Neutrosophic Systems with Applications, Vol. 25, 2025
https://doi.org/10.61356/j.nswa.2025.25453

Some Graph Parameters for Superhypertree-width and Neutrosophictree-width

Takaaki Fujita [1,*] and Florentin Smarandache [2]

[1] Independent Researcher, Shinjuku, Shinjuku-ku, Tokyo, Japan; t171d603@gunma-u.ac.jp.
[2] Department of Mathematics & Sciences, University of New Mexico, Gallup, NM 87301, USA; smarand@unm.edu.

* Correspondence: t171d603@gunma-u.ac.jp.

Abstract: Graph characteristics are often studied through various parameters, with ongoing research dedicated to exploring these aspects. Among these, graph width parameters—such as treewidth—are particularly important due to their practical applications in algorithms and real-world problems. A hypergraph generalizes traditional graph theory by abstracting and extending its concepts [77]. More recently, the concept of a SuperHyperGraph has been introduced as a further generalization of the hypergraph. Neutrosophic logic [133], a mathematical framework, extends classical and fuzzy logic by allowing the simultaneous consideration of truth, indeterminacy, and falsity within an interval. In this paper, we explore Superhypertree-width, Neutrosophic treewidth, and t-Neutrosophic tree-width.

Keywords: Neutrosophic Graph; Hypertree Width; Superhypertree Width; Tree-width; Neutrosophictree-width.

1. Introduction

1.1 Graph Width Parameters

Graphs have been extensively researched in recent years [52], with particular emphasis on characterizing their properties. Graph characteristics are frequently analyzed using various parameters, and ongoing studies aim to deepen our understanding of these aspects. Analyzing graph parameters is essential for understanding the structural properties of graphs, which is crucial for solving computational problems in fields like network optimization, database theory, and machine learning applications (cf. [73]).

Among these parameters, graph width parameters such as tree-width [25,27,28,91,126–128], cut-width [85, 98], clique-width [45], modular-width [3], tree-cut-width [68, 99], boolean-width [4, 155], bandwidth [30, 44, 124], rank-width [93, 116, 118], and path-width [48, 94, 152] are particularly significant due to their practical applications in algorithms and real-world problems. This has fueled active research, focusing on how these parameters influence computational efficiency and problem-solving methods.

For instance, tree-width measures how closely a graph resembles a tree, while path-width shows how similar a graph is to a path. Graph width parameters often correspond to efficient, manageable graph structures, which is why they have been the focus of extensive research for many years (e.g., [79, 85, 124, 152–154]). Given this background, research on graph width parameters is of great importance.

1.2 Hypergraph and SuperHyperGraph

A hypergraph is a generalization of a conventional graph, extending concepts from graph theory [22,77]. A hypergraph is a generalization of a graph where edges, called hyperedges, can connect any number of vertices, not just two. Formally, a hypergraph is defined as H = (V, E), where V is the vertex set and E is the set of hyperedges. Hypergraphs are widely used in fields such as machine learning and network analysis [41, 69, 87, 100].

In many real-world applications, evaluating how closely a graph approximates a tree structure is essential. This has led to significant research on Hypertree-width [6, 72, 74, 103, 161] and Hyperpath-width [5, 105, 115], which quantify how much a hypergraph resembles a tree or a path. Hypertree width, in particular, plays a key role in database systems and practical applications [73, 74].

Recently, the concept of a SuperHyperGraph has emerged as a further generalization of the hypergraph, attracting significant research interest similar to that seen with hypergraphs [66, 82–84, 137–139, 143, 149]. It generalizes hypergraphs, allowing vertices to be individual elements or subsets (super-vertices), and edges to connect groups of vertices or super-vertices. This structure models complex relationships. Similar to hypergraphs, various applications are also being explored [61, 84].

1.3 Neutrosophic Graph and Neutrosophic HyperGraph

In any scientific field, a classical theorem defined within a specific space is a statement that holds true for 100 percent of the elements in that space. However, when applied to real-world scenarios, various constraints often arise, leading to the need for considering uncertain concepts, such as fuzzy sets [162–166], Vague set [29, 86, 167], plithogenic sets [1, 67, 136, 149], Rough sets [119–122], soft sets [106], Hypersoft set [141, 142], and neutrosophic sets [133–135, 146, 147] To address uncertainty and the relationships between concepts, several graph classes have been introduced, including Fuzzy Graphs [109, 125], Vague Graphs [131], and Rough Graphs [49]. Among these, this paper focuses on Neutrosophic Graphs [89].

A Neutrosophic Set generalizes the concept of fuzzy sets by introducing three membership functions: truth, indeterminacy, and falsity, allowing for a more nuanced representation of uncertainty [18, 150]. In recent years, Neutrosophic Graphs [11, 34, 76, 88, 130, 133, 140, 147, 148] and Neutrosophic Hypergraphs [8, 12, 102, 112] have been actively studied within Neutrosophic Set Theory. Neutrosophic refers to a mathematical framework that generalizes classical and fuzzy logic, simultaneously handling degrees of truth, indeterminacy, and falsity within an interval. These graphs, as generalizations of Fuzzy Graphs [110, 129], have garnered attention for their potential applications similar to those of Fuzzy Graphs. The concept of a Neutrosophic Super- hypergraph, which further generalizes Neutrosophic Hypergraphs, has also been the subject of active research in recent years [112, 144]. Due to the significance of Neutrosophic graphs, many other graph classes have also been proposed, such as those in [32, 32, 47, 58, 60, 65].

1.4 Our Contributions

Research on tree and path structures in Neutrosophic Graphs and SuperHyperGraphs is still in its early stages, and while several graph parameters have been proposed, there remains significant room for further exploration. In this context, [59] introduced the concept of SuperHyperTree-width.

We also explore Neutrosophic Graphs and Neutrosophic Hypergraphs. By applying treewidth and path width to these types of graphs, we aim to accelerate research and applications related to graph width parameters, as well as to Neutrosophic Graphs and SuperHyperGraphs. This paper is structured as follows: Section 2 provides definitions and examples of general graphs, fuzzy graphs, and hypergraphs. Section 3 reviews Neutrosophic Graphs and SuperHy- perGraphs, and defines and examines properties such as Neutrosophic tree-width, Neutrosophic hypertree-width, and n-Neutrosophic tree-width. Section 4 discusses future tasks.

2. Preliminaries and Definitions

In this section, we briefly explain the definitions and notations used in this paper.

2.1 Basic Graph Concepts

A graph G is a mathematical structure consisting of nodes (vertices) connected by edges, representing relationships or connections. In a graph G, $V(G)$ denotes the set of vertices, and $E(G)$ denotes the set of edges. The notation $G = (V, E)$ indicates that the graph G is defined by the pair of sets V (vertices) and E (edges).

In this paper, we provide several essential graph theory definitions and simple examples necessary for the discussions that follow.

Definition 2.1. A subgraph is formed by selecting specific vertices and edges from a graph.

Definition 2.2. A graph $G = (V, E)$ is called *connected* if, for every pair of vertices $u, v \in V$, there exists a path in G connecting u and v. In other words, a graph is connected if there is a sequence of edges that allows traversal between any two vertices in the graph. If no such path exists between certain vertices, the graph is called *disconnected*.

Definition 2.3. (cf. [42]) A path is a walk with no repeated vertices, a cycle is a closed path, and a tree is a connected acyclic graph.

Definition 2.4 (Tree). (cf. [158]) A tree is a connected, acyclic undirected graph. In other words, a tree is a graph where there is exactly one path between any two vertices, and no cycles exist. Additionally, a tree with n vertices has n − 1 edges.

Properties of Trees:

- There is exactly one path between any two vertices in a tree.
- A tree contains no cycles, and removing any edge from a tree will disconnect it.
- Adding an edge to a tree will create exactly one cycle.

For more basic graph notation and concepts, please refer to [19, 42, 52, 75].

2.2 Hypergraph Concepts

A hypergraph is a generalization of a graph where edges, called hyperedges, can connect any number of vertices, not just two. This structure is useful for modeling complex relationships in various fields like computer science and biology [56,71,123]. The definition is provided below.

Definition 2.5. [31] A hypergraph is a pair $H = (V(H), E(H))$, consisting of a nonempty set $V(H)$ of vertices and a set $E(H)$ of subsets of $V(H)$, called the hyperedges of H. In this paper, we consider only finite hypergraphs.

Example 2.6. Let H be a hypergraph with vertex set $V(H) = \{A, B, C, D, E\}$ and hyperedge set $E(H) = \{e_1, e_2, e_3\}$, where:

$e_1 = \{A, D\}, \quad e_2 = \{D, E\}, \quad e_3 = \{A, B, C\}.$

Thus, H is represented by the pair $H(V, E) = (\{A, B, C, D, E\}, \{\{A, D\}, \{D, E\}, \{A, B, C\}\})$.

Definition 2.7. [31] For a hypergraph H and a subset $X \subseteq V(H)$, the sub hypergraph induced by X is defined as $H[X] = (X, \{e \cap X \mid e \in E(H)\})$. We denote the hypergraph obtained by removing X from H as $H \setminus X := H[V(H) \setminus X]$.

Definition 2.8. A *hyperpath* in a hypergraph $H = (V(H), E(H))$ is a sequence of alternating vertices and hyperedges:

(v0, e1, v1, e2, v2, . . . , et, vt)

Such that:

- For each consecutive pair vi−1 and vi, there is a hyperedge ei ∈ E(H) where both vertices vi−1 and vi are contained in ei, i.e., vi−1, vi ∈ ei,
- vi−1 /= vi, meaning no vertex is repeated consecutively in the path.

A hyperpath is called a *simple hyperpath* if all vertices v_i and all hyperedges e_i are distinct.

Definition 2.9. (cf. [6,104]) A *hypertree* T is a connected hypergraph in which the removal of any hyperedge from T results in a disconnected hypergraph. Specifically:

- For any hyperedge e ∈ E(T), if e is removed, the resulting sub hypergraph is no longer connected.
- A hypertree can contain cycles, as long as removing any hyperedge disconnects the hypergraph.

Example 2.10. Let T be a hypergraph with vertex set $V(T) = \{A, B, C, D, E\}$ and hyperedge set $E(T) = \{e_1, e_2, e_3\}$, where:

$e_1 = \{A, B\}$, $e_2 = \{B, C, D\}$, $e_3 = \{D, E\}$.

In this case, T is a hypertree because removing any hyperedge e_1, e_2, or e_3 would disconnect the hypergraph.

For more basic hypergraph notation and concepts, please refer to [31,46].

2.3 Tree-width and Hypertree-width

We now define Tree-width and Hypertree-width. Tree-width measures how closely a graph resembles a tree by representing the graph in a tree-like structure with minimal width [25– 27, 126, 128]. Hypertree width generalizes tree width to hypergraphs, quantifying how well a hypergraph can be decomposed into a tree-like structure [6, 72, 74, 103, 161]. The formal definitions of Tree-width and Hypertree-width are provided below. For more details on Treewidth, please refer to [25,26].

Definition 2.11. [128] A tree-decomposition of an undirected graph G is a pair (T, W), where T is a tree, and $W = (W_t \mid t \in V(T))$ is a family of subsets that associates with every node t of T a subset W_t of vertices of G such that:

(T1) $\cup_{t\in V(T)} W_t = V(G)$,

(T2) For each edge $(u, v) \in E(G)$, there exists some node t of T such that $\{u, v\} \subseteq W_t$, and (T3) For all nodes r, s, t in T, if s is on the unique path from r to t then $W_r \cap W_t \subseteq W_s$.

The width of a tree decomposition (T, W) is the maximum of $|W_t| - 1$ over all nodes t of T. The tree-width of G is the minimum width over all tree-decompositions of G.

Example 2.12. Consider the following graph G:

$V(G) = \{v_1, v_2, v_3, v_4, v_5\}$

$E(G) = \{(v_1, v_2), (v_1, v_3), (v_2, v_4), (v_3, v_4), (v_4, v_5)\}$

We want to find a tree-decomposition for this graph G. Let T be a tree with three nodes t_1, t_2, t_3. We define the bags W_t as follows:

$W_{t_1} = \{v_1, v_2, v_3\}$, $W_{t_2} = \{v_2, v_3, v_4\}$, $W_{t_3} = \{v_4, v_5\}$

- *(T1):* The union of all bags covers all vertices of G:
  $W_{t_1} \cup W_{t_2} \cup W_{t_3} = \{v_1, v_2, v_3, v_4, v_5\} = V(G)$
- *(T2):* For each edge $(u, v) \in E(G)$, there exists a bag that contains both u and v:
  (v_1, v_2) is in W_{t_1},
  (v_1, v_3) is in W_{t_1},
  (v_2, v_4) is in W_{t_2},
  (v_3, v_4) is in W_{t_2},
  (v_4, v_5) is in W_{t_3}.
- *(T3):* For all nodes r, s, t in T, if s lies on the unique path from r to t, then:
  $W_r \cap W_t \subseteq W_s$

This is satisfied, as $W_{t_1} \cap W_{t_3} = \{v_4\} \subseteq W_{t_2}$.

The size of each bag is:

$|W_{t_1}| = 3$, $|W_{t_2}| = 3$, $|W_{t_3}| = 2$

The width of this tree decomposition is the maximum bag size minus 1:

$\text{width} = \max(3 - 1, 3 - 1, 2 - 1) = 2$

Definition 2.13. [6] A generalized hypertree decomposition of H is a triple (T, B, C), where (T, B) is a tree-decomposition of H and $C = (C_t)_{t \in V(T)}$ is a family of subsets of $E(H)$ such that for every $t \in V(T)$ we have $B_t \subseteq \cup C_t$. Here $\cup C_t$ denotes the union of the sets (hyperedges) in C_t, that is, the set $\{v \in V(H) \mid \exists e \in C_t : v \in e\}$. The sets C_t are called the guards of the decomposition. The width of the decomposition (T, B, C) is $\max\{|C_t| \mid t \in V(T)\}$. The generalized hypertree width of H, denoted by ghw(H), is the minimum of the widths of the generalized hypertree decompositions of H.

Definition 2.14. [6] A hypertree decomposition of H is a generalized hypertree decomposition (T, B, C) that satisfies the following special condition: $(\cup C_t) \cap \cup_{u \in V(T_t)} B_u \subseteq B_t$ for all $t \in V(T)$. Recall that T_t denotes the subtree of the T with root t. The hypertree width of H, denoted by hw(H), is the minimum of the widths of all hypertree decompositions of H.

2.4 Fuzzy Graph

A Fuzzy Graph represents relationships involving uncertainty by assigning membership degrees to both vertices and edges, allowing for more flexible and nuanced analysis. Due to its significance, Fuzzy Graphs have been the subject of extensive research [9, 10, 14–17, 23, 24, 92, 101,107,111,151,157]. The formal definition of a Fuzzy Graph is as follows [108,129].

Definition 2.15. [129] A fuzzy graph $G = (\sigma, \mu)$ with V as the underlying set is defined as follows:

- $\sigma : V \rightarrow [0, 1]$ is a fuzzy subset of vertices, where $\sigma(x)$ represents the membership degree of vertex $x \in V$.
- $\mu : V \times V \rightarrow [0, 1]$ is a fuzzy relation on σ, such that $\mu(x, y) \leq \sigma(x) \wedge \sigma(y)$ for all $x, y \in V$, where $\wedge$ denotes the minimum operation.

The underlying crisp graph of G is denoted by $G^* = (\sigma^*, \mu^*)$, where:

- $\sigma^* = \text{supp}(\sigma) = \{x \in V : \sigma(x) > 0\}$
- $\mu^* = \text{supp}(\mu) = \{(x, y) \in V \times V : \mu(x, y) > 0\}$

Definition 2.16. A fuzzy subgraph $H = (\sigma', \mu')$ of G is defined as follows:

- There exists $X \subseteq V$ such that $\sigma' : X \rightarrow [0, 1]$ is a fuzzy subset.
- $\mu' : X \times X \rightarrow [0, 1]$ is a fuzzy relation on σ', satisfying $\mu'(x, y) \leq \sigma'(x) \wedge \sigma'(y)$ for all $x, y \in X$.

Example 2.17. (cf. [35]) Consider a fuzzy graph $G = (\sigma, \mu)$ with four vertices $V = \{v_1, v_2, v_3, v_4\}$, as depicted in the diagram.

The membership degrees of the vertices are as follows:

$\sigma(v_1) = 0.1, \quad \sigma(v_2) = 0.3, \quad \sigma(v_3) = 0.2, \quad \sigma(v_4) = 0.4$

The fuzzy relation on the edges is defined by the values of μ, where $\mu(x, y) \leq \sigma(x) \wedge \sigma(y)$ for all $x, y \in V$. The fuzzy membership degrees of the edges are as follows:

$\mu(v_1, v_2) = 0.1, \ \mu(v_2, v_3) = 0.1, \ \mu(v_3, v_4) = 0.1$

$\mu(v_4, v_1) = 0.1, \ \mu(v_2, v_4) = 0.3$

In this case, the fuzzy graph G has the following properties:

- Vertices v_1, v_2, v_3, v_4 are connected by edges with varying membership degrees.
- The fuzzy relations ensure that $\mu(x, y)$ for any edge (x, y) does not exceed the minimum membership of the corresponding vertices.

In [62], Fuzzy Tree-width was defined as an extension of Tree-width for Fuzzy Graphs, serving as a graph width parameter. The definition is provided below.

Definition 2.18. [62] Let $G = (\sigma, \mu)$ be a fuzzy graph, where $\sigma : V \rightarrow [0, 1]$ is a fuzzy subset representing vertex membership, and $\mu : V \times V \rightarrow [0, 1]$ is a fuzzy relation on σ. A fuzzy tree-decomposition of G is a pair $(T, \{B_t\}_{t \in T})$, where:

- $T = (I, F)$ is a tree with nodes I and edges F.
- $\{B_t\}_{t \in T}$ is a collection of fuzzy subsets of V (called bags) associated with the nodes of

T such that:

i). For each vertex $v \in V$, the set $\{t \in I : v \in B_t\}$ is connected in the tree T.

ii). For each edge $(u, v) \in V \times V$ with membership degree $\mu(u, v) \leq \sigma(u) \wedge \sigma(v)$, there exists some node $t \in I$ such that both u and v are in B_t, and the membership degree of u and v in B_t is at least $\mu(u, v)$.

The width of a fuzzy tree-decomposition $(T, \{B_t\}_{t\in T})$ is defined as $\max_{t\in I}\left(\max_{v \in Bt} \mu(v, Bt) - 1\right)$,
Where $\mu(v, B_t)$ represents the maximum membership degree of vertex v in the fuzzy set B_t. The fuzzy treewidth of the fuzzy graph G is the minimum width among all possible fuzzy tree-decompositions of G.

Example 2.19. Definition of Fuzzy Graph: Consider the following Fuzzy Graph $G = (\sigma, \mu)$, which has vertices and edges defined as follows:

$V = \{v_1, v_2, v_3, v_4\}$

$\sigma(v_1) = 0.1, \quad \sigma(v_2) = 0.3, \quad \sigma(v_3) = 0.2, \quad \sigma(v_4) = 0.4$

$\mu(v_1, v_2) = 0.1,\ \mu(v_2, v_3) = 0.1,\ \mu(v_3, v_4) = 0.1$
$\mu(v_4, v_1) = 0.1,\ \mu(v_2, v_4) = 0.3$

Given this Fuzzy Graph G, we aim to construct a Fuzzy Tree-Decomposition $(T, \{B_t\}_{t\in T})$. We will set the following bags $\{B_t\}_{t\in T}$ and the tree T:

Bags:

B1 = {v1, v2}, B2 = {v2, v3, v4}, B3 = {v4, v1}

- B1 and B2 share the vertex v2 and are connected.
- B2 and B3 share the vertex v4 and are connected.

Verification of the Connectivity Condition:

- Vertex v1: Appears in B1 and B3, and these bags are connected in T .
- Vertex v2: Appears in B1 and B2, and these bags are connected.
- Vertex v3: Appears only in B2.
- Vertex v4: Appears in B2 and B3, and these bags are connected.

Thus, the connectivity condition is satisfied for all vertices.

Next, we verify the edge condition for each edge:

- For (v1, v2), v1 and v2 appear in B1, and μ(v1, v2) = 0.1 ≤ σ(v1) ∧ σ(v2) = 0.1.
- For (v2, v3), v2 and v3 appear in B2, and μ(v2, v3) = 0.1 ≤ σ(v2) ∧ σ(v3) = 0.1.
- For (v3, v4), v3 and v4 appear in B2, and μ(v3, v4) = 0.1 ≤ σ(v3) ∧ σ(v4) = 0.2.
- For (v4, v1), v4 and v1 appear in B3, and μ(v4, v1) = 0.1 ≤ σ(v4) ∧ σ(v1) = 0.1.
- For (v2, v4), v2 and v4 appear in B2, and μ(v2, v4) = 0.3 ≤ σ(v2) ∧ σ(v4) = 0.3.

Thus, the edge condition is satisfied for all edges.

The width of the Fuzzy Tree-Decomposition (T, {Bt}) is defined as: $\max_{t\in T}\left(\max_{v \in Bt} \mu(v, Bt) - 1\right)$

- For Bag B1 = {v1, v2}:
  μ(v1, v2) = 0.1, $\sup_{v\in B1}$ μ(v, B1) = 0.1

Width = 0.1 − 1 = −0.9

- For Bag B2 = {v2, v3, v4}:

max(μ(v2, v3), μ(v3, v4), μ(v2, v4)) = max(0.1, 0.1, 0.3) = 0.3

Width = 0.3 − 1 = −0.7

- For Bag B3 = {v4, v1}:

μ(v4, v1) = 0.1, $\sup_{v\in B3}$ μ(v, B3) = 0.1

Width = 0.1 − 1 = −0.9

max(−0.9, −0.7, −0.9) = −0.7

The width of this Fuzzy Tree-Decomposition is −0.7.

2.5 *Graph Parameter Hierarchy*

The Graph Parameter Hierarchy is frequently studied in research on graph parameters and algorithms. It describes the relationships between graph parameters based on their ability to upper or lower bound one another [63,78,132,156].

Definition 2.20. A graph parameter is a function f : G → R, where G represents the set of all undirected finite graphs, and the function returns a real number. We say that Parameter A upper bounds Parameter B if there exists a non-decreasing function fA,B such that

fA,B(A(G)) ≥ B(G)

for all graphs G. Conversely, if Parameter A does not upper bound Parameter B, we say that Parameter B is unbounded to Parameter A.

Lemma 2.21. [132] If a upper bounds b, and b upper bounds c, then a also upper bounds c. Proof. Refer to [132] for details as necessary.

3. Result of this Paper

In this section, we present the results of this paper. We focus on the study of Tree-width in the context of SuperHypergraphs and Neutrosophic graphs.

3.1 *SuperHyperTree-width*

A SuperHyperGraph is an advanced structure that extends hypergraphs by allowing both vertices and edges to be sets [64, 137, 138]. First, we will explain the definition of a SuperHy- perGraph. The definition is provided below.

Definition 3.1 (SuperHyperGraph (SHG))**.** [137, 138] A *SuperHyperGraph (SHG)* is an ordered pair SHG = (G, E), where:

(1) $G \subseteq P(V)$ is the set of vertices, and $E \subseteq P(V)$ is the set of edges, with V = {V1, V2, . . . , Vm}, where m ≥ 0.

(2) P (V) denotes the power set of V, i.e., all subsets of V . Each element in G is referred to as an SHG-vertex, which can take the following forms:
- A single vertex (as in classical graphs).
- A super-vertex, representing a subset of vertices (e.g., a group or organization).
- An indeterminate-vertex, representing an unclear or unknown entity.
- A null-vertex, represented by ∅, signifying a vertex without elements.

(3) E = {E1, E2, . . . , Ek}, with k ≥ 1, represents the family of SHG-edges. Each Ej ∈ P (V) is a subset of V, and SHG-edges can include:
- A single edge (classical edge), connecting two single vertices.
- A super-edge, connecting super-vertices.
- A hyper-super-edge, connecting three or more groups or organizations.
- A multi-edge connects multiple vertices or super-vertices simultaneously.
- An indeterminate edge represents unclear or unknown relationships.
- A null edge ∅represents no connection between the given vertices.

Definition 3.2 (Elements of a SuperHyperGraph)**.** [137, 138] A *SuperHyperGraph (SHG)* consists of the following elements:
- Single Vertices Vi: Individual vertices as in classical graphs, e.g., V1, V2.
- SuperVertices (SubsetVertices) SVi,j: Subsets of vertices, such as SV1,3 = {V1, V3}. SuperVertices can represent groups or organizations. For example:
  - SV1,23 = {V1, V23}, combining single vertices V1 and V23.
  - SV1,2,3 = {V1, V2, V3}, combining three single vertices.

- **NullVertex** $\emptyset_V$: A vertex with no elements.
- **Single Edges** $E_{i,j}$: Edges connecting two single vertices, e.g., $E_{1,5} = \{V_1, V_5\}, E_{2,3} = \{V_2, V_3\}$.
- **HyperEdges** HEi,j,k: Edges connecting three or more single vertices, e.g., HE1,4,6 = {V1, V4, V6}.
- **SuperEdges (SubsetEdges)** SE(i,j),(k,l): Edges connecting super-vertices, e.g., SE(1,3),(4,5) = {SV1,3, SV4,5}.
- **HyperSuperEdges (or HyperSubsetEdges)** $HSE_{i,j,k}$: Edges connecting three or more vertices, with at least one being a super-vertex. For example:
  $HSE_{1,45,23} = \{V_1, SV_{45}, SV_{23}\}$.
- **IndeterminateEdges** $IE_{x,y}$: Edges with unknown or unclear connections between vertices.
- **NullEdges** $\emptyset_E$: Edges representing no connection between vertices.

We consider the relationship between a Hypergraph and a SuperHypergraph. The following holds.

Theorem 3.3. Any SuperHypergraph can be reduced to a Hypergraph.

Proof. To map any SuperHypergraph to a Hypergraph, we need to handle both SuperVertices and SuperEdges from the SHG.

Each super-vertex in the SuperHypergraph corresponds to a set of vertices in a classical graph. Specifically, a super-vertex $SV_{i,j} \in G$ (e.g., $\{V_i, V_j\}$) can be represented by multiple individual vertices in the corresponding hypergraph. The vertex mapping is therefore straightforward: each element of the super-vertex (which is a set) will be treated as an individual vertex in the hypergraph. Formally:

- For each super-vertex SVi,j = {Vi, Vj} ∈ VSHG, create individual vertices Vi, Vj ∈ VH.
  SuperEdges or HyperSuperEdges in SHG, which may connect multiple super-vertices or sets, need to be flattened into simple hyperedges in the hypergraph. For each super-edge or hyper-super-edge $SE_{(i,j),(k,l)}$ connecting super-vertices $SV_{i,j}$ and $SV_{k,l}$, create a hyperedge in the hypergraph that connects all the corresponding individual vertices. Formally:
- For each super-edge SE(i,j),(k,l) ∈ ESHG, create a hyperedge eH ∈ EH, where eH = {Vi, Vj, Vk, Vl}.

The general rule is that any SuperEdge connecting multiple super-vertices in the SHG corresponds to a single hyperedge connecting the individual vertices (elements) of those super-vertices in the hypergraph.

Example 3.4. Consider the following SuperHypergraph SHG:

- Vertices: VSHG = {{V1, V2}, {V3}}
- Edges: ESHG = {{V1, V2}, {V1, V2, V3}}

To convert this into a Hypergraph:

(1) The super-vertices {V1, V2} and {V3} are flattened into individual vertices V1, V2, V3.
(2) The super-edge {V1, V2} becomes a hyperedge {V1, V2} in the hypergraph.
(3) The super-edge {V1, V2, V3} becomes a hyperedge {V1, V2, V3} in the hypergraph.

Thus, the resulting Hypergraph H is:

- Vertices: V (H) = {V1, V2, V3}
- Hyperedges: $E(H) = \{\{V_1, V_2\}, \{V_1, V_2, V_3\}\}$

Next, we consider the SuperHyperTree. The following will be written as definitions.

Definition 3.5 (SuperHyperTree (SHT))**.** A *SuperHyperTree (SHT)* is a specific type of SuperHyperGraph SHT = (V, E) that satisfies the following conditions:

(1) **Host Graph Condition:** There exists a host graph $T = (V, E_T)$, which is a tree, such that:
- T shares the same vertex set V as SHT.

- The edges in T represent the connections between vertices in V.

(2) **SuperHyperTree Condition:** Every hyperedge $E_i \in E$ of the SuperHyperGraph corresponds to a connected subtree of the host tree T. Specifically:

- If E_i is a single edge, it connects two vertices directly within T.
- If E_i is a super-edge (connecting subsets of vertices), the vertices in each subset must form a connected subtree in T.
- If E_i is a hyper-edge (connecting more than two vertices), all vertices in E_i must form a connected subtree in T.
- If E_i is an indeterminate edge, any realization of E_i must satisfy the condition that the vertices involved form a connected subtree in T.

(3) **Acyclic Condition:** The host graph T must be acyclic, which is a fundamental property of trees. Consequently, SHT inherits this acyclic nature through its hyperedges.

Key Properties of a SuperHyperTree:

- **Connectedness:** A SuperHyperTree is connected, meaning there exists a path be- tween any two vertices via a sequence of hyperedges.
- **No Cycles:** Since the host graph T is a tree, SHT does not contain any cycles, including those involving super-vertices or super-edges.
- **Generalization of Trees:** A SuperHyperTree extends the concept of a tree by al- lowing super-vertices and super-edges while maintaining the acyclic and connected properties of a tree.

Theorem 3.6. *A SuperHyperTree (SHT) is a generalization of a hypertree. Specifically, every hypertree can be represented as a SuperHyperTree, but not every SuperHyperTree is a hypertree.*

Proof. Let SHT = (V, E) be a SuperHyperTree and $HT = (V, E')$ be a hypertree.

1. SuperHyperTree as a Generalization of Hypertree. A hypertree is a connected hypergraph where removing any hyperedge disconnects the hypergraph. For a hypertree HT:
   - Each hyperedge $e \in E'(HT)$ connects a set of vertices that maintains the connectedness of the hypergraph.
   - Removing any hyperedge e results in a disconnected hypergraph.

In a SuperHyperTree SHT, the following conditions are satisfied:

- Host Graph Condition: SHT is embedded within a host tree T = (V, ET), ensuring an acyclic structure.
- SuperHyperTree Condition: Each hyperedge in E corresponds to a connected subtree of T .
- Acyclicity: SHT inherits the acyclic property from the host tree T .

Since hypertrees are connected and acyclic, any hypertree HT can be embedded as a SuperHyperTree SHT by constructing a host tree T such that each hyperedge $e \in E'(HT)$ corresponds to a connected subtree of T.

2. Differences Between SuperHyperTrees and Hypertrees. While hypertrees can contain cycles within hyperedges, SuperHyperTrees explicitly disallow cycles due to the acyclic nature of their host tree T . Furthermore, SuperHyperTrees allow for more complex structures such as super-edges (which connect subsets of vertices) and indeterminate edges, which are not defined in hypertrees.
3. Examples.
   - *Hypertree as a Special Case of SuperHyperTree:* Consider a hypertree $HT = (V, E')$. By embedding HT within a host tree T, where each hyperedge $e \in E'(HT)$ corresponds to a connected subtree of T, HT satisfies all the conditions of a SuperHyperTree SHT.

- *SuperHyperTree That Is Not a Hypertree:* Consider a SuperHyperTree with a super-edge that connects subsets of vertices. This structure does not satisfy the conditions of a hypertree because hypertrees do not allow for super-edges or indeterminate edges.

Thus, SuperHyperTrees generalize the concept of hypertrees by introducing super-edges, indeterminate edges, and the requirement of a host tree, while hypertrees are a specific subclass of SuperHyperTrees.

Corollary 3.7. The hypertree-width of a hypertree HT is equal to its treewidth, whereas the SuperHyperTree-width of a SuperHyperTree SHT may exceed its treewidth due to the inclusion of super-edges and indeterminate edges.

Proof. For Hypertrees: Since hypertrees can be decomposed into a tree structure where each hyperedge corresponds to a single bag in the tree decomposition, their hypertree-width coincides with their tree-width.

For SuperHyperTrees: In a SuperHyperTree SHT, the inclusion of super-edges and indeterminate edges increases the complexity of its decomposition. The SuperHyperTree-width accounts for these additional structures, leading to a possible increase in width compared to the tree-width.

Next, we define superhypertree-width by extending the concept of hypertree-width and tree-width(cf. [61]). Following that, we will briefly explore the Graph Parameter Hierarchy.

Definition 3.8 (SuperHyperTree Decomposition)**.** (cf. [61]) Let SHT = (V, E) be a SuperHyperGraph, where V is the set of vertices and E is the set of SuperEdges. A *SuperHyperTree Decomposition* of SHT is defined as a tuple (T, B, C), where:

- T = (VT , ET) is a tree.
- B = {Bt | t ∈ VT } is a family of subsets of V (called bags) associated with the nodes of T , satisfying:

(1) **Coverage Condition for SuperEdges:** For each SuperEdge e ∈ E, there exists a node t ∈ VT such that the entire SuperEdge e is contained in the corresponding bag Bt, i.e., e ⊆ Bt.

(2) **Vertex Connectivity Condition:** For each vertex v ∈ V , the set of nodes {t ∈ VT | v ∈ Bt} forms a connected subtree of T .

- C = {Ct | t ∈ VT } is a family of subsets of E (called guards) associated with the nodes of T , satisfying:

(1) **Guard Condition for SuperEdges:** For each t ∈ VT , Bt ⊆ ∪ Ct, where ∪ Ct denotes the union of all vertices in the SuperEdges of Ct, i.e., ∪ Ct = {v ∈ V | $\exists e \in C_t : v \in e$}.

(2) **SuperHyperTree Condition:** For each $t \in V_T$,

$$(\cup C_t) \cap \bigcup_{u \in V(T_t)} B_u \subseteq B_t,$$

where T_t is the subtree of T rooted at t.

Width of a SuperHyperTree Decomposition: The width of a SuperHyperTree Decomposition (T, B, C) is defined as:

$$\text{width}(T, B, C) = \max_{t \in V_T} |C_t|,$$

where $|C_t|$ is the cardinality of the guard C_t.

SuperHyperTree-width: The SuperHyperTree-width of the SuperHyperGraph SHT, denoted by SHT-width(SHT), is the minimum width over all possible SuperHyperTree Decom- positions:

$$\text{SHT-width(SHT)} = \min_{(T,B,C)} \text{width}(T, B, C).$$

- The SuperHyperTree-width measures how closely a SuperHyperGraph resembles a SuperHyperTree.

- For classical graphs, the SuperHyperTree-width coincides with the traditional treewidth.

SuperHyperPath Decomposition: A SuperHyperPath Decomposition is a path-based variant of the SuperHyperTree Decomposition, where the host structure T is a path instead of a tree.

Lemma 3.9. *Let SH be a SuperHypergraph. The following inequalities hold:*

$\mathrm{shw}(SH) \leq \mathrm{hw}(SH) \leq \mathrm{tw}(G_p(SH))+1.$

Where:

- shw(SH) denotes the SuperHyperTree-width of SH.
- hw(SH) denotes the hypertree-width of SH.
- tw(Gp(SH)) denotes the treewidth of the primal graph Gp(SH) of SH.

Proof. Since every SuperHyperTree decomposition is a special case of a hypertree decomposition:

$$\mathrm{shw(SH)} \leq \mathrm{hw(SH)}$$

Proof of hw(SH) ≤ tw(Gp(SH)) + 1:

From [74]:

$$\mathrm{hw(H)} \leq \mathrm{tw(Gp(H))} + 1$$

Applying this to SH:

$$\mathrm{hw(SH)} \leq \mathrm{tw(Gp(SH))} + 1 \ \mathrm{shw(SH)}$$

Conclusion:

$$\mathrm{hw(SH)} \leq \mathrm{hw(SH)} \leq \mathrm{tw(Gp(SH))} + 1$$

3.2 NeutrosophicTree-width

In this subsection, we consider the concept of neutrosophic tree-width.

First, we introduce the concept of a neutrosophic graph [11, 34, 76, 88, 130, 145]. A neutrosophic graph generalizes traditional graph theory by incorporating degrees of truth, indeterminacy, and falsity in its edges and vertices, enabling more complex and nuanced relationships. It extends the concept of a fuzzy graph [92]. Neutrosophic graph theory is particularly useful in uncertain environments, contributing to fields like human networks and decision-making [7,43]. Fuzzy graph theories have numerous applications in modern science and technology, especially in operations research, neural networks [2, 101], artificial intelligence [2, 101], and decision-making [92]. The definition is as follows.

Definition 3.10. (cf. [148]) A neutrosophic graph NTG = (V, E, σ = (σ1, σ2, σ3), μ = (μ1, μ2, μ3)) is defined as a graph where σi : V → [0, 1], μi : E → [0, 1], and for every vivj ∈ E, the following condition holds: μ(vivj) ≤ σ(vi) ∧ σ(vj).

(1) σ is called the neutrosophic vertex set.
(2) μ is called the neutrosophic edge set.
(3) $|V|$ is called the order of NTG, and it is denoted by $O(NTG)$.
(4) $\sum_{v\in V} \sigma(v)$ is called the neutrosophic order of NTG, and it is denoted by $On(NTG)$.
(5) $|E|$ is called the size of NTG, and it is denoted by $S(NTG)$.
(6) $\sum_{e\in E} \mu(e)$ is called the neutrosophic size of NTG, and it is denoted by $Sn(NTG)$.

Definition 3.11.

i). A sequence of vertices P : x0, x1, · · · , xn is called a path where xixi+1 ∈ E, i = 0, 1, · · · , n − 1.

ii). The strength of the path P : x0, x1, · · · , xn is /\i=0,··· ,n−1 μ(xixi+1).

iii). The connectedness between vertices x0 and xn is defined as:

μ∞(x, y) = /\ P :x0,x1,··· ,xn /\ i=0,··· ,n−1 μ(xixi+1).

The Examples of the neutrosophic graph are the following.

Example 3.12. (cf. [35]) Consider a neutrosophic graph NTG = (V, E, σ = (σ1, σ2, σ3), μ = (μ1, μ2, μ3)) with four vertices V = {v1, v2, v3, v4}, as shown in the diagram.

The neutrosophic membership degrees of the vertices are as follows:

σ(v1) = (0.5, 0.1, 0.4), σ(v2) = (0.6, 0.3, 0.2),
σ(v3) = (0.2, 0.3, 0.4), σ(v4) = (0.4, 0.2, 0.5)

The neutrosophic membership degrees of the edges are as follows:

μ(v1v2) = (0.2, 0.3, 0.4), μ(v2v3) = (0.3, 0.3, 0.4),
μ(v3v4) = (0.2, 0.3, 0.4), μ(v4v1) = (0.1, 0.2, 0.5)

In this case, the neutrosophic graph NTG has the following properties:

- Vertices v1, v2, v3, v4 are connected by edges with varying neutrosophic membership degrees.
- The neutrosophic relations ensure that for every edge vivj ∈ E, μ(vivj) ≤ σ(vi) ∧ σ(vj), where ∧ denotes the minimum operation.

Theorem 3.13. A Neutrosophic Graph can be transformed into a Fuzzy Graph by mapping the neutrosophic truth-membership values of vertices and edges directly to the fuzzy membership values, effectively disregarding the indeterminacy and falsity components.

Proof. Let NTG = (V, E, σ = (σ1, σ2, σ3), μ = (μ1, μ2, μ3)) be a neutrosophic graph. We aim to transform NTG into a fuzzy graph G = (V, σi, μi).

In a neutrosophic graph, the vertex membership σ(v) = (σ1(v), σ2(v), σ3(v)) includes truth, indeterminacy, and falsity components. To transform this into a fuzzy graph, we only retain the truth-membership component σ1(v). The transformed fuzzy vertex membership is thus:

σi(v) = σ1(v)

where σ1(v) ∈ [0, 1] represents the fuzzy membership degree of vertex v.

Similarly, for the edge set, the neutrosophic edge membership μ(e) = (μ1(e), μ2(e), μ3(e)) includes truth, indeterminacy, and falsity components. We retain only the truth-membership component μ1(e). The transformed fuzzy edge membership is thus:

μi(e) = μ1(e)

where μ1(e) ∈ [0, 1] represents the fuzzy membership degree of edge e.

To ensure that the transformed graph satisfies the properties of a fuzzy graph, we check the following conditions:

(1) Vertex Membership Condition: In the fuzzy graph G, the fuzzy vertex membership function σi(v) must satisfy σi(v) ∈ [0, 1]. Since σ1(v) ∈ [0, 1] in the neutrosophic graph, this condition holds automatically.

(2) Edge Membership Condition: In the fuzzy graph G, the fuzzy edge membership function μi(e) must satisfy μi(e) ≤ σi(vi) ∧ σi(vj) for all edges e = (vi, vj). In the neutrosophic graph, the truth-membership component of an edge μ1(e) satisfies the condition μ1(e) ≤ σ1(vi) ∧ σ1(vj), which is equivalent to the fuzzy graph condition.

This proof is completed.

Next, we consider a neutrosophic tree. The definition is similar to that of a general tree and a fuzzy tree. A neutrosophic tree is defined as follows:

Definition 3.14. (cf. [13, 70, 80]) A Neutrosophic Tree NT = (V, E, σ, μ) is a connected acyclic neutrosophic graph satisfying:

(1) **Connectedness**: For every pair of distinct vertices u, v ∈ V , there exists a unique path P from u to v such that for each edge e in P , μ(e) > 0.

(2) **Acyclicity**: The neutrosophic tree contains no cycles.

(3) **Neutrosophic Degree Condition**: For each vertex $v \in V$, the neutrosophic degree $\sigma(v)$ satisfies:

$$\sigma(v) = \sum_{u \in N(v)} \mu(vu),$$

Where $N(v)$ is the set of neighbors of v.

Based on the above definitions, we define the Neutrosophic Tree-width.

Definition 3.15. A **Neutrosophic Tree-Decomposition** of a neutrosophic graph $NG = (V, E, \sigma, \mu)$ is a pair (T, B) where:

- T = (VT , ET) is a tree.
- B = {Bt ∣ t ∈ VT } is a family of subsets of V (called bags), each associated with a node t of T .

This decomposition satisfies:

(1) **Coverage Condition**: For every edge e = (vi, vj) ∈ E, there exists a node t ∈ VT such that {vi, vj} ⊆ Bt.

(2) **Connectivity Condition**: For each vertex $v \in V$, the set $\{t \in V_T \mid v \in B_t\}$ forms a connected subtree of T.

The **width** of a Neutrosophic Tree-Decomposition $(T, \mathcal{B})$ is defined as:

$$\text{width}(T, \mathcal{B}) = \max_{t \in V_T} \left(\left\lceil \sum_{v \in B_t} \sigma(v) \right\rceil - 1 \right),$$

where $\lceil x \rceil$ denotes the ceiling function.

The **Neutrosophic Tree-Width** of the neutrosophic graph NG, denoted NTW(NG), is the minimum width over all possible Neutrosophic Tree-Decompositions of NG:

$$\text{NTW}(NG) = \min_{(T,\mathcal{B})} \text{width}(T, \mathcal{B}).$$

Example 3.16. Consider a neutrosophic graph NTG = (V, E, σ, μ) where the vertex set

V = {v1, v2, v3, v4} and the edge set E = {(v1, v2), (v2, v3), (v3, v4)}. The neutrosophic vertex membership functions σ = (σ1, σ2, σ3) and the neutrosophic edge membership functions μ = (μ1, μ2, μ3) are defined as follows:

- σ(v1) = (0.9, 0.1, 0.0), σ(v2) = (0.8, 0.2, 0.0)
- σ(v3) = (0.7, 0.3, 0.0), σ(v4) = (0.6, 0.4, 0.0)
- μ(v1v2) = (0.8, 0.1, 0.1), μ(v2v3) = (0.7, 0.2, 0.1), μ(v3v4) = (0.6, 0.3, 0.1)

We create a tree T = (VT , ET) where VT = {t1, t2, t3} and associate the following bags

B = {Bt1 , Bt2 , Bt3 }:

- Bt1 = {v1, v2}
- Bt2 = {v2, v3}
- Bt3 = {v3, v4}
- Coverage Condition: For each edge in E, there exists a corresponding bag:
  – (v1, v2) ∈ Bt1
  – (v2, v3) ∈ Bt2
  – (v3, v4) ∈ Bt3

For all edges (vi, vj), the condition μ(vivj) ≤ min(σ(vi), σ(vj)) is satisfied.

- Vertex Connectivity Condition: Each vertex forms a connected subtree of T :
  – v1 appears only in Bt1.
  – v2 appears in both Bt1 and Bt2 , and these bags are connected in T .
  – v3 appears in both Bt2 and Bt3 , and these bags are connected in T .
  – v4 appears only in Bt3.

For each bag Bt, we calculate the neutrosophic width:

- $\text{width}(B_{t_1}) = \sum_{v \in B_{t_1}} \sigma(v) - 1 = (0.9 + 0.8) - 1 = 1.7 - 1 = 0.7$
- $\text{width}(B_{t_2}) = \sum_{v \in B_{t_2}} \sigma(v) - 1 = (0.8 + 0.7) - 1 = 1.5 - 1 = 0.5$
- $\text{width}(B_{t_3}) = \sum_{v \in B_{t_3}} \sigma(v) - 1 = (0.7 + 0.6) - 1 = 1.3 - 1 = 0.3$

The maximum width is 0.7. Therefore, the Neutrosophic Tree-width of the graph is 0.7.

Theorem 3.17. The Neutrosophic Tree-Width NTW(NG) of a neutrosophic graph NG satisfies:

- If NG is a single vertex with σ(v) = 1, then NTW(NG) = 0.
- If NG consists of isolated vertices with σ(v) = 1 for all v ∈ V , then NTW(NG) = 0.

Proof. For a single vertex v with σ(v) = 1, construct a tree T with a single node t and set Bt = {v}. The width is:

width(T, B) = Iσ(vi)l − 1 = 1 − 1 = 0.

For multiple isolated vertices {v1, v2, . . . , vn} with σ(vi) = 1, create singleton bags Bt = {vi} in T. For each bag:

width(T, B) = Iσ(vi)l − 1 = 1 − 1 = 0.

Thus, NTW(NG) = 0

Theorem 3.18. The Neutrosophic Tree-width (NT-width) of an empty graph is −1.

Proof. In an empty graph (no vertices), there are no bags. By convention, we define:

NTW(NG) = −1.

Theorem 3.19. For any neutrosophic graph NG = (V, E, σ, μ), the Neutrosophic Tree-Width NTW(NG) satisfies:

NTW(NG) ≤ tw(G),

where tw(G) is the tree-width of the underlying simple graph G = (V, E).

Proof. Since σ(v) ∈ [0, 1], we have:

$$\sum_{v \in B_t} \sigma(v) \leq |B_t|.$$

Therefore,

$$\left\lceil \sum_{v \in B_t} \sigma(v) \right\rceil - 1 \leq |B_t| - 1.$$

Since $\text{tw}(G)$ is the minimum over all tree-decompositions, we have:

$$\text{NTW}(NG) \leq \text{tw}(G).$$

Theorem 3.20. If all vertices in NG have σ(v) = 1 and all edges have μ(e) = 1, then:

NTW(NG) = tw(G).

Proof. With $\sigma(v) = 1$ and $\mu(e) = 1$, NG behaves like G. For each bag B_t:

$$\sum_{v \in B_t} \sigma(v) = |B_t|.$$

Thus,

$$\text{width}(T, \mathcal{B}) = |B_t| - 1.$$

Therefore, $\text{NTW}(NG) = \text{tw}(G)$.

Therefore, NTW(NG) = tw(G).

Additionally, in the field of Neutrosophic Graphs, several classes such as Single Valued Neutrosophic Graphs [11, 53, 81, 113], Fermatean neutrosophic graphs [33], Single valued penpartitioned neutrosophic graphs [47], Interval Valued Neutrosophic Graphs [34,36,37,95,160] have been proposed. We plan to explore and characterize these classes in more detail in the future.

3.3 Neutrosophic HyperTree-width

Next, we introduce the concept of a Neutrosophic Hypergraph. Similar to Neutrosophic Graphs and Hypergraphs, Neutrosophic Hypergraphs have been the subject of extensive re- research [8, 51, 102]. The definition is provided below.

Definition 3.21. (cf. [8, 51, 102]) A **Neutrosophic Hypergraph** $NHG = (V, E, \sigma, \mu)$ consists of:

- A finite set V of vertices.
- A set $E \subseteq P(V) \setminus \{\varnothing\}$ of hyperedges.
- A neutrosophic vertex membership function $\sigma : V \to [0, 1]^3$, where for each $v \in V$:

$$\sigma(v) = (\sigma T(v), \sigma I(v), \sigma F(v)),$$

Representing the truth-membership, indeterminacy-membership, and falsity-membership degrees of v.

A neutrosophic hyperedge membership function $\mu : E \to [0, 1]3$, where for each $e \in E$:

$$\mu(e) = (\mu T(e), \mu I(e), \mu F(e)),$$

representing the truth-membership, indeterminacy-membership, and falsity-membership degrees of e.

These functions satisfy the following condition for every hyperedge $e \in E$:

$$\mu_T(e) \leq \min_{v\in e} \sigma_T(v), \quad \mu_I(e) \geq \max_{v\in e} \sigma_I(v), \quad \mu_F(e) \geq \max_{v\in e} \sigma_F(v).$$

Next, a Neutrosophic Hypertree is defined as follows.

Definition 3.22. A Neutrosophic Hypertree NHT = (V, E, σ, μ) is a connected, acyclic neutrosophic hypergraph satisfying:

(1) **Acyclicity**: The hypergraph contains no cycles. The incidence graph associated with NHT is acyclic.

(2) **Connectedness**: For every pair of distinct vertices $u, v \in V$, there exists a sequence of hyperedges $e_1, e_2, \ldots, e_k \in E$ such that:
- $u \in e1$ and $v \in ek$.
- $ei \cap ei+1$ /= ∅ for $i = 1, \ldots, k - 1$.
- $\mu T(ei) > 0$ for all i.

(3) **Neutrosophic Degree Condition**: For each $v \in V$:

$$\sum_{e\in E_v} \mu_T(e) = \sigma_T(v),$$

where $E_v = \{e \in E \mid v \in e\}$.

Based on the above, we define Neutrosophic HyperTree-decomposition as follows.

Definition 3.23. A **Neutrosophic HyperTree-Decomposition** of a neutrosophic hyper-graph $NHG = (V, E, \sigma, \mu)$ is a triple (T, B, C) where:

- $T = (VT, ET)$ is a tree.
- $B = \{Bt \subseteq V \mid t \in VT\}$ is a family of bags.
- $C = \{Ct \subseteq E \mid t \in VT\}$ is a family of guards.

This decomposition satisfies:

(1) **Coverage Condition**: For every hyperedge $e \in E$, there exists $t \in VT$ such that $e \subseteq Bt$ and $\mu T(e) \leq \min v{\in}e\ \sigma T(v)$.

(2) **Vertex Connectivity Condition**: For each $v \in V$, the set $\{t \in VT \mid v \in Bt\}$ forms a connected subtree of T.

(3) **Guard Condition:** For each $t \in V_T,\ B_t \subseteq \bigcup_{e \in C_t} e.$

The **width** of a Neutrosophic HyperTree-Decomposition $(T, \mathcal{B}, \mathcal{C})$ is defined as:

$$\text{width}(T, \mathcal{B}, \mathcal{C}) = \max_{t \in V_T} \left(\sum_{e \in C_t} \mu_T(e) \right)$$

The **Neutrosophic HyperTree-Width** of NHG, denoted NHTW(NHG), is:

$$\text{NHTW}(NHG) = \min_{(T, \mathcal{B}, \mathcal{C})} \text{width}(T, \mathcal{B}, \mathcal{C}).$$

Theorem 3.24. For any neutrosophic hypergraph NHG:

$$\text{NHTW(NHG)} \leq \text{HTW(NHG)},$$

where HTW(NHG) is the standard hypertree-width.

Proof. Since $\mu_T(e) \leq 1$, we have:

$$\sum_{e \in C_t} \mu_T(e) \leq \sum_{e \in C_t} 1 = |C_t|.$$

Thus,

$$\text{width}_{NHT}(T, B, C) \leq \text{width}_{HT}(T, B, C).$$

Taking the minimum over all decompositions:

$$\text{NHTW}(NHG) \leq \text{HTW}(NHG).$$

Theorem 3.25. *If $\sigma_T(v) = 1$ for all $v \in V$ and $\mu_T(e) = 1$ for all $e \in E$, then:*

$$NHTW(NHG) = HTW(NHG).$$

Proof. Under these conditions, the width becomes:

$$\text{width}(T, B, C) = \max_{t \in V_T} |C_t|.$$

Therefore:

$$\text{NHTW}(NHG) = \text{HTW}(NHG).$$

Theorem 3.26. *For any neutrosophic hypergraph NHG:*

$$NHTW(NHG) \leq NTW(G),$$

where NTW(G) is the Neutrosophic Tree-Width of the incidence graph G of NHG.

Proof. By constructing a corresponding Neutrosophic Tree-Decomposition of G, we establish:

$$\text{NHTW}(NHG) \leq \text{NTW}(G).$$

Theorem 3.27. *For any neutrosophic hypergraph NHG:*

$$NHTW(NHG) \leq tw(G) + 1,$$

where tw(G) is the tree-width of the primal graph G of NHG.

Proof. Using the inequality $\text{HTW}(NHG) \leq \text{tw}(G) + 1$ and Theorem 1:

$$\text{NHTW}(NHG) \leq \text{HTW}(NHG) \leq \text{tw}(G) + 1.$$

We intend to further examine the validity of the above definitions.

3.4 Definition of t-NeutrosophicTree-width

The t-neutrosophic approach links different values of the parameter "t" to various layers of the graph, allowing for multi-level analysis. This method enables a detailed exploration of the relationships within the graph, incorporating varying degrees of confidence. As a result, it provides a more nuanced understanding of the underlying structure [96,97].

Definition 3.28. [96, 97] Let G be a Neutrosophic Set (NS) over a universal set U with $t \in [0, 1]$. The t-Neutrosophic Graph NSG_t of U, also known as a t-Neutrosophic Set (t-NS), is defined for each $u_1 \in U$ as follows:

$T_{G_t}(u_1) = \min\{T_G(u_1), t\}$, $I_{G_t}(u_1) = \max\{I_G(u_1), 1 - t\}$, $F_{G_t}(u_1) = \max\{F_G(u_1), 1 - t\}$,

where T_G, I_G, and F_G represent the truth-membership, indeterminacy-membership, and falsity-membership functions, respectively. The t-Neutrosophic Set can then be represented as:

$$G_t = \{u_1, T_G(u_1), I_G(u_1), F_G(u_1) \mid u_1 \in U\}.$$

Furthermore, the membership functions satisfy the following conditions:

$$0 \leq T_G(u_1) + I_G(u_1) + F_G(u_1) \leq 1.$$

Definition 3.29. [96, 97] Let $G = (V, E)$ be a simple graph where V is the set of vertices and $E \subseteq V \times V$ is the set of edges. A t-Neutrosophic Graph G_t is represented as:

$$G_t = (A_t, B_t),$$

where A_t is a t-Neutrosophic Set on the vertices V and B_t is a t-Neutrosophic Set on the edges E.

- $A_t = \{(u_i, T_G(u_i), I_G(u_i), F_G(u_i)) \mid u_i \in V\}$ represents the t-Neutrosophic Set on the vertex set V .
- B_t = {((ui, uj), TG(ui, uj), IG(ui, uj), FG(ui, uj)) | (ui, uj) ∈ E} represents the t-Neutrosophic Set on the edge set E ⊆ V × V .

For each edge $(u_i, u_j) \in E$, the following conditions hold:

$$T_{B_t}(u_i, u_j) \leq \min\{T_{A_t}(u_i), T_{A_t}(u_j)\},\ I_{B_t}(u_i, u_j) \leq \max\{I_{A_t}(u_i), I_{A_t}(u_j)\},\ F_{B_t}(u_i, u_j) \leq \max\{F_{A_t}(u_i), F_{A_t}(u_j)\}.$$

The t-Neutrosophic Graph also satisfies the following conditions for both vertices and edges:

$$0 \leq T_{A_t}(u_i) + I_{A_t}(u_i) + F_{A_t}(u_i) \leq 1, \text{ for all } u_i \in V,$$

$$0 \leq T_{B_t}(u_i, u_j) + I_{B_t}(u_i, u_j) + F_{B_t}(u_i, u_j) \leq 1, \text{ for all } (u_i, u_j) \in E.$$

The t-Neutrosophic Tree and t-Neutrosophic Tree-width extend classical tree and tree-width concepts to account for uncertainty, truth, indeterminacy, and falseness. The threshold parameter t allows for varying levels of confidence in the analysis of graph structures. The definition of a *t-Neutrosophic Tree* is as follows.

Definition 3.30. A t-Neutrosophic Tree (t-NTT) is a connected, acyclic t-Neutrosophic graph Gt = (V, E) that satisfies:

- Acyclicity: The graph contains no cycles, meaning for every pair of distinct vertices u, v ∈ V , there is exactly one path connecting them with no repeated vertices.
- Connectedness: For every pair of distinct vertices u, v ∈ V , there exists a path P ⊆ Gt connecting u and v.

Additionally, the neutrosophic membership conditions hold for all vertices and edges as defined in the t-Neutrosophic Graph framework.

The t-Neutrosophic tree width measures how closely the graph resembles a tree in the neutrosophic framework.

Definition 3.31. A t-Neutrosophic Tree-decomposition of a t-Neutrosophic graph Gt = (V, E) is a pair (T, B), where:

- T = (VT , ET) is a tree.
- B = {Bt | t ∈ VT } is a collection of subsets (called bags) of vertices from Gt, satisfying:
  (1) For every edge (u, v) ∈ E, there exists a bag Bt ∈ B such that {u, v} ⊆ Bt.
  (2) For each vertex u ∈ V , the set {t ∈ VT | u ∈ Bt} forms a connected subtree of T .

Definition 3.32. The *width* of a t-Neutrosophic Tree-decomposition is defined as:

$$\text{width}(T, \mathcal{B}) = \max_{t \in V_T} \left(\sum_{v \in B_t} \sigma(v) - 1 \right),$$

Where $\sigma(v)$ is the neutrosophic degree of vertex v. The *t-Neutrosophic Tree-width* of a *t*-Neutrosophic graph G_t, denoted by tNT-width(G_t), is the minimum width over all possible *t*-Neutrosophic Tree-decompositions of G_t:

$$\text{t-NTT-width}(G_t) = \min_{(T,\mathcal{B})} \text{width}(T, \mathcal{B}).$$

The following theorem holds.

Theorem 3.33. Let t-NTT = (Vt, Et) be a t-Neutrosophic Tree, where t ∈ [0, 1]. The t- Neutrosophic Tree-width of t-NTT is less than or equal to the Neutrosophic Tree-width of t-NTT, that is,

t-NTT-width(t-NTT) ≤ NTT-width(t-NTT).

Proof. We prove this theorem by considering the role of the parameter t.

Case 1: t = 1

When t = 1, the t-Neutrosophic Tree is identical to a standard Neutrosophic Tree, and their widths are equal:

t-NTT-width(t-NTT) = NTT-width(t-NTT).

Case 2: t < 1

When t < 1, the t-Neutrosophic Tree incorporates the threshold t, imposing constraints on the relationships between vertices and edges. This results in fewer edges being considered in the decomposition.

Let B_t be the bag in a *t*-Neutrosophic Tree decomposition, and B^* the corresponding bag in a Neutrosophic Tree decomposition (without the threshold *t*). The difference between B_t and B_t^* arises because *t* restricts certain vertices and edges based on neutrosophic membership values.

We have:

$$|B_t| \leq |B_t{}^*| \qquad \forall t \in V_T,$$

Where B_t^* includes all vertices and edges without the threshold restriction. Therefore, the *t*-neutrosophic tree decomposition bags are equal to or smaller than those in the Neutrosophic Tree decomposition.

Thus, the width of the *t*-Neutrosophic Tree decomposition is less than or equal to the Neutrosophic Tree decomposition, leading to:

t-NTT-width(t-NTT) ≤ NTT-width(t-NTT).

This proof is completed.

4. Conclusion and Future Research Goals

We intend to explore the relationship between SuperHyperTree decomposition, Neutrosophic HyperTree decomposition, and Neutrosophic Tree decomposition with other graph width parameters.

In classical graph theory, various width parameters such as boolean width [4, 21, 38–40], modular width [3], clique-width [55,90], and rank width [57,99,116–118] have been extensively explored. Our research aims to investigate whether any new characteristics arise when these concepts are extended to Neutrosophic graphs and SuperHypergraphs.

Additionally, we are interested in extending fundamental concepts like tree depth [50, 114, 159], tree length [20, 54], and tree breadth to fuzzy hypergraphs and fuzzy directed graphs, to uncover any unique properties or emerging behaviors.

Declarations

Ethics Approval and Consent to Participate

The results/data/figures in this manuscript have not been published elsewhere, nor are they under consideration by another publisher. All the material is owned by the authors, and/or no permissions are required.

Consent for Publication

This article does not contain any studies with human participants or animals performed by any of the authors.

Availability of Data and Materials

The data that support the findings of this study are available from the corresponding author upon reasonable request.

Competing Interests

The authors declare no competing interests in the research.

Funding

This research was not supported by any funding agency or institute.

Author Contribution

All authors contributed equally to this research.

Acknowledgment

The author is grateful to the editorial and reviewers, as well as the correspondent author, who offered assistance in the form of advice, assessment, and checking during the study period.

References

1. Mohamed Abdel-Basset, Mohamed El-Hoseny, Abduallah Gamal, and Florentin Smarandache. A novel model for evaluation hospital medical care systems based on plithogenic sets. *Artificial intelligence in medicine*, 100:101710, 2019.
2. Shigeo Abe. Neural networks and fuzzy systems: theory and applications. Springer Science & Business Media, 2012.
3. Faisal N Abu-Khzam, Shouwei Li, Christine Markarian, Friedhelm Meyer auf der Heide, and Pavel Podlipyan. Modular-width: An auxiliary parameter for parameterized parallel complexity. In *Frontiers in Algorithmics: 11th International Workshop, FAW 2017, Chengdu, China, June 23-25, 2017, Proceedings 11*, pages 139–150. Springer, 2017.
4. Isolde Adler, Binh-Minh Bui-Xuan, Yuri Rabinovich, Gabriel Renault, Jan Arne Telle, and Martin Vatshelle. On the boolean-width of a graph: Structure and applications. In *International Workshop on Graph-Theoretic Concepts in Computer Science*, pages 159–170. Springer, 2010.
5. Isolde Adler, Tomáš Gavenčiak, and Tereza Klimošová. Hypertree-depth and minors in hypergraphs. *Theoretical Computer Science*, 463:84–95, 2012.
6. Isolde Adler, Georg Gottlob, and Martin Grohe. Hypertree width and related hypergraph invariants. *European Journal of Combinatorics*, 28(8):2167–2181, 2007.
7. D. Ajay, S. John Borg, and P. Chellamani. Domination in pythagorean neutrosophic graphs with an application in fuzzy intelligent decision making. In *International Conference on Intelligent and Fuzzy Systems*, pages 667–675, Cham, July 2022. Springer International Publishing.
8. Muhammad Akram and Anam Luqman. Intuitionistic single-valued neutrosophic hypergraphs. *Opsearch*, 54:799–815, 2017.
9. Muhammad Akram and Anam Luqman. *Fuzzy hypergraphs and related extensions*. Springer, 2020.
10. Muhammad Akram, Danish Saleem, and Talal Al-Hawary. Spherical fuzzy graphs with application to decision-making. *Mathematical and Computational Applications*, 25(1):8, 2020.
11. Muhammad Akram and Gulfam Shahzadi. *Operations on single-valued neutrosophic graphs*. Infinite Study, 2017.
12. Muhammad Akram and KP Shum. *Bipolar neutrosophic planar graphs*. Infinite Study, 2017.

13. Muhammad Akram and Muzzamal Sitara. Novel applications of single-valued neutrosophic graph structures in decision-making. *Journal of Applied Mathematics and Computing*, 56:501–532, 2018.
14. T Al-Hawary, S Al-Shalaldeh, and MUHAMMAD Akram. Certain matrices and energies of fuzzy graphs. *TWMS Journal of Applied and Engineering Mathematics*, 2:2146–1147, 2021.
15. Talal Al-Hawary and Laith AlMomani. -balanced fuzzy graphs. *arXiv preprint arXiv:1804.08677*, 2018.
16. Talal Ali Al-Hawary. Certain classes of fuzzy graphs. *European Journal of Pure and Applied Mathematics*, 10(3):552–560, 2017.
17. TALAL ALI AL-HAWARY and MAREF YM ALZOUBI. β-product of product fuzzy graphs. *Journal of applied mathematics & informatics*, 42(2):283–290, 2024.
18. Shawkat Alkhazaleh. Neutrosophic vague set theory. *Critical Review*, 10:29–39, 2015.
19. Rangaswami Balakrishnan and Kanna Ranganathan. *A textbook of graph theory*. Springer Science & Business Media, 2012.
20. Rémy Belmonte, Fedor V Fomin, Petr A Golovach, and MS Ramanujan. Metric dimension of bounded tree-length graphs. *SIAM Journal on Discrete Mathematics*, 31(2):1217–1243, 2017.
21. Rémy Belmonte and Martin Vatshelle. On graph classes with logarithmic boolean-width. *arXiv preprint arXiv:1009.0216*, 2010.
22. Claude Berge. *Hypergraphs: combinatorics of finite sets*, volume 45. Elsevier, 1984.
23. Leonid S Bershtein and Alexander V Bozhenyuk. Fuzzy graphs and fuzzy hypergraphs. In *Encyclopedia of Artificial Intelligence*, pages 704–709. IGI Global, 2009.
24. Prabir Bhattacharya. Some remarks on fuzzy graphs. *Pattern recognition letters*, 6(5):297–302, 1987.
25. Hans L Bodlaender. A tourist guide through treewidth. *Acta cybernetica*, 11(1-2):1–21, 1993.
26. Hans L Bodlaender. A partial k-arboretum of graphs with bounded treewidth. *Theoretical computer sci- ence*, 209(1-2):1–45, 1998.
27. Hans L Bodlaender, Michael R Fellows, and Michael T Hallett. Beyond np-completeness for problems of bounded width (extended abstract) hardness for the w hierarchy. In *Proceedings of the twenty-sixth annual ACM symposium on Theory of computing*, pages 449–458, 1994.
28. Hans L Bodlaender and Arie MCA Koster. Treewidth computations i. upper bounds. *Information and Computation*, 208(3):259–275, 2010.
29. Zbigniew Bonikowski and Urszula Wybraniec-Skardowska. Rough sets and vague sets. In Rough Sets and Intelligent Systems Paradigms: International Conference, RSEISP 2007, Warsaw, Poland, June 28-30, 2007. Proceedings 1, pages 122–132. Springer, 2007.
30. Julia Böttcher, Klaas P Pruessmann, Anusch Taraz, and Andreas Würfl. Bandwidth, treewidth, separators, expansion, and universality. *Electronic Notes in Discrete Mathematics*, 31:91–96, 2008.
31. Alain Bretto. Hypergraph theory. An introduction. Mathematical Engineering. Cham: Springer, 1, 2013.
32. Said Broumi, D Ajay, P Chellamani, Lathamaheswari Malayalan, Mohamed Talea, Assia Bakali, Philippe Schweizer, and Saeid Jafari. Interval valued pentapartitioned neutrosophic graphs with an application to mcdm. *Operational Research in Engineering Sciences: Theory and Applications*, 5(3):68–91, 2022.
33. Said Broumi, R Sundareswaran, M Shanmugapriya, Assia Bakali, and Mohamed Talea. Theory and applications of fermatean neutrosophic graphs. *Neutrosophic sets and systems*, 50:248–286, 2022.
34. Said Broumi, Mohamed Talea, Assia Bakali, and Florentin Smarandache. Interval valued neutrosophic graphs. *Critical Review, XII*, 2016:5–33, 2016.
35. Said Broumi, Mohamed Talea, Assia Bakali, and Florentin Smarandache. Single valued neutrosophic graphs. *Journal of New theory*, (10):86–101, 2016.
36. Said Broumi, Mohamed Talea, Assia Bakali, Florentin Smarandache, Quek Shio Gai, and Ganeshsree Selvachandran. Introduction of some new results on interval-valued neutrosophic graphs. *Journal of Information and Optimization Sciences*, 40(7):1475–1498, 2019.
37. Said Broumi, Mohamed Talea, Assia Bakali, Florentin Smarandache, and PK Kishore Kumar. Shortest path problem on single valued neutrosophic graphs. In *2017 international symposium on networks, computers and communications (ISNCC)*, pages 1–6. IEEE, 2017.
38. Binh-Minh Bui-Xuan, Jan Arne Telle, and Martin Vatshelle. Boolean-width of graphs. *Theoretical Computer Science*, 412(39):5187–5204, 2011.

39. Binh-Minh Bui-Xuan, Jan Arne Telle, and Martin Vatshelle. Boolean-width of graphs. *Theoretical Computer Science*, 412(39):5187–5204, 2011.
40. BM Bui-Xuan, JA Telle, and M Vatshelle. Fast algorithms for vertex subset and vertex partitioning problems on graphs of low boolean-width. *submitted to IWPEC*, 2009.
41. Derun Cai, Moxian Song, Chenxi Sun, Baofeng Zhang, Shenda Hong, and Hongyan Li. Hypergraph structure learning for hypergraph neural networks. In *IJCAI*, pages 1923–1929, 2022.
42. Gary Chartrand. *Introductory graph theory*. Courier Corporation, 2012.
43. P Chellamani, D Ajay, Mohammed M Al-Shamiri, and Rashad Ismail. *Pythagorean Neutrosophic Planar Graphs with an Application in Decision-Making*. Infinite Study, 2023.
44. Phyllis Z Chinn, Jarmila Chvátalová, Alexander K Dewdney, and Norman E Gibbs. The bandwidth problem for graphs and matrices—a survey. *Journal of Graph Theory*, 6(3):223–254, 1982.
45. Bruno Courcelle and Stephan Olariu. Upper bounds to the clique width of graphs. *Discrete Applied Mathematics*, 101(1-3):77–114, 2000.
46. Qionghai Dai and Yue Gao. Mathematical foundations of hypergraph. In *Hypergraph Computation*, pages 19–40. Springer, 2023.
47. Suman Das, Rakhal Das, and Surapati Pramanik. Single valued pentapartitioned neutrosophic graphs. *Neutrosophic Sets and Systems*, 50(1):225–238, 2022.
48. Dariusz Dereniowski. From pathwidth to connected pathwidth. *SIAM Journal on Discrete Mathematics*, 26(4):1709–1732, 2012.
49. R Aruna Devi and K Anitha. Construction of rough graph to handle uncertain pattern from an information system. *arXiv preprint arXiv:2205.10127*, 2022.
50. Matt DeVos, O-joung Kwon, and Sang-il Oum. Branch-depth: Generalizing tree-depth of graphs. *European Journal of Combinatorics*, 90:103186, 2020.
51. PM Dhanya and PB Ramkumar. Text analysis using morphological operations on a neutrosophic text hypergraph. *Neutrosophic Sets and Systems*, 61:337–364, 2023.
52. Reinhard Diestel. *Graph theory*. Springer (print edition); Reinhard Diestel (eBooks), 2024.
53. Juanjuan Ding, Wenhui Bai, and Chao Zhang. A new multi-attribute decision making method with single-valued neutrosophic graphs. *International Journal of Neutrosophic Science*, 11(2):76–86, 2020.
54. Yon Dourisboure, Feodor F Dragan, Cyril Gavoille, and Chenyu Yan. Spanners for bounded tree-length graphs. *Theoretical Computer Science*, 383(1):34–44, 2007.
55. Michael R. Fellows, Frances A. Rosamond, Ulrike Rotics, and Stefan Szeider. Clique-width is np-complete. *SIAM Journal on Discrete Mathematics*, 23(2):909–939, 2009.
56. Song Feng, Emily Heath, Brett Jefferson, Cliff Joslyn, Henry Kvinge, Hugh D Mitchell, Brenda Praggastis, Amie J Eisfeld, Amy C Sims, Larissa B Thackray, et al. Hypergraph models of biological networks to identify genes critical to pathogenic viral response. *BMC bioinformatics*, 22(1):287, 2021.
57. Fedor V Fomin, Sang-il Oum, and Dimitrios M Thilikos. Rank-width and tree-width of h-minor-free graphs. *European Journal of Combinatorics*, 31(7):1617–1628, 2010.
58. Takaaki Fujita. Note for neutrosophic incidence and threshold graph. *SciNexuses*, 1:97–125, 2024.
59. Takaaki Fujita. Obstruction for hypertree width and superhypertree width. *preprint(researchgate)*, 2024.
60. Takaaki Fujita. A review of the hierarchy of plithogenic, neutrosophic, and fuzzy graphs: Survey and applications. *ResearchGate(Preprint)*, 2024.
61. Takaaki Fujita. Short note of supertree-width and n-superhypertree-width. *Neutrosophic Sets and Systems*, 77:54–78, 2024.
62. Takaaki Fujita. Tree-decomposition on fuzzy graph. *preprint(researchgate)*, 2024.
63. Takaaki Fujita. Various properties of various ultrafilters, various graph width parameters, and various connectivity systems. *arXiv preprint arXiv:2408.02299*, 2024.
64. Takaaki Fujita. Fundamental computational problems and algorithms for superhypergraphs. 2025.
65. Takaaki Fujita and Florentin Smarandache. Antipodal turiyam neutrosophic graphs. *Neutrosophic Optimization and Intelligent Systems*, 5:1–13, 2024.
66. Takaaki Fujita and Florentin Smarandache. A concise study of some superhypergraph classes. *Neutrosophic Sets and Systems*, 77:548–593, 2024.

67. Takaaki Fujita and Florentin Smarandache. Study for general plithogenic soft expert graphs. *Plithogenic Logic and Computation*, 2:107–121, 2024.
68. Robert Ganian, Eun Jung Kim, and Stefan Szeider. Algorithmic applications of tree-cut width. In *International Symposium on Mathematical Foundations of Computer Science*, pages 348–360. Springer, 2015.
69. Yue Gao, Yifan Feng, Shuyi Ji, and Rongrong Ji. Hgnn+: General hypergraph neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(3):3181–3199, 2022.
70. Masoud Ghods and Zahra Rostami. Connectivity index in neutrosophic trees and the algorithm to find its maximum spanning tree, volume 36. Infinite Study, 2020.
71. Sathyanarayanan Gopalakrishnan, Supriya Sridharan, Soumya Ranjan Nayak, Janmenjoy Nayak, and Swaminathan Venkataraman. Central hubs prediction for bio networks by directed hypergraph-ga with validation to covid-19 ppi. *Pattern Recognition Letters*, 153:246–253, 2022.
72. Georg Gottlob, Gianluigi Greco, Francesco Scarcello, et al. Treewidth and hypertree width. *Tractability: Practical Approaches to Hard Problems*, 1:20, 2014.
73. Georg Gottlob, Nicola Leone, and Francesco Scarcello. Hypertree decompositions and tractable queries. In *Proceedings of the eighteenth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 21–32, 1999.
74. Georg Gottlob, Nicola Leone, and Francesco Scarcello. Hypertree decompositions: A survey. In Mathematical Foundations of Computer Science 2001: 26th International Symposium, MFCS 2001 Mariánské Lázne, Czech Republic, August 27–31, 2001 Proceedings 26, pages 37–57. Springer, 2001.
75. Jonathan L Gross, Jay Yellen, and Mark Anderson. *Graph theory and its applications*. Chapman and Hall/CRC, 2018.
76. Muhammad Gulistan, Naveed Yaqoob, Zunaira Rashid, Florentin Smarandache, and Hafiz Abdul Wahab. A study on neutrosophic cubic graphs with real life applications in industries. *Symmetry*, 10(6):203, 2018.
77. Tom Gur, Noam Lifshitz, and Siqi Liu. Hypercontractivity on high dimensional expanders. In *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing*, pages 176–184, 2022.
78. Frank Gurski and Carolin Rehs. Comparing linear width parameters for directed graphs. *Theory of Computing Systems*, 63:1358–1387, 2019.
79. Jens Gustedt, Ole A Mæhle, and Jan Arne Telle. The treewidth of java programs. In *Workshop on Algorithm Engineering and Experimentation*, pages 86–97. Springer, 2002.
80. Mohammad Hamidi and Arsham Borumand Saeid. Achievable single-valued neutrosophic graphs in wireless sensor networks. *New Mathematics and Natural Computation*, 14(02):157–185, 2018.
81. Mohammad Hamidi and Florentin Smarandache. Derivable single valued neutrosophic graphs based on km-fuzzy metric. *IEEE Access*, 8:131076–131087, 2020.
82. Mohammad Hamidi, Florentin Smarandache, and Elham Davneshvar. Spectrum of superhypergraphs via flows. *Journal of Mathematics*, 2022(1):9158912, 2022.
83. Mohammad Hamidi, Florentin Smarandache, and Mohadeseh Taghinezhad. *Decision Making Based on Valued Fuzzy Superhypergraphs*. Infinite Study, 2023.
84. Mohammad Hamidi and Mohadeseh Taghinezhad. Application of Superhypergraphs-Based Domination Number in Real World. Infinite Study, 2023.
85. Pinar Heggernes, Daniel Lokshtanov, Rodica Mihai, and Charis Papadopoulos. Cutwidth of split graphs and threshold graphs. *SIAM Journal on Discrete Mathematics*, 25(3):1418–1437, 2011.
86. Dug Hun Hong and Chul Kim. A note on similarity measures between vague sets and between elements. *Information sciences*, 115(1-4):83–96, 1999.
87. Jing Huang and Jie Yang. Unignn: a unified framework for graph and hypergraph neural networks. *arXiv preprint arXiv:2105.00956*, 2021.
88. Liangsong Huang, Yu Hu, Yuxia Li, PK Kishore Kumar, Dipak Koley, and Arindam Dey. A study of regular and irregular neutrosophic graphs with real life applications. *Mathematics*, 7(6):551, 2019.
89. S Satham Hussain, R Jahir Hussain, and Florentin Smarandache. *On neutrosophic vague graphs*. Infinite Study, 2019.
90. Sang il Oum and Paul Seymour. Approximating clique-width and branch-width. *Journal of Combinatorial Theory, Series B*, 96(4):514–528, 2006.

91. Remie Janssen, Mark Jones, Steven Kelk, Georgios Stamoulis, and Taoyang Wu. Treewidth of display graphs: bounds, brambles and applications. *arXiv preprint arXiv:1809.00907*, 2018.
92. Cengiz Kahraman. Fuzzy multi-criteria decision making: theory and applications with recent developments, volume 16. Springer Science & Business Media, 2008.
93. Mamadou Moustapha Kanté and Michaël Rao. -rank-width of (edge-colored) graphs. In *International Conference on Algebraic Informatics*, pages 158–173. Springer, 2011.
94. Haim Kaplan and Ron Shamir. Pathwidth, bandwidth, and completion problems to proper interval graphs with small cliques. *SIAM Journal on Computing*, 25(3):540–561, 1996.
95. Faruk Karaaslan and Bijan Davvaz. Properties of single-valued neutrosophic graphs. *Journal of Intelligent & Fuzzy Systems*, 34(1):57–79, 2018.
96. M Kaviyarasu, Hossein Rashmanlou, Saeed Kosari, Said Broumi, R Venitha, M Rajeshwa, and Farshid Mofidnakhaei. Circular economy strategies to promote sustainable development using t-neutrosophic fuzzy graph. *Neutrosophic Sets and Systems*, 72:186–210, 2024.
97. Murugan Kaviyarasu, Luminița-Ioana Cotîrlă, Daniel Breaz, Murugesan Rajeshwari, and Eleonora Rapeanu. A study on complex t-neutrosophic graph with intention to preserve biodiversity. *Symmetry*, 16(8):1033, 2024.
98. Ephraim Korach and Nir Solel. Tree-width, path-width, and cutwidth. *Discrete Applied Mathematics*, 43(1):97–101, 1993.
99. Choongbum Lee, Joonkyung Lee, and Sang-il Oum. Rank-width of random graphs. *Journal of Graph Theory*, 70(3):339–347, 2012.
100. Xiaowei Liao, Yong Xu, and Haibin Ling. Hypergraph neural networks for hypergraph matching. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1266–1275, 2021.
101. Puyin Liu and Hong-Xing Li. *Fuzzy neural network theory and application*, volume 59. World Scientific, 2004.
102. Anam Luqman, Muhammad Akram, and Florentin Smarandache. Complex neutrosophic hypergraphs: new social network models. *Algorithms*, 12(11):234, 2019.
103. Dániel Marx. Approximating fractional hypertree width. *ACM Transactions on Algorithms (TALG)*, 6(2):1–17, 2010.
104. Rogers Mathew, Ilan Newman, Yuri Rabinovich, and Deepak Rajendraprasad. Boundaries of hypertrees, and hamiltonian cycles in simplicial complexes. *arXiv preprint arXiv:1507.04471*, 2015.
105. Zoltán Miklós. Understanding Tractable Decompositions for Constraint Satisfaction. PhD thesis, University of Oxford, 2008.
106. Dmitriy Molodtsov. Soft set theory-first results. *Computers & mathematics with applications*, 37(4-5):19–31, 1999.
107. John N Mordeson and Peng Chang-Shyh. Operations on fuzzy graphs. *Information sciences*, 79(3-4):159–170, 1994.
108. John N Mordeson and Sunil Mathew. *Advanced topics in fuzzy graph theory*, volume 375. Springer, 2019.
109. John N Mordeson and Premchand S Nair. Applications of fuzzy graphs. In *Fuzzy graphs and fuzzy hypergraphs*, pages 83–133. Springer, 2000.
110. John N Mordeson and Premchand S Nair. *Fuzzy graphs and fuzzy hypergraphs*, volume 46. Physica, 2012.
111. Susana Munoz, M Teresa Ortuno, Javier Ramírez, and Javier Yanez. Coloring fuzzy graphs. *Omega*, 33(3):211–221, 2005.
112. J Muthuerulappan and S Chelliah. A study of the neutrosophic dominating path-coloring number and multivalued star chromatic number in neutrosophic graphs. *Machine Intelligence Research*, 18(1):585–606, 2024.
113. Sumera Naz and Muhammad Aslam Malik. Single-valued neutrosophic line graphs. *TWMS Journal of Applied and Engineering Mathematics*, 8(2):483–494, 2018.
114. Jaroslav Nešetřil and Patrice Ossona De Mendez. Tree-depth, subgraph coloring and homomorphism bounds. *European Journal of Combinatorics*, 27(6):1022–1041, 2006.
115. Dan Olteanu and Jakub Závodnỳ. Factorised representations of query results: size bounds and readability. In *Proceedings of the 15th International Conference on Database Theory*, pages 285–298, 2012.

116. Sang-il Oum. Approximating rank-width and clique-width quickly. *ACM Transactions on Algorithms (TALG)*, 5(1):1–20, 2008.
117. Sang-il Oum. Rank-width is less than or equal to branch-width. *Journal of Graph Theory*, 57(3):239–244, 2008.
118. Sang-il Oum. Rank-width: Algorithmic and structural results. *Discrete Applied Mathematics*, 231:15–24, 2017.
119. Zdzisław Pawlak. Rough sets. International journal of computer & information sciences, 11:341–356, 1982.
120. Zdzislaw Pawlak, Jerzy Grzymala-Busse, Roman Slowinski, and Wojciech Ziarko. Rough sets. *Communications of the ACM*, 38(11):88–95, 1995.
121. Zdzislaw Pawlak, Lech Polkowski, and Andrzej Skowron. Rough set theory. *KI*, 15(3):38–39, 2001.
122. Zdzislaw Pawlak, S. K. Michael Wong, Wojciech Ziarko, et al. Rough sets: probabilistic versus deterministic approach. *International Journal of Man-Machine Studies*, 29(1):81–95, 1988.
123. Nicole Pearcy, Jonathan J Crofts, and Nadia Chuzhanova. Hypergraph models of metabolism. *International Journal of Biological, Veterinary, Agricultural and Food Engineering*, 8(8):752–756, 2014.
124. Andrzej Proskurowski and Jan Arne Telle. From bandwidth to pathwidth k. *_THEORETICAL-E*, page 90, 1996.
125. Hossein Rashmanlou and Madhumangal Pal. Balanced interval-valued fuzzy graphs. 2013.
126. Neil Robertson and Paul D. Seymour. Graph minors. i. excluding a forest. *Journal of Combinatorial Theory, Series B*, 35(1):39–61, 1983.
127. Neil Robertson and Paul D. Seymour. Graph minors. iii. planar tree-width. *Journal of Combinatorial Theory, Series B*, 36(1):49–64, 1984.
128. Neil Robertson and Paul D. Seymour. Graph minors. x. obstructions to tree-decomposition. *Journal of Combinatorial Theory, Series B*, 52(2):153–190, 1991.
129. Azriel Rosenfeld. Fuzzy graphs. In Fuzzy sets and their applications to cognitive and decision processes, pages 77–95. Elsevier, 1975.
130. Rıdvan Şahin. An approach to neutrosophic graph theory with applications. *Soft Computing*, 23(2):569–581, 2019.
131. Sovan Samanta, Madhumangal Pal, Hossein Rashmanlou, and Rajab Ali Borzooei. Vague graphs and strengths. *Journal of Intelligent & Fuzzy Systems*, 30(6):3675–3680, 2016.
132. Robert Sasak. Comparing 17 graph parameters. Master's thesis, The University of Bergen, 2010.
133. Florentin Smarandache. A unifying field in logics: Neutrosophic logic. In *Philosophy*, pages 1–141. American Research Press, 1999.
134. Florentin Smarandache. Neutrosophic set-a generalization of the intuitionistic fuzzy set. *International journal of pure and applied mathematics*, 24(3):287, 2005.
135. Florentin Smarandache. A unifying field in logics: neutrosophic logic. Neutrosophy, neutrosophic set, neutrosophic probability: neutrsophic logic. Neutrosophy, neutrosophic set, neutrosophic probability. Infinite Study, 2005.
136. Florentin Smarandache. Plithogenic set, an extension of crisp, fuzzy, intuitionistic fuzzy, and neutrosophic sets-revisited. Infinite study, 2018.
137. Florentin Smarandache. n-superhypergraph and plithogenic n-superhypergraph. *Nidus Idearum*, 7:107–113, 2019.
138. Florentin Smarandache. Extension of HyperGraph to n-SuperHyperGraph and to Plithogenic n-SuperHyperGraph, and Extension of HyperAlgebra to n-ary (Classical-/Neutro-/Anti-) HyperAlgebra. Infi- nite Study, 2020.
139. Florentin Smarandache. Introduction to SuperHyperAlgebra and Neutrosophic SuperHyperAlgebra. Infinite Study, 2022.
140. Florentin Smarandache. Decision making based on valued fuzzy superhypergraphs. 2023.
141. Florentin Smarandache. Foundation of the superhypersoft set and the fuzzy extension superhypersoft set: A new vision. *Neutrosophic Systems with Applications*, 11:48–51, 2023.
142. Florentin Smarandache. New types of soft sets "hypersoft set, indetermsoft set, indetermhypersoft set, and treesoft set": an improved version. Infinite Study, 2023.
143. Florentin Smarandache. Foundation of superhyperstructure & neutrosophic superhyperstructure. *Neutrosophic Sets and Systems*, 63(1):21, 2024.

144. Florentin Smarandache. *Nidus Idearum. Scilogs, XIV: SuperHyperAlgebra*. Infinite Study, 2024.
145. Florentin Smarandache and Said Broumi. *Neutrosophic graph theory and algorithms*. IGI Global, 2019.
146. Florentin Smarandache and NM Gallup. Generalization of the intuitionistic fuzzy set to the neutrosophic set. In *International Conference on Granular Computing*, pages 8–42. Citeseer, 2006.
147. Florentin Smarandache, WB Kandasamy, and K Ilanthenral. Applications of bimatrices to some fuzzy and neutrosophic models. 2005.
148. Florentin Smarandache, WB Kandasamy, and K Ilanthenral. Neutrosophic graphs: A new dimension to graph theory. 2015.
149. Florentin Smarandache and Nivetha Martin. Plithogenic n-super hypergraph in novel multi-attribute deci- sion making. Infinite Study, 2020.
150. Metawee Songsaeng and Aiyared Iampan. *Neutrosophic set theory applied to UP-algebras*. Infinite Study, 2019.
151. P Srivastava and RL Gupta. Fuzzy proximity structures and fuzzy ultrafilters. *Journal of Mathematical Analysis and Applications*, 94(2):297–311, 1983.
152. Atsushi Takahashi, Shuichi Ueno, and Yoji Kajitani. Minimal acyclic forbidden minors for the family of graphs with bounded path-width. *Discrete Mathematics*, 127(1-3):293–304, 1994.
153. Atsushi Takahashi, Shuichi Ueno, and Yoji Kajitani. Mixed searching and proper-path-width. *Theoretical Computer Science*, 137(2):253–268, 1995.
154. Atsushi TAKAHASHI, Shuichi UENO, and Yoji KAJITANI. On the proper-path-decomposition of trees. *IEICE transactions on fundamentals of electronics, communications and computer sciences*, 78(1):131–136, 1995.
155. CB ten Brinke. Variations on boolean-width. Master's thesis, 2015.
156. Duc Long Tran. *Expanding the Graph Parameter Hierarchy*. PhD thesis, Institute of Software, 2022.
157. Danyang Wang and Ping Zhu. Fuzzy rough digraph based on strength of connectedness with application. *Neural Computing and Applications*, 35(16):11847–11866, 2023.
158. Douglas Brent West et al. *Introduction to graph theory*, volume 2. Prentice hall Upper Saddle River, 2001.
159. Marcin Wrochna. Reconfiguration in bounded bandwidth and tree-depth. *Journal of Computer and System Sciences*, 93:1–10, 2018.
160. Lehua Yang, Dongmei Li, and Ruipu Tan. Research on the shortest path solution method of interval valued neutrosophic graphs based on the ant colony algorithm. *IEEE Access*, 8:88717–88728, 2020.
161. Nikola Yolov. Minor-matching hypertree width. In Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, pages 219–233. SIAM, 2018.
162. Lotfi A Zadeh. Fuzzy sets. *Information and control*, 8(3):338–353, 1965.
163. Lotfi A Zadeh. A fuzzy-set-theoretic interpretation of linguistic hedges. 1972.
164. Lotfi A Zadeh. Fuzzy sets versus probability. *Proceedings of the IEEE*, 68(3):421–421, 1980.
165. Lotfi A Zadeh. Fuzzy logic, neural networks, and soft computing. In *Fuzzy sets, fuzzy logic, and fuzzy systems: selected papers by Lotfi A Zadeh*, pages 775–782. World Scientific, 1996.
166. Lotfi Asker Zadeh. Fuzzy sets as a basis for a theory of possibility. *Fuzzy sets and systems*, 1(1):3–28, 1978.
167. Qian-Sheng Zhang and Sheng-Yi Jiang. A note on information entropy measures for vague sets and its applications. *Information Sciences*, 178(21):4184–4191, 2008.

Received: 01 Jul 2024, **Revised:** 15 Dec 2024,

Accepted: 03 Jan 2025, **Available online:** 05 Jan 2025.