 0.4 \ ; \ 0.5, 0.5 \ ; \ 0.4, 0.4)$ and
$B = (0.4, 0.4, 0.4, 0.4, 0.4, 0.4 \ ; \ 0.5, 0.5, 0.5, 0.5 \ ; \ 0.4, 0.4, 0.4, 0.4)$.
Since $A_a = (T_a, I_a, F_a) = (0.4, 0.5, 0.4) = (T'_a, I'_a, F'_a) = B_a$, Steps 1-3 fail to give us a decision. We go to the next steps by computing the weighted triplet of A and B:

$$W(A) = (\frac{T_a \times p_1}{n_1}, \frac{I_a \times q_1}{n_1}, \frac{F_a \times r_1}{n_1}) = (\frac{0.4 \times 3}{7}, \frac{0.5 \times 2}{7}, \frac{0.4 \times 2}{7}) = (\frac{1.2}{7}, \frac{1}{7}, \frac{1.2}{7}) = W(B).$$

Since $W(A) = W(B)$, we have Steps 4-6 fail, also, to give us a decision. We go to Step 7. But $s(A_a) = \frac{2+0.4-0.5-0.4}{3} = 0.5$, so Step 7 fails to give us a decision. We go to Step 8. Since $a(A_a) = T_a - F_a = 0.4 - 0.4 = 0$, Step 8, also, fails. We go to Step 9. Since $c(A_a) = T_a = 0.4 < 0.5$ and $n_1 = 7 < n_2 = 14$, we have $A >_S B$ (by the second part of Step 9).

In the following example only Step 10 successes to give us a decision. From Proposition 3.19 the only such examples are those with $T = I = F = 0.5$.

Example 3.21. Let $A = (0.5, 0.5, 0.5 \ ; \ 0.5, 0.5 \ ; \ 0.5, 0.5)$ and
$B = (0.5, 0.5, 0.5, 0.5, 0.5, 0.5 \ ; \ 0.5, 0.5, 0.5, 0.5 \ ; \ 0.5, 0.5, 0.5, 0.5)$.
Since $A_a = (T_a, I_a, F_a) = (0.5, 0.5, 0.5) = (T'_a, I'_a, F'_a) = B_a$, Steps 1-4 fail to give us a decision. We go to the next step by computing the weighted triplet of A and B:

$$W(A) = (\frac{T_a \times p_1}{n_1}, \frac{I_a \times q_1}{n_1}, \frac{F_a \times r_1}{n_1}) = (\frac{0.5 \times 3}{7}, \frac{0.5 \times 2}{7}, \frac{0.5 \times 2}{7}) = (\frac{1.5}{7}, \frac{1}{7}, \frac{1}{7}) = W(B).$$

Since $W(A) = W(B)$, we have Steps 4-6 fail, also, to give us a decision. We go to Steps 7-9. But$s(A_a) = \frac{2+0.5-0.5-0,5}{3} = 0.5$, $a(A_a) = T - F = 0.5 - 0.5 = 0$ and $c(A_a) = 0.5$. Thus Steps 7-9 fail to give us a decision. We go to Step 11. Since $n_1 = 7 < n_2 = 14$, we have (by Step 10) $A <_N B$.

4 Summary

In n-valued neutrosophic sets the number of sources is essential, and must be considered in any ranking algorithm for $NVNT's$. There are two ranking algorithms of $NVNT's$:

1. The S-Ranking Algorithm of $SVMNT's$. Which is introduced by F.Smarandache in 2023.
2. The N-Ranking Algorithm of $SVNT's$. Which is introduced by V. L. Nayagam and Bharanidharan R.in 2023.

These algorithms do not consider the number of sources. We introduce two ranking algorithms for $SVNT's$:

1. The integrated S-Ranking Algorithm of $SVNT's$.
2. The integrated N-Nanking Algorithm of $SVNT's$.

These algorithms are improvements for the S-ranking algorithm of $SVMNT's$ and the N-Ranking Algorithm of $NVNT's$, respectively. The integrated ranking algorithms are total orderings of $NVNT's$, but the old ones are not.

Funding: The authors extend their appreciation to Prince Sattam bin Abdulaziz University for funding this research work through the project number (PSAU/2023/01/24751).

Conflicts of Interest: The author declare that he has no conflict of interest.

References

[1] K. Atanassov, *Intuitionistic fuzzy sets*, Fuzzy Sets and Systems, 20 (1986),87–96.

[2] V. L. G. Nayagam and R. Bharanidharan, *A Total order on single valued and interval valued neutrosophic triplets*, Neutrosophic Sets and Systems, 55 (2023), 2364 - 4974.

[3] V. L. G. Nayagam and R. Bharanidharan, *A Total Ordering on n - Valued Refined Neutrosophic Sets using Dictionary Ranking based on Total ordering on n - Valued Neutrosophic Tuplets*, Neutrosophic Sets and Systems, 58 (2023), 379-396.

[4] A. E. Samuel and R. Narmadhagnanam, *Neutrosophic Refined Sets in Medical Diagnosis*, International Journal of Fuzz Mathematical Archive, 14(1), 2017, 117-123.

[5] J. Dai, Z. Xu, and H. Liu, ”A novel approach to multi-attribute decision-making based on neutrosophic sets,” IEEE Transactions on Fuzzy Systems, vol. 29, no. 5, pp. 1208–1221, May 2021.

[6] M. Ali and F. Smarandache, ”Applications of neutrosophic logic in physics: A concise review,” Journal of Applied Mathematics and Physics, vol. 8, no. 10, pp. 2071–2080, Oct. 2020.

[7] F. Smarandache, *The Score, Accuracy, and Certainty Functions determine a Total Order on the Set of Neutrosophic Triplets (T,I,F)*, Neutrosophic Sets and Systems, Vol. 38, 2020.

[8] F. Smarandache, *Introduction to the MultiNeutrosophic Sets*, Neutrosophic Sets and Systems, Vol. 61, 2023.

[9] Zadeh, L.A. *Fuzzy sets*, Information and Control, 8 (1965), 338–353.