NSS

Neutrosophic Sets and Systems, Vol. 86, 2025

University of New Mexico

Short note of SuperHyperClique-width and Local Superhypertree-width

Takaaki Fujita[1]∗ and Talal Ali Al-Hawary[2]

[1]∗Independece Researcher; Shinjuku, Shinjuku-ku, Tokyo, Japan; Takaaki.fujita060@gmail.com

[2] Department of Mathematics, Yarmouk University, Irbid, Jordan; talalhawary@yahoo.com

Abstract. Tree-width is a fundamental parameter that quantifies how ”tree-like” an undirected graph is, based on its optimal tree decomposition [32]. Several related concepts, including Hypertree-width [20], Branch-width [30], Linear-width [5], Local Tree-width [22], and SuperHypertree-width [13], have been extensively studied. Clique-width, on the other hand, is defined as the minimum number of labels required to construct a graph using four operations: vertex creation, disjoint union, edge insertion, and relabeling [24]. Local tree-width is a function mapping radius r to the maximum tree-width among all r-neighborhood induced subgraphs in a graph. A hypergraph is a generalization of a graph where each edge can connect any number of vertices, not just two. The concept of a SuperHyperGraph generalizes the classical notion of a hypergraph by introducing recursive hierarchical structures.

In this paper, we introduce new graph parameters: HyperClique-width, SuperHyperClique-width, Local Hypertree-width, and Local SuperHypertree-width. We formally define these parameters and provide an initial mathematical exploration of their structural properties.

Keywords: Hypergraph, Superhypergraph, Tree-width, Clique-width, Local Tree-width

---

1. **Introduction**

1.1. *Graph Width Parameters*

Graphs have been extensively studied in recent years [9, 26], with particular emphasis on structural parameters that capture how “tree-like” or “path-like” a graph is. Understanding these parameters is essential for designing efficient algorithms in areas such as network optimization, database theory, and machine learning [20].

Key width parameters include:

- **Tree-width** [31]: measures the minimum size of the largest bag minus one in a tree decomposition, quantifying how closely a graph resembles a tree.
- **Path-width** [25]: similar to tree-width but restricted to path-like decompositions.
- **Clique-width** [28]: the minimum number of labels needed to construct the graph using vertex creation, disjoint union, edge insertion between labels, and relabeling.
- **Local Tree-width** [8,22]: the function $r \mapsto \max_{v \in V} \operatorname{tw}\big(G[N_r(v)]\big)$, where $N_r(v)$ is the set of vertices at distance at most r from v, capturing the maximum tree-width of all r-neighborhoods.
- Additional parameters such as cut-width [23], rank-width [29], boolean-width [1], and bandwidth [7] have also been widely investigated.

These width parameters often correspond to tractable structural decompositions, driving much of the research on parameterized and approximation algorithms.

1.2. *Hypergraph and SuperHyperGraph*

A *hypergraph* generalizes a graph by allowing each edge (hyperedge) to connect any number of vertices [4]. Hypertree-width [19] and hyperpath-width [27] extend tree- and path-width to hypergraphs, with important applications in database theory and constraint satisfaction.

More recently, the *SuperHyperGraph* has been introduced [34] as a further generalization. In a SuperHyperGraph, each *supervertex* may be an element, a subset, or a nested subset up to n levels, and each *superedge* connects groups of supervertices at possibly different hierarchical levels. This framework is well suited to modeling hierarchical structures, multi-level networks, and layered relationships in real-world systems.

1.3. *Our Contribution*

Although several width parameters have been defined for graphs and hypergraphs, the study of width parameters in SuperHyperGraphs remains in its infancy [13]. In this paper, we introduce and investigate four new parameters:

- **HyperClique-width**: an extension of clique-width to r-uniform hypergraphs, measuring the minimum number of labels needed to construct the hypergraph using hyperedge-insertion operations.
- **SuperHyperClique-width**: a further generalization of HyperClique-width to n-SuperHyperGraphs, capturing the label complexity required to build hierarchical supervertices and superedges.
- **Local Hypertree-width**: the function $r \mapsto \max_{v \in V} \operatorname{htw}\big(H[N_H^r(v)]\big)$, which records the maximum hypertree-width among all r-neighborhood subhypergraphs of H.

- **Local**

  SuperHypertree-width: the analogous local parameter for n-SuperHyperGraphs, defined by $r \mapsto \max_{v \in V} \text{n-SHTW}\big(S[N_S^r(v)]\big)$.

We hope that the introduction of the above parameters will promote further research into the mathematical structure and applications of graph width parameters, stimulate deeper exploration of hypergraphs and superhypergraphs, and encourage the study of relationships between these new parameters and other established graph invariants.

1.4. *Structure of This Paper*

This paper is organized as follows:

- **Section 2** provides definitions and examples of hypergraphs, SuperHyperGraphs, tree-width, hypertree-width, and SuperHypertree-width.
- **Section 3** introduces our new parameters: HyperClique-width, SuperHyperClique-width, Local Hypertree-width, and Local SuperHypertree-width.
- **Section 4** concludes the paper and discusses directions for future work.

2. **Preliminaries and Definitions**

This section presents a structured overview of the fundamental concepts and definitions essential for understanding the main results of this paper. Throughout this paper, we consider only concepts that are undirected, finite, and simple.

2.1. *Graph and Hypergraph*

A hypergraph extends the concept of a graph by allowing edges, known as hyperedges, to connect multiple vertices rather than just pairs. This generalization provides a more flexible framework for modeling complex relationships [4, 20]. The fundamental structures of graphs and hypergraphs are formally defined below.

Definition 2.1 (Complete Graph)**.** A *complete graph* K_n is a graph with n vertices where every pair of distinct vertices is connected by an edge. Formally,

$$E(K_n) = \{\{u, v\} \mid u, v \in V(K_n), u \neq v\}.$$

Definition 2.2 (Hypergraph [4,6])**.** A *hypergraph* $H = (V(H), E(H))$ is a pair where:

- $V(H)$: A non-empty set of vertices.
- $E(H)$: A set of hyperedges, each of which is a subset of $V(H)$.

This paper focuses exclusively on finite hypergraphs.

Definition 2.3 (Induced Subhypergraph [6])**.** Let $H = (V(H), E(H))$ be a hypergraph, and let $X \subseteq V(H)$ be a subset of its vertices. The *subhypergraph induced by* X is defined as:

$$H[X] = \big(X, \{e \cap X \mid e \in E(H)\}\big).$$

The hypergraph obtained by removing X from H is denoted by:

$$H \setminus X := H[V(H) \setminus X].$$

Definition 2.4 (Separation in a Hypergraph)**.** Let $H = (V(H), E(H))$ be a hypergraph. A *separation* of H is a pair of subhypergraphs (A, B) that satisfies the following:

- $A = H[V_A]$ and $B = H[V_B]$, where $V_A, V_B \subseteq V(H)$ are subsets of the vertex set $V(H)$.
- $V_A \cup V_B = V(H)$, ensuring that A and B collectively cover all vertices in H.
- $V_A \cap V_B$, referred to as the *separator*, ensures $E(A) \cap E(B) = \emptyset$, meaning no hyperedge in H is shared between A and B.

The *order* of the separation (A, B) is defined as the size of the separator:

$$|V_A \cap V_B|.$$

2.2. *n-SuperHyperGraph*

The concept of a SuperHyperGraph generalizes the classical notion of a hypergraph by introducing recursive hierarchical structures. It has been recently formalized and studied extensively in the literature [15, 16, 34]. Below, we present its formal definition.

Definition 2.5 (n-th Powerset)**.** (cf. [17, 35])

The n-th powerset of a set H, denoted $P_n(H)$, is defined iteratively as follows:

$$P_1(H) = P(H), \quad P_{n+1}(H) = P(P_n(H)), \quad \text{for } n \geq 1,$$

where $P(H)$ is the standard powerset of H. Similarly, the n-th non-empty powerset, $P_n^*(H)$, is defined recursively as:

$$P_1^*(H) = P^*(H), \quad P_{n+1}^*(H) = P^*(P_n^*(H)),$$

with $P^*(H)$ representing the powerset of H excluding the empty set.

Definition 2.6 (n-SuperHyperGraph)**.** (cf. [16, 33]) Let V_0 be a finite base set of vertices. The n-th iterated powerset of V_0 is defined recursively as:

$$\mathcal{P}^0(V_0) = V_0, \quad \mathcal{P}^{k+1}(V_0) = \mathcal{P}\left(\mathcal{P}^k(V_0)\right),$$

where $\mathcal{P}(A)$ denotes the standard powerset of set A.

An *n-SuperHyperGraph* is an ordered pair $H = (V, E)$, where:

- $V \subseteq \mathcal{P}^n(V_0)$: The set of *supervertices*, which are elements of the n-th powerset of V_0.

- $E \subseteq \mathcal{P}^n(V_0)$: The set of *superedges*, which are also elements of $\mathcal{P}^n(V_0)$.

Each supervertex $v \in V$ can be one of the following:

- A single vertex ($v \in V_0$).
- A subset of V_0 ($v \subseteq V_0$).
- A nested subset up to n levels ($v \in \mathcal{P}^n(V_0)$).
- An indeterminate or fuzzy set.
- The null set ($v = \emptyset$).

Each superedge $e \in E$ connects supervertices, which may reside at different hierarchical levels up to n.

Example 2.7 (A 2-SuperHyperGraph)**.** Let the base set be

$$V_0 = \{a, b\}.$$

Then

$$\mathcal{P}^1(V_0) = \{\emptyset, \{a\}, \{b\}, \{a, b\}\},$$
$$\mathcal{P}^2(V_0) = \mathcal{P}(\mathcal{P}^1(V_0)),$$

the powerset of these four subsets. We choose the set of supervertices

$$V = \{\{a\},\ \{b\},\ \{\{a\}, \{b\}\},\ \{\{a, b\}\}\} \ \subseteq\ \mathcal{P}^2(V_0),$$

where for instance $\{\{a\}, \{b\}\}$ is a nested subset of level 2. Let the superedges be

$$E = \{\, e_1, e_2, e_3\},$$

with

$$e_1 = \{\{a\},\, \{\{a\}, \{b\}\}\}, \quad e_2 = \{\{b\},\, \{\{a\}, \{b\}\}\}, \quad e_3 = \{\{a, b\},\, \{\{a, b\}\}\}.$$

Each superedge $e_i \in E$ connects two supervertices possibly at different hierarchical levels.

Definition 2.8 (n-SuperHypertree)**.** (cf. [13, 18]) An *n-SuperHypertree (n-SHT)* is an n-SuperHyperGraph $\mathrm{SHT}_n = (V, E)$ that satisfies the following properties:

(1) *Host Tree Condition*: There exists a tree $T = (V_T, E_T)$, called the *host tree*, such that:
   - The vertex set of T is $V_T = V$, where $V \subseteq \mathcal{P}^n(V_0)$.
   - Each n-superedge $e \in E$ corresponds to a connected subtree $T_e \subseteq T$. Specifically, for each $e \in E$, there exists a subtree T_e such that:

$$\bigcup_{t \in V(T_e)} B_t \supseteq e,$$

   where $B_t \subseteq V$ are subsets associated with the nodes of T.

(2) *Acyclicity Condition*: The host tree T must be acyclic, ensuring that SHT_n inherits the acyclic structure of T.

(3) *Connectedness Condition*: For any two n-supervertices $v, w \in V$, there must exist a sequence of n-superedges $e_1, e_2, \ldots, e_k \in E$ such that:
   (a) $v \in e_1$ and $w \in e_k$.
   (b) $e_i \cap e_{i+1} \neq \emptyset$ for all $1 \leq i < k$.

Example 2.9 (A 2-SuperHypertree)**.** Using the same V and E as above, we construct a host tree

$$T = (V_T, E_T), \qquad V_T = V,$$

with edges

$$E_T = \{ (\{a\}, \{\{a\}, \{b\}\}), (\{b\}, \{\{a\}, \{b\}\}), (\{a, b\}, \{\{a, b\}\}) \}.$$

For each node $t \in V_T$ we assign the bag $B_t = \{t\}$. Then:

- For $e_1 = \{\{a\}, \{\{a\}, \{b\}\}\}$, the subtree T_{e_1} is the path $\{a\}$–$\{\{a\}, \{b\}\}$, and $\bigcup_{t \in V(T_{e_1})} B_t = \{ \{a\}, \{\{a\}, \{b\}\}\} = e_1$.
- Similarly for e_2 and e_3, each e_i is covered by the connected subtree on its two endpoints.
- T is acyclic by construction, and any two supervertices are joined by a path in T, so the connectedness condition holds.

Hence (V, E) together with T and the bags B_t form a 2-SuperHypertree.

2.3. *n-SuperHypertree-width*

Tree-width quantifies how closely a graph resembles a tree by considering the minimum width of its optimal tree decomposition, where the width is defined as the size of the largest bag minus one [14,32]. Hypertree-width extends tree-width to hypergraphs by minimizing the maximum bag size in a hypertree decomposition while ensuring connectivity and full hyperedge coverage [20]. The concept of SuperHypertree-width has been widely studied in the literature, including [13]. Below, we formally define Tree-width, Hypertree-width, and SuperHypertree-width.

Definition 2.10 (Tree-width)**.** [32] Let $G = (V, E)$ be a graph. A *tree-decomposition* of G is a pair $(T, \{X_t \mid t \in V(T)\})$, where:

- $T = (V(T), E(T))$ is a tree,
- $X_t \subseteq V$ for each $t \in V(T)$ (called *bags*),

such that:

(1) $\bigcup_{t \in V(T)} X_t = V$, i.e., every vertex of G appears in at least one bag.
(2) For every edge $\{u, v\} \in E$, there exists $t \in V(T)$ such that $u, v \in X_t$, ensuring edge coverage.

(3) For all $t_1, t_2, t_3 \in V(T)$, if t_2 lies on the path between t_1 and t_3 in T, then $X_{t_1} \cap X_{t_3} \subseteq X_{t_2}$, ensuring connectivity.

The *width* of a tree-decomposition is defined as:

$$\text{width}(T, \{X_t\}) = \max_{t \in V(T)} \big(|X_t| - 1\big),$$

where $|X_t|$ is the number of vertices in the bag X_t. The *tree-width* of G, denoted $\text{tw}(G)$, is the minimum width over all possible tree-decompositions of G:

$$\text{tw}(G) = \min_{(T, \{X_t\})} \text{width}(T, \{X_t\}).$$

Definition 2.11 (Hypertree-width)**.** [20,21] Let $H = (V(H), E(H))$ be a hypergraph, where $V(H)$ is the set of vertices and $E(H)$ is the set of hyperedges. A *tree decomposition* of H is a tuple $(T, (B_t)_{t \in V(T)})$, where:

- $T = (V(T), F(T))$ is a tree.
- $(B_t)_{t \in V(T)}$ is a family of subsets of $V(H)$, called *bags*, such that:
  (1) For every hyperedge $e \in E(H)$, there exists a node $t \in V(T)$ such that $e \subseteq B_t$.
  (2) For every vertex $v \in V(H)$, the set $\{t \in V(T) \mid v \in B_t\}$ induces a connected subtree of T.

The *width* of a tree decomposition $(T, (B_t)_{t \in V(T)})$ is defined as:

$$\text{width}(T, (B_t)_{t \in V(T)}) = \max_{t \in V(T)} (|B_t| - 1).$$

The *hypertree-width* of H, denoted by $\text{tw}(H)$, is the minimum width over all possible tree decompositions of H.

Example 2.12 (Hypertree-width of a Simple Hypergraph)**.** Consider the hypergraph

$$H = (V, E), \quad V = \{v_1, v_2, v_3, v_4\}, \quad E = \{e_1, e_2\},$$

where

$$e_1 = \{v_1, v_2, v_3\}, \qquad e_2 = \{v_2, v_3, v_4\}.$$

We construct a tree decomposition $(T, \{B_t\}_{t \in V(T)})$ of H as follows:

- The host tree T has two nodes t_1 and t_2 joined by an edge:

$$t_1 \; - \; t_2.$$

- The bags are

$$B_{t_1} = \{v_1, v_2, v_3\}, \qquad B_{t_2} = \{v_2, v_3, v_4\}.$$

We verify the decomposition conditions:

(1) *Coverage of hyperedges:* $e_1 = \{v_1, v_2, v_3\} \subseteq B_{t_1}$ and $e_2 = \{v_2, v_3, v_4\} \subseteq B_{t_2}$.

(2) *Connectedness of vertex appearances:*

- v_1 appears only in B_{t_1}, so its index set $\{t_1\}$ is connected.
- v_4 appears only in B_{t_2}, so $\{t_2\}$ is connected.
- v_2 and v_3 each appear in both B_{t_1} and B_{t_2}, and $\{t_1, t_2\}$ is connected in T.

The width of this decomposition is

$$\max\{|B_{t_1}| - 1,\ |B_{t_2}| - 1\} = \max\{3 - 1,\ 3 - 1\} = 2.$$

Hence the *hypertree-width* of H is

$$\mathrm{htw}(H) = 2.$$

Definition 2.13 (n-SuperHypertree-width)**.** (cf. [12, 13]) Let $H = (V, E)$ be an n-SuperHyperGraph, where:

- $V \subseteq \mathcal{P}^n(V_0)$ is the set of n-supervertices.
- $E \subseteq \mathcal{P}^n(V_0)$ is the set of n-superedges.

An n-SuperHypertree decomposition of H is a tuple $(T, \mathcal{B}, \mathcal{C})$, where:

- $T = (V_T, E_T)$ is a tree.
- $\mathcal{B} = \{B_t \mid t \in V_T\}$ is a collection of subsets of V (called *bags*), satisfying:
  (1) For every n-superedge $e \in E$, there exists a node $t \in V_T$ such that $e \subseteq B_t$.
  (2) For every n-supervertex $v \in V$, the set $\{t \in V_T \mid v \in B_t\}$ induces a connected subtree of T.
- $\mathcal{C} = \{C_t \mid t \in V_T\}$ is a collection of subsets of E (called *guards*), satisfying:
  (1) For every node $t \in V_T$, $B_t \subseteq \bigcup C_t$, where:
  $$\bigcup C_t = \{v \in V \mid \exists e \in C_t \text{ such that } v \in e\}.$$
  (2) For every node $t \in V_T$, the following holds:
  $$(\bigcup C_t) \cap \bigcup_{u \in T_t} B_u \subseteq B_t,$$
  where T_t is the subtree of T rooted at t.

The *width* of an n-SuperHypertree decomposition $(T, \mathcal{B}, \mathcal{C})$ is defined as:

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

The *n-SuperHypertree-width* of H, denoted n-SHT-width(H), is the minimum width over all possible n-SuperHypertree decompositions:

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

Example 2.14 (2-SuperHypertree-width of a 2-SuperHyperGraph)**.** Let $V_0 = \{a, b\}$ and $n = 2$. Then

$$\mathcal{P}^1(V_0) = \{\{a\}, \{b\}, \{a, b\}, \emptyset\}, \qquad \mathcal{P}^2(V_0) = \mathcal{P}\big(\mathcal{P}^1(V_0)\big).$$

Consider the 2-SuperHyperGraph $H = (V, E)$ with

$$V = \big\{\{a\},\ \{b\},\ \{\{a\},\{b\}\},\ \{\{a,b\}\}\big\}, \quad E = \{e_1, e_2, e_3\},$$

where

$$e_1 = \{\{a\},\ \{\{a\},\{b\}\}\}, \quad e_2 = \{\{b\},\ \{\{a\},\{b\}\}\}, \quad e_3 = \{\{a,b\},\ \{\{a,b\}\}\}.$$

A valid 2-SuperHypertree decomposition $(T, \mathcal{B}, \mathcal{C})$ is given by:

$$T: \quad t_0 \ - \ t_3,$$

$$B_{t_0} = \{\{a\}, \{b\}, \{\{a\}, \{b\}\}\}, \quad C_{t_0} = \{e_1, e_2\},$$
$$B_{t_3} = \{\{a,b\}, \{\{a,b\}\}\}, \quad C_{t_3} = \{e_3\}.$$

- $e_1, e_2 \subseteq B_{t_0}$ and $e_3 \subseteq B_{t_3}$.
- For each supervertex $v \in V$, the set $\{\, t \mid v \in B_t\}$ induces a connected subtree of T.
- Each bag B_t is covered by its guard: $\bigcup C_{t_0} = e_1 \cup e_2 \supseteq B_{t_0}$, $\bigcup C_{t_3} = e_3 \supseteq B_{t_3}$.

Since $\max\{|C_{t_0}|, |C_{t_3}|\} = 2$, the width of this decomposition is 2. Therefore

$$\text{2-SHT-width}(H) = 2.$$

Example 2.15 (3-SuperHypertree-width of a 3-SuperHyperGraph)**.** Let the base set be

$$V_0 = \{a, b\},$$

so that $\mathcal{P}^1(V_0) = \{\emptyset, \{a\}, \{b\}, \{a, b\}\}$, $\mathcal{P}^2(V_0) = \mathcal{P}(\mathcal{P}^1(V_0))$, and $\mathcal{P}^3(V_0) = \mathcal{P}(\mathcal{P}^2(V_0))$. Choose four 3-supervertices in $\mathcal{P}^3(V_0)$:

$$X = \big\{\{\{a\}\}\big\}, \quad Y = \big\{\{\{b\}\}\big\}, \quad Z = \big\{\{\{a\},\{b\}\}\big\}, \quad U = \big\{\{\{a,b\}\}\big\}.$$

Let

$$V = \{X, Y, Z, U\}, \qquad E = \big\{\, e_1, e_2\big\},$$

where

$$e_1 = \{X, Y, Z\}, \qquad e_2 = \{Y, Z, U\}.$$

Then $H = (V, E)$ is a 3-SuperHyperGraph. We now give a 3-SuperHypertree decomposition $(T, \mathcal{B}, \mathcal{C})$ of H of width 2.

- The host tree T is the path $t_1 - t_2 - t_3$.
- Bags:
$$B_{t_1} = \{X, Y, Z\}, \quad B_{t_2} = \{Y, Z, U\}, \quad B_{t_3} = \{Y, Z, U\}.$$
- Guards:
$$C_{t_1} = \{\, e_1\}, \quad C_{t_2} = \{\, e_1, e_2\}, \quad C_{t_3} = \{\, e_2\}.$$

Verification:

(1) Each superedge is covered: $e_1 \subseteq B_{t_1}, B_{t_2}$ and $e_2 \subseteq B_{t_2}, B_{t_3}$.

(2) For each supervertex $v \in \{X, Y, Z, U\}$, the indices $\{t_i \mid v \in B_{t_i}\}$ form a connected subtree of T.

(3) Each bag B_{t_i} is covered by its guard: $\bigcup C_{t_1} = e_1 \supseteq B_{t_1}$, $\bigcup C_{t_2} = e_1 \cup e_2 \supseteq B_{t_2}$, $\bigcup C_{t_3} = e_2 \supseteq B_{t_3}$.

Since $\max\{|C_{t_1}|, |C_{t_2}|, |C_{t_3}|\} = 2$, the width of this decomposition is 2. Hence

$$\text{3-SHT-width}(H) = 2.$$

2.4. *Clique-width*

Clique-width is the minimum number of labels required to construct a graph using four operations: vertex creation, disjoint union, edge insertion, and relabeling [21].

Definition 2.16 (Clique-width)**.** [11] Let k be a positive integer. A *k-graph* is a graph whose vertices are labeled with integers from $\{1, 2, \ldots, k\}$. The *clique-width* of a graph G, denoted by $cwd(G)$, is the smallest integer k such that G can be constructed using the following four operations:

(1) **Creation:** For any label $i \in \{1, \ldots, k\}$, create a new vertex with label i, denoted by $i(v)$.

(2) **Disjoint Union:** If G_1 and G_2 are k-graphs, then their disjoint union $G_1 \oplus G_2$ is also a k-graph.

(3) **Edge Insertion:** For any two distinct labels i and j, the operation $\eta_{i,j}$ adds an edge between every vertex labeled i and every vertex labeled j.

(4) **Relabeling:** For any labels $i, j \in \{1, \ldots, k\}$, the operation $\rho_{i\to j}$ changes every vertex with label i to label j.

A *k-expression* is an algebraic term built from these operations that constructs the graph G. Hence, $cwd(G)$ is the minimum number k for which there exists a k-expression that defines G.

Example 2.17 (Clique-width of a Complete Graph)**.** Consider the complete graph K_4 with vertex set $\{u, v, w, x\}$. One possible 2-expression constructing K_4 is:

$$\rho_{2\to1}\Big(\eta_{1,2}\Big(\rho_{2\to1}\Big(\eta_{1,2}\Big(\rho_{2\to1}\Big(\eta_{1,2}\big(2(u) \oplus 1(v)\big)\Big) \oplus 2(w)\Big)\Big) \oplus 2(x)\Big)\Big).$$

This expression shows that K_4 can be constructed using only 2 labels, so $cwd(K_4) \le 2$. In fact, every complete graph on at least two vertices has clique-width exactly 2.

2.5. *Local Tree-width*

Local tree-width measures the maximum tree-width of an r-neighborhood in a graph, capturing localized structural complexity within distance r [10].

Definition 2.18 (Local Tree-width)**.** [8, 22] Let $G = (V, E)$ be a graph. For each vertex $v \in V$ and each $r \in \mathbb{N}$, define the *r-neighborhood* of v by

$$N_G^r(v) = \{w \in V \mid d_G(v, w) \leq r\},$$

and let $G[N_G^r(v)]$ be the subgraph of G induced by $N_G^r(v)$. Then the *local tree-width* of G is the function

$$\mathrm{ltw}_G : \mathbb{N} \to \mathbb{N}, \quad \mathrm{ltw}_G(r) = \max_{v \in V} \mathrm{tw}\big(G[N_G^r(v)]\big),$$

where $\mathrm{tw}(H)$ denotes the tree-width of a graph H.

Example 2.19 (Local Tree-width of a Tree)**.** Let T be a tree. Since every tree has tree-width 1 and every induced subgraph of a tree is a forest (hence of tree-width at most 1), it follows that

$$\mathrm{ltw}_T(r) = 1 \quad \text{for all } r \in \mathbb{N}.$$

3. **Results**

This section presents the main results of this paper.

3.1. *HyperClique-width*

HyperClique-width measures the complexity of constructing an r-uniform hypergraph using a fixed number of labels and operations such as creation, union, relabeling, and hyperedge insertion.

Definition 3.1 (HyperClique-width)**.** Let $r \geq 2$ be an integer and let $H = (V, E)$ be an r-uniform hypergraph (i.e., every hyperedge in E contains exactly r vertices). A *k-hypergraph expression* is defined by extending the clique-width operations to hypergraphs as follows. Each vertex carries a label from $\{1, 2, \ldots, k\}$ and the allowed operations are:

(1) **Creation:** For any label $i \in \{1, \ldots, k\}$, create an isolated vertex with label i, denoted by $i(v)$.
(2) **Disjoint Union:** If H_1 and H_2 are labeled hypergraphs, then their disjoint union $H_1 \oplus H_2$ is also a labeled hypergraph.
(3) **Relabeling:** For any labels $i, j \in \{1, \ldots, k\}$, the operation $\rho_{i \to j}$ changes every vertex with label i to label j.
(4) **Hyperedge Insertion:** For any r-tuple of *distinct* labels $(i_1, i_2, \ldots, i_r)$, the operation

$$\eta_{i_1, i_2, \ldots, i_r}$$

adds, for every r-tuple of vertices $(v_1, v_2, \ldots, v_r)$ such that v_j has label i_j (for $1 \leq j \leq r$), the hyperedge $\{v_1, v_2, \ldots, v_r\}$ to the hypergraph. (Existing hyperedges are not duplicated.)

The *hyperClique-width* of the r-uniform hypergraph H, denoted by $\mathrm{hcwd}(H)$, is the smallest integer k such that there exists a k-hypergraph expression that constructs H.

Example 3.2 (HyperClique-width of a 3-uniform Hypergraph)**.** Consider the 3-uniform hypergraph H with vertex set

$$V = \{a, b, c, d\},$$

and hyperedge set

$$E = \big\{\{a, b, c\},\ \{a, b, d\}\big\}.$$

We show that H can be constructed using 3 labels, so that $\mathrm{hcwd}(H) \leq 3$.

A possible 3-hypergraph expression for H is as follows:

(1) **Creation:** Create vertices with initial labels:

$$1(a), \quad 2(b), \quad 3(c).$$

(2) **Union and Hyperedge Insertion:** Form the disjoint union

$$H_1 = 1(a) \oplus 2(b) \oplus 3(c),$$

and then apply $\eta_{1,2,3}$ to insert the hyperedge $\{a, b, c\}$.

(3) **Add an Additional Vertex:** Create the vertex

$$1(d).$$

Form the new hypergraph

$$H_2 = H_1 \oplus 1(d).$$

(4) **Relabeling and Hyperedge Insertion:** In order to insert the hyperedge $\{a, b, d\}$, we require the labels on a, b, and d to be 1, 2, and 3, respectively. Currently, a is labeled 1, b is labeled 2, and d is labeled 1. Hence, apply the relabeling operation $\rho_{1\to 3}$ to change the label of d from 1 to 3. Finally, apply $\eta_{1,2,3}$ to insert the hyperedge $\{a, b, d\}$.

This construction produces the hypergraph H with the desired hyperedges, showing that $\mathrm{hcwd}(H) \leq 3$. (In this framework, one may also show that at least 3 labels are necessary.)

Theorem 3.3 (HyperClique-width of Complete Hypergraphs)**.** *Let K_n^r be the complete r-uniform hypergraph on $n \geq r$ vertices. Then*

$$hcwd(K_n^r) = r.$$

Proof. **(Lower bound)**: Observe that in any hypergraph expression constructing K_n^r, every hyperedge insertion operation $\eta_{i_1,i_2,\ldots,i_r}$ requires an r-tuple of *distinct* labels. Since K_n^r contains at least one hyperedge (in fact, every r-subset of vertices is a hyperedge), the construction must at some point produce an r-tuple of vertices with pairwise distinct labels. Hence, at least r labels are required, so

$$\text{hcwd}(K_n^r) \geq r.$$

(Upper bound): We now show that r labels suffice to construct K_n^r by induction on n.

Base case: For $n = r$, create r vertices with distinct labels $1, 2, \ldots, r$ using the creation operation. Then, applying the hyperedge insertion operation $\eta_{1,2,\ldots,r}$ inserts the single hyperedge consisting of all r vertices. Thus, K_r^r is constructed with r labels.

Inductive step: Assume that for some $n \geq r$ the complete r-uniform hypergraph K_n^r can be constructed with r labels. We now construct K_{n+1}^r.

(1) **Vertex Addition:** Create a new vertex v with an arbitrary label, say label 1.
(2) **Hyperedge Insertions:** For each $(r-1)$-subset S of the n existing vertices, consider the r-subset $S \cup \{v\}$. Since K_n^r was constructed using r labels, by appropriate (temporary) relabeling of the vertices in S if necessary, we can ensure that the vertices in $S \cup \{v\}$ receive r distinct labels. Then, apply $\eta_{1,2,\ldots,r}$ to insert the hyperedge corresponding to $S \cup \{v\}$.
(3) **Idempotence:** Note that inserting a hyperedge more than once does not affect the hypergraph.

By processing all $(r-1)$-subsets of the original vertex set, we insert every hyperedge containing v. Since the previous hyperedges of K_n^r remain intact, we have constructed K_{n+1}^r using only r labels. Therefore,

$$\text{hcwd}(K_{n+1}^r) \leq r.$$

Combining the lower and upper bounds, we conclude that

$$\text{hcwd}(K_n^r) = r.$$

□

Theorem 3.4 (Disjoint Union). *Let H_1 and H_2 be two r-uniform hypergraphs such that*

$$hcwd(H_1) \leq k \quad and \quad hcwd(H_2) \leq k.$$

Then the disjoint union $H_1 \oplus H_2$ satisfies

$$hcwd(H_1 \oplus H_2) \leq k.$$

Proof. A k-hypergraph expression for H_1 uses at most k labels, and similarly for H_2. The disjoint union operation $\oplus$ simply takes the union of the two expressions without requiring any new labels. Hence, a k-expression for $H_1 \oplus H_2$ is obtained by taking the disjoint union of the expressions for H_1 and H_2. Therefore,

$$\mathrm{hcwd}(H_1 \oplus H_2) \leq k.$$

□

Theorem 3.5 (Monotonicity)**.** *If H' is a subhypergraph of an r-uniform hypergraph H, then*

$$hcwd(H') \leq hcwd(H).$$

Proof. Let H' be a subhypergraph of H. Suppose that H is constructed by a k-hypergraph expression using $\mathrm{hcwd}(H) = k$ labels. Then, by simply restricting the construction (i.e., removing those vertices and hyperedges not in H'), we obtain a valid k-expression for H'. Hence, it follows that

$$\mathrm{hcwd}(H') \leq k = \mathrm{hcwd}(H).$$

□

Theorem 3.6 (Idempotence)**.** *Let H be an r-uniform hypergraph and let $\eta_{i_1,i_2,\ldots,i_r}$ be a hyperedge insertion operation. Then applying $\eta_{i_1,i_2,\ldots,i_r}$ repeatedly (on vertices with appropriate labels) does not change H after the first insertion; that is, hyperedge insertion is idempotent.*

Proof. By definition, the hyperedge insertion operation $\eta_{i_1,i_2,\ldots,i_r}$ adds a hyperedge among every r-tuple of vertices that currently have labels $i_1, i_2, \ldots, i_r$, respectively. Once the hyperedge $e = \{v_1, v_2, \ldots, v_r\}$ is inserted, any further application of $\eta_{i_1,i_2,\ldots,i_r}$ will attempt to insert the same hyperedge e again. Since the hyperedge set E of a hypergraph is a set (or a multiset where repeated insertions do not change the outcome), repeated insertion does not alter E. Hence, the operation is idempotent. □

3.2. *SuperhyperClique-width*

SuperhyperClique-width measures the complexity of constructing an n-SuperHyperGraph using a fixed number of labels and operations like creation, union, relabeling, and superedge insertion.

Definition 3.7 (SuperhyperClique-width)**.** Let $H = (V, E)$ be an n-SuperHyperGraph, where

$$V \subseteq \mathcal{P}^n(V_0)$$

for some finite base set V_0, and

$$E \subseteq \mathcal{P}^n(V_0).$$

An *SHC-expression* (SuperhyperClique-width expression) is an algebraic term that constructs H using a fixed set of labels from $\{1, 2, \ldots, k\}$ and the following operations:

(1) **Creation:** For any label $i \in \{1, \ldots, k\}$, create a new n-supervertex with label i, denoted by $i(v)$. (Here, $v \in \mathcal{P}^n(V_0)$ is chosen as the new supervertex.)
(2) **Disjoint Union:** If H_1 and H_2 are n-SuperHyperGraphs constructed by SHC-expressions, then their disjoint union, denoted by $H_1 \oplus H_2$, is also an n-SuperHyperGraph.
(3) **Superedge Insertion:** For any tuple of distinct labels $(i_1, i_2, \ldots, i_r)$ (with r chosen according to the uniformity of the superhypergraph), the operation

$$\eta_{i_1,i_2,\ldots,i_r}$$

adds, for every r-tuple of n-supervertices $(v_1, v_2, \ldots, v_r)$ such that for every $j \in \{1, \ldots, r\}$ the vertex v_j is currently labeled i_j, a superedge connecting them; that is, it inserts the superedge

$$\{v_1, v_2, \ldots, v_r\}.$$

(If an edge already exists, it is not duplicated.)
(4) **Relabeling:** For any labels $i, j \in \{1, \ldots, k\}$, the operation

$$\rho_{i\to j}$$

changes every supervertex with label i to have label j.

The *superhyperClique-width* of H, denoted by $\text{shcwd}(H)$, is the minimum integer k for which there exists an SHC-expression using at most k distinct labels that constructs H.

Example 3.8 (SuperhyperClique-width of a Simple 1-SuperHyperGraph)**.** Let $V_0 = \{a, b\}$ and choose $n = 1$, so that $\mathcal{P}^1(V_0) = \mathcal{P}(V_0)$. Consider the 1-SuperHyperGraph

$$H = (V, E),$$

with

$$V = \{\,\{a\},\, \{b\},\, \{a, b\}\},$$

and superedges

$$E = \{e_1, e_2\},$$

where we interpret e_1 as connecting $\{a\}$ to $\{a, b\}$ and e_2 as connecting $\{b\}$ to $\{a, b\}$.

A possible SHC-expression constructing H using only 2 labels is as follows:

(1) **Creation:** Create the following supervertices:

$$1(\{a\}), \quad 2(\{b\}), \quad 1(\{a, b\}).$$

(2) **Disjoint Union:** Form the disjoint union

$$H_1 = 1(\{a\}) \oplus 2(\{b\}) \oplus 1(\{a, b\}).$$

(3) **Superedge Insertion:** To insert the superedge corresponding to e_1, apply the operation $\eta_{1,2}$ to insert a superedge connecting the vertex with label 1 (i.e. $\{a\}$) and the vertex with label 2 (temporarily, we consider $\{a, b\}$ as having label 1; see next step). Next, to "prepare" for the second edge, relabel the vertex $\{a, b\}$ from 1 to 2 using $\rho_{1\to 2}$, so that it now carries label 2. Then apply $\eta_{1,2}$ again to insert the superedge corresponding to e_2, which connects the vertex with label 1 (i.e., $\{a\}$ or $\{b\}$, as required) and the vertex with label 2.

This construction shows that $\mathrm{shcwd}(H) \leq 2$, and one may further argue that at least 2 labels are necessary.

Example 3.9 (2-SuperhyperClique-width of a 2-SuperHyperGraph)**.** Let the base set be

$$V_0 = \{a, b\},$$

so that

$$\mathcal{P}^1(V_0) = \{\emptyset, \{a\}, \{b\}, \{a, b\}\}, \quad \mathcal{P}^2(V_0) = \mathcal{P}\big(\mathcal{P}^1(V_0)\big).$$

Define four supervertices in $\mathcal{P}^2(V_0)$ by

$$X = \{\{a\}\}, \quad Y = \{\{b\}\}, \quad Z = \{\{a\}, \{b\}\}, \quad W = \{\{a, b\}\},$$

and let

$$V = \{X, Y, Z, W\}.$$

Choose three superedges connecting these supervertices:

$$e_1 = \{X, Z\}, \quad e_2 = \{Y, Z\}, \quad e_3 = \{Z, W\},$$

so that $E = \{e_1, e_2, e_3\}$ and $H = (V, E)$ is a 2-SuperHyperGraph.

We will construct H using an SHC-expression with just two labels, showing $\mathrm{shcwd}(H) = 2$. Denote by $i(v)$ the creation of supervertex v with label i, and by $\eta_{1,2}$ the superedge-insertion between every label-1 and label-2 vertex. Then:

$$H \;=\; \eta_{1,2}\Big(\,1(X) \;\oplus\; 1(Y) \;\oplus\; 2(Z) \;\oplus\; 1(W)\Big).$$

- **Creation:** $1(X) \oplus 1(Y) \oplus 2(Z) \oplus 1(W)$ creates the four supervertices with labels in $\{1, 2\}$.
- **Superedge Insertion:** Applying $\eta_{1,2}$ inserts every superedge between each label-1 and each label-2 vertex, namely $\{1(X), 2(Z)\} = e_1$, $\{1(Y), 2(Z)\} = e_2$, and $\{1(W), 2(Z)\} = e_3$.

No relabeling is needed, and two labels suffice because a single label cannot distinguish both ends of these superedges. Hence

$$\mathrm{shcwd}(H) \;=\; 2.$$

Theorem 3.10 (Generalization Theorem)**.** *Let H be an r-uniform hypergraph. Consider H as a 0-SuperHyperGraph (i.e. with $n = 0$, so that $\mathcal{P}^0(V_0) = V_0$). Then the hyperClique-width of H, denoted by hcwd(H), is equal to the superhyperClique-width of H, i.e.,*

$$hcwd(H) = shcwd(H).$$

Proof. When $n = 0$, the construction model for an n-SuperHyperGraph coincides with that of a standard hypergraph. In this case, every vertex of H is simply an element of V_0, and the operations used in a k-hypergraph expression (creation, disjoint union, hyperedge insertion, and relabeling) are identical to those defined for SHC-expressions. Thus, any expression that constructs H as a hypergraph (and so demonstrates $\mathrm{hcwd}(H) \leq k$) is also a valid SHC-expression constructing H. Conversely, any SHC-expression in the 0-super setting is a hypergraph expression. Therefore,

$$\mathrm{hcwd}(H) = \mathrm{shcwd}(H).$$

□

Theorem 3.11 (Extension Theorem)**.** *Let H be an n-SuperHyperGraph with $n \geq 1$, and let H' be the hypergraph obtained by "flattening" the hierarchical structure of H (i.e., by considering only the base elements in V_0). Then there exists a constant $c(n)$, depending only on n, such that*

$$shcwd(H) \leq hcwd(H') + c(n).$$

Proof. We prove the theorem by induction on n.

Base Case ($n = 1$): A 1-SuperHyperGraph has vertices belonging to $\mathcal{P}(V_0)$ (the usual powerset). One can "flatten" such a hypergraph by mapping each supervertex (a subset of V_0) to an element in a hypergraph H' whose vertices are these subsets. The operations in a hyperClique-width expression for H' can be simulated in the 1-super setting using an additional fixed number of labels to encode the extra structure. Hence, there exists a constant $c(1)$ such that

$$\mathrm{shcwd}(H) \leq \mathrm{hcwd}(H') + c(1).$$

Inductive Step: Assume the statement holds for all m-SuperHyperGraphs with $m < n$. Let H be an n-SuperHyperGraph. Its vertices lie in $\mathcal{P}^n(V_0)$ and possess an inherent hierarchical structure with n levels. One may "flatten" one level to obtain an $(n-1)$-SuperHyperGraph H''. By the induction hypothesis,

$$\text{shcwd}(H'') \leq \text{hcwd}(H''') + c(n-1),$$

where H''' is the hypergraph obtained by fully flattening H''. To recover the original n-level structure, one must encode the additional hierarchical information. This encoding requires only a fixed number d of extra labels (depending solely on n), so that

$$\text{shcwd}(H) \leq \text{shcwd}(H'') + d.$$

Thus,

$$\text{shcwd}(H) \leq \text{hcwd}(H''') + c(n-1) + d.$$

Defining $c(n) = c(n-1) + d$ completes the inductive step. □

3.3. *Local Hypertree-width and Local n-SuperHypertree-width*

The definitions of Local Hypertree-width and Local n-SuperHypertree-width are provided below.

Definition 3.12 (Local Hypertree-width)**.** Let $H = (V, E)$ be a hypergraph and suppose that a notion of distance $d_H(v, w)$ is defined on H (for instance, via the length of the shortest hyperpath between v and w). For each $v \in V$ and $r \in \mathbb{N}$, define the *r-neighborhood* of v in H by

$$N_H^r(v) = \{w \in V \mid d_H(v, w) \leq r\},$$

and let $H[N_H^r(v)]$ denote the subhypergraph induced on $N_H^r(v)$. Then the *local hypertree-width* of H is the function

$$\text{lhtw}_H : \mathbb{N} \to \mathbb{N}, \quad \text{lhtw}_H(r) = \max_{v \in V} \text{htw}\big(H[N_H^r(v)]\big),$$

where htw$(\cdot)$ denotes the hypertree-width of a hypergraph.

Example 3.13 (Local Hypertree-width of an Acyclic Hypergraph)**.** If H is an acyclic hypergraph, then htw$(H) = 1$. Since every induced subhypergraph of an acyclic hypergraph is also acyclic, for every $v \in V$ and every $r \in \mathbb{N}$ we have

$$\text{htw}\big(H[N_H^r(v)]\big) = 1.$$

Thus,

$$\text{lhtw}_H(r) = 1 \quad \text{for all } r \in \mathbb{N}.$$

We can define the local n-superhypertree-width as follows.

Definition 3.14 (Local n-SuperHypertree-width)**.** Let $S = (V, E)$ be an n-SuperHyperGraph. For each $v \in V$ and $r \in \mathbb{N}$, define the *r-neighborhood* of v by

$$N_S^r(v) = \{w \in V \mid d_S(v, w) \leq r\},$$

and let $S[N_S^r(v)]$ denote the substructure of S induced on $N_S^r(v)$ (viewed as an n-SuperHyperGraph). Then the *local n-superhypertree-width* of S is the function

$$\mathrm{lshtw}_S : \mathbb{N} \to \mathbb{N}, \quad \mathrm{lshtw}_S(r) = \max_{v \in V} \text{n-SHTW}\big(S[N_S^r(v)]\big),$$

where n-SHTW(H) denotes the n-superhypertree-width of an n-SuperHyperGraph H.

Example 3.15 (Local 1-SuperHypertree-width of a Simple 1-SuperHyperGraph)**.** Consider a 1-SuperHyperGraph S with base set $V_0 = \{a, b\}$ so that

$$V = \{\, \{a\}, \{b\}, \{a, b\}\}.$$

Suppose that S has an overall 1-superhypertree-width 1-SHTW$(S) = 2$ and assume that every vertex is within distance 1 of every other vertex. Then

$$\mathrm{lshtw}_S(1) = \max_{v \in V} \text{1-SHTW}\big(S[N_S^1(v)]\big) = 2.$$

Example 3.16 (Local 2-SuperHypertree-width of a Simple 2-SuperHyperGraph)**.** Let $V_0 = \{a, b\}$ and $n = 2$. As before, set

$$V = \{\, X, Y, Z, W\}, \quad X = \{\{a\}\},\ Y = \{\{b\}\},\ Z = \{\{a\}, \{b\}\},\ W = \{\{a, b\}\},$$

and superedges

$$e_1 = \{X, Z\}, \quad e_2 = \{Y, Z\}, \quad e_3 = \{Z, W\}.$$

Then $S = (V, E)$ with $E = \{e_1, e_2, e_3\}$ is a 2-SuperHyperGraph of 2-SHT-width 2.

For radius $r = 1$, the 1-neighborhoods are

$$N_S^1(X) = \{X, Z\},\ N_S^1(Y) = \{Y, Z\},\ N_S^1(Z) = \{X, Y, Z, W\},\ N_S^1(W) = \{Z, W\}.$$

Consider the induced subgraphs $S_X = S[N_S^1(X)]$, S_Y, S_Z, and S_W. Each of S_X, S_Y, and S_W contains a single superedge (e_1, e_2, or e_3 respectively), so each has 2-SHT-width 1. Meanwhile S_Z is the whole graph S and thus has 2-SHT-width 2. Therefore

$$\mathrm{lshtw}_S(1) = \max\big\{\text{2-SHTW}(S_X),\ \text{2-SHTW}(S_Y),\ \text{2-SHTW}(S_Z),\ \text{2-SHTW}(S_W)\big\} = \max\{1, 1, 2, 1\} = 2.$$

Theorem 3.17 (Local Hypertree-width in Acyclic Hypergraphs)**.** *Let $H = (V, E)$ be an acyclic hypergraph, so that by definition* $\mathrm{htw}(H) = 1$*. Then for every radius $r \in \mathbb{N}$,*

$$\mathrm{lhtw}_H(r) \;=\; \max_{v \in V} \mathrm{htw}\big(H[N_H^r(v)]\big) \;=\; 1.$$

Proof. Recall that a hypergraph H is *acyclic* precisely when it admits a hypertree decomposition of width 1, and that any induced subhypergraph of an acyclic hypergraph remains acyclic.

Fix $r \geq 0$ and a vertex $v \in V$. By definition the r-neighborhood

$$N_H^r(v) = \{\, w \in V : \text{there is a hyperpath of length} \leq r \text{ from } v \text{ to } w\}$$

induces the subhypergraph $H[N_H^r(v)]$. Since H is acyclic, $H[N_H^r(v)]$ is also acyclic and thus admits a width-1 hypertree decomposition. Hence

$$\operatorname{htw}\bigl(H[N_H^r(v)]\bigr) = 1.$$

Taking the maximum over all $v \in V$ gives

$$\operatorname{lhtw}_H(r) = \max_{v \in V} \operatorname{htw}\bigl(H[N_H^r(v)]\bigr) = \max_{v \in V} 1 = 1,$$

as required. □

Theorem 3.18 (Local n-SuperHypertree-width in Structured n-SuperHyperGraphs)**.** *Let $S = (V, E)$ be an n-SuperHyperGraph whose global n-superhypertree-width satisfies*

$$\text{n-SHTW}(S) \;\leq\; k.$$

Then for every radius $r \in \mathbb{N}$,

$$\operatorname{lshtw}_S(r) \;=\; \max_{v \in V} \text{n-SHTW}\bigl(S[N_S^r(v)]\bigr) \;\leq\; k.$$

Proof. By assumption there exists an n-SuperHypertree decomposition

$$(T, \mathcal{B} = \{B_t\}, \mathcal{C} = \{C_t\})$$

of S of width at most k, meaning $|C_t| \leq k$ for every $t \in V_T$.

Now fix $v \in V$ and $r \geq 0$. Let

$$U \;=\; N_S^r(v) \quad \text{and} \quad S_U \;=\; S[U]$$

be the sub-n-SuperHyperGraph induced on the r-neighborhood of v. We claim that restricting the original decomposition yields a valid decomposition of S_U of width $\leq k$.

Define

$$T' \;=\; \bigl\{\, t \in V_T \mid B_t \cap U \neq \emptyset \bigr\},$$

which, by the connectedness property of the original decomposition, forms a connected subtree of T. For each $t \in T'$ set

$$B'_t \;=\; B_t \;\cap\; U, \qquad C'_t \;=\; \{\, e \in C_t : e \subseteq U\}.$$

One checks directly:

(1) Every superedge $e \in E$ with $e \subseteq U$ is still contained in some B'_t, since it lay in B_t originally and $t \in T'$.

(2) For each $u \in U$, the set $\{t \in T' \mid u \in B'_t\}$ remains connected in T' by restriction of the original connectivity.

(3) Each bag B'_t is covered by its guard: $\bigcup C'_t \supseteq B'_t$, since $B_t \subseteq \bigcup C_t$ and we only remove guards not entirely inside U.

(4) The running intersection property

$$\Big(\bigcup C'_t\Big) \;\cap\; \bigcup_{u \in T'_t} B'_u \;\subseteq\; B'_t$$

follows immediately from the corresponding property in the original decomposition.

Moreover, $|C'_t| \le |C_t| \le k$ for each $t \in T'$. Therefore $(T', \{B'_t\}, \{C'_t\})$ is an n-SuperHypertree decomposition of S_U of width $\le k$, showing

$$\text{n-SHTW}(S_U) \;\le\; k.$$

Taking the maximum over all $v \in V$ gives $\text{lshtw}_S(r) \le k$, as claimed. □

Theorem 3.19 (Monotonicity and Boundedness of Local Hypertree-width)**.** *Let $H = (V, E)$ be any hypergraph with hypertree-width* $\text{htw}(H) = w$*. Then for all* $0 \le r \le s$*,*

$$\text{lhtw}_H(r) \;\le\; \text{lhtw}_H(s) \;\le\; w.$$

Proof. Fix $v \in V$. Since $N^r_H(v) \subseteq N^s_H(v)$, the induced subhypergraphs satisfy

$$H[N^r_H(v)] \;\subseteq\; H[N^s_H(v)],$$

and hypertree-width is monotone under taking induced subhypergraphs, so

$$\text{htw}\big(H[N^r_H(v)]\big) \;\le\; \text{htw}\big(H[N^s_H(v)]\big) \;\le\; \text{htw}(H) \;=\; w.$$

Taking the maximum over $v \in V$ yields $\text{lhtw}_H(r) \le \text{lhtw}_H(s) \le w$, as claimed. □

Theorem 3.20 (Local Hypertree-width Stabilizes at the Global Radius)**.** *Let $H = (V, E)$ be a connected hypergraph of finite diameter D. Then for all $r \ge D$,*

$$\text{lhtw}_H(r) \;=\; \text{htw}(H).$$

Proof. Since the diameter D is the maximum distance between any two vertices, for every $v \in V$,

$$N^D_H(v) \;=\; V,$$

hence $H[N^D_H(v)] = H$. Thus $\text{htw}(H[N^D_H(v)]) = \text{htw}(H)$ for all v, and

$$\text{lhtw}_H(D) = \max_{v \in V} \text{htw}\big(H[N^D_H(v)]\big) = \text{htw}(H).$$

Monotonicity then gives $\mathrm{lhtw}_H(r) = \mathrm{htw}(H)$ for all $r \geq D$. □

Theorem 3.21 (Zero-Radius Local Width)**.** *For any hypergraph $H = (V, E)$ without loops,*

$$\mathrm{lhtw}_H(0) \; = \; 0.$$

Proof. By definition $N_H^0(v) = \{v\}$, so the induced subhypergraph $H[\{v\}]$ has only the single vertex v and no hyperedges of size ≥ 2. Any hypertree decomposition of a singleton vertex uses one bag of size 1, hence width $|X| - 1 = 0$. Therefore $\mathrm{htw}(H[\{v\}]) = 0$ for every v, and $\mathrm{lhtw}_H(0) = \max_v 0 = 0$. □

Theorem 3.22 (Monotonicity and Boundedness of Local n-SuperHypertree-width)**.** *Let $S = (V, E)$ be an n-SuperHyperGraph with global width*

$$\text{n-SHTW}(S) = k.$$

Then for all integers $0 \leq r \leq s$,

$$\mathrm{lshtw}_S(r) \; \leq \; \mathrm{lshtw}_S(s) \; \leq \; k.$$

Proof. Recall that

$$\mathrm{lshtw}_S(r) = \max_{v \in V} \text{ n-SHTW}\big(S[N_S^r(v)]\big),$$

where $N_S^r(v) = \{w : d_S(v, w) \leq r\}$ and $S[N_S^r(v)]$ is the induced subgraph on that neighborhood.

(1) Monotonicity: Fix any vertex $v \in V$. Since $r \leq s$, by definition of the distance,

$$N_S^r(v) \; \subseteq \; N_S^s(v).$$

Therefore the induced sub-n-SuperHyperGraph on $N_S^r(v)$ is a substructure of that on $N_S^s(v)$:

$$S[N_S^r(v)] \; \subseteq \; S[N_S^s(v)].$$

It is a standard fact that taking an induced substructure cannot increase the superhypertree-width. Hence

$$\text{n-SHTW}\big(S[N_S^r(v)]\big) \; \leq \; \text{n-SHTW}\big(S[N_S^s(v)]\big).$$

Taking the maximum over all $v \in V$ yields

$$\mathrm{lshtw}_S(r) = \max_v \text{n-SHTW}(S[N_S^r(v)]) \; \leq \; \max_v \text{n-SHTW}(S[N_S^s(v)]) = \mathrm{lshtw}_S(s).$$

(2) Boundedness by the Global Width: Again for each $v \in V$,

$$S[N_S^s(v)] \; \subseteq \; S,$$

so

$$\text{n-SHTW}\big(S[N_S^s(v)]\big) \;\le\; \text{n-SHTW}(S) \;=\; k.$$

Taking the maximum over v gives

$$\text{lshtw}_S(s) = \max_v \text{n-SHTW}(S[N_S^s(v)]) \;\le\; k.$$

Combining these two parts establishes the claimed inequalities. □

Theorem 3.23 (Stabilization of Local n-SuperHypertree-width). *Let $S = (V, E)$ be a connected n-SuperHyperGraph of finite diameter D, i.e.* $\max_{u,w\in V} d_S(u, w) = D$*. Then for every $r \ge D$,*

$$\text{lshtw}_S(r) \;=\; \text{n-SHTW}(S).$$

Proof. By definition of diameter, for each $v \in V$ and every $r \ge D$, we have

$$N_S^r(v) = \{w \in V : d_S(v, w) \le r\} = V,$$

so the induced substructure $S[N_S^r(v)]$ coincides with the entire S. Hence

$$\text{n-SHTW}\big(S[N_S^r(v)]\big) = \text{n-SHTW}(S)$$

for every v. Taking the maximum over $v \in V$ yields

$$\text{lshtw}_S(r) = \max_v \text{n-SHTW}(S[N_S^r(v)]) = \text{n-SHTW}(S).$$

□

4. Conclusion and Future Work

In this paper, we have introduced four new structural parameters for hypergraphs and superhypergraphs—HyperClique-width, SuperHyperClique-width, Local Hypertree-width, and Local SuperHypertree-width—providing formal definitions and an initial mathematical analysis of their properties.

Future research will deepen the theoretical study of these parameters and develop efficient algorithms to compute or approximate them. We also anticipate applications in diverse domains such as decision-making, machine learning, deep learning, chemistry, and network theory. Moreover, we plan to extend our investigation to width parameters in Crisp, Fuzzy, Intuitionistic Fuzzy, Soft, Hypersoft, Plithogenic, and Neutrosophic Graph (cf. [2, 3]).

Funding

This study was carried out without receiving any financial support or external funding from any organization or individual.

Acknowledgments

We sincerely thank all those who offered insights, inspiration, and assistance during the course of this research. Special appreciation is extended to our readers for their interest and to the authors of cited works, whose contributions laid the groundwork for this study. We also acknowledge the individuals and institutions that provided the resources and infrastructure needed to produce and disseminate this paper. Finally, we are grateful to everyone who supported us in various capacities throughout this project.

Ethical Approval

As this research is entirely theoretical in nature and does not involve human participants or animal subjects, no ethical approval is required.

Data Availability

This research is purely theoretical, involving no data collection or analysis. I encourage future researchers to pursue empirical investigations to further develop and validate the concepts introduced here.

Research Integrity

The authors hereby confirm that, to the best of their knowledge, this manuscript is their original work, has not been published in any other journal, and is not currently under consideration for publication elsewhere at this stage.

Conflicts of Interest

The authors confirm that there are no conflicts of interest related to this research or its publication.

Consent to Publish declaration

The author approved to Publish declarations.

Disclaimer (Note on Computational Tools)

No computer-assisted proof, symbolic computation, or automated theorem proving tools (e.g., Mathematica, SageMath, Coq, etc.) were used in the development or verification of the results presented in this paper. All proofs and derivations were carried out manually and analytically by the authors.

Disclaimer (Limitations and Claims)

The theoretical concepts presented in this paper have not yet been subject to practical implementation or empirical validation. Future researchers are invited to explore these ideas in applied or experimental settings. Although every effort has been made to ensure the accuracy of the content and the proper citation of sources, unintentional errors or omissions may persist. Readers should independently verify any referenced materials.

To the best of the authors' knowledge, all mathematical statements and proofs contained herein are correct and have been thoroughly vetted. Should you identify any potential errors or ambiguities, please feel free to contact the authors for clarification.

The results presented are valid only under the specific assumptions and conditions detailed in the manuscript. Extending these findings to broader mathematical structures may require additional research. The opinions and conclusions expressed in this work are those of the authors alone and do not necessarily reflect the official positions of their affiliated institutions.

References

1. 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.
2. Muhammad Akram, Danish Saleem, and Talal Al-Hawary. Spherical fuzzy graphs with application to decision-making. *Mathematical and Computational Applications*, 25(1):8, 2020.
3. TALAL ALI AL-HAWARY. On dombi fuzzy graphs. *Journal of applied mathematics & informatics*, 43(1):33–41, 2025.
4. Claude Berge. *Hypergraphs: combinatorics of finite sets*, volume 45. Elsevier, 1984.
5. Daniel Bienstock. Graph searching, path-width, tree-width and related problems (a survey). *Reliability of computer and communication networks*, 5:33–50, 1989.
6. Alain Bretto. Hypergraph theory. *An introduction. Mathematical Engineering. Cham: Springer*, 1, 2013.
7. 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.
8. Erik D Demaine, Fedor V Fomin, Mohammad Taghi Hajiaghayi, and Dimitrios M Thilikos. Bidimensional parameters and local treewidth. *SIAM Journal on Discrete Mathematics*, 18(3):501–511, 2004.
9. Reinhard Diestel. *Graph theory.* Springer (print edition); Reinhard Diestel (eBooks), 2024.
10. Andrea Ferrara, Guoqiang Pan, and Moshe Y Vardi. Treewidth in verification: Local vs. global. In *International Conference on Logic for Programming Artificial Intelligence and Reasoning*, pages 489–503. Springer, 2005.
11. Eldar Fischer, Johann A Makowsky, and Elena V Ravve. Counting truth assignments of formulas of bounded tree-width or clique-width. *Discrete Applied Mathematics*, 156(4):511–529, 2008.
12. Takaaki Fujita. Superhypertree-depth: A structural analysis within superhypergraphs. *Advancing Uncertain Combinatorics through Graphization, Hyperization, and Uncertainization: Fuzzy, Neutrosophic, Soft, Rough, and Beyond*, page 11.
13. Takaaki Fujita. Short note of supertree-width and n-superhypertree-width. *Neutrosophic Sets and Systems*, 77:54–78, 2024.

14. Takaaki Fujita. Improved version of short note: Exploring ideals in graph theory. *International Journal of Advanced Multidisciplinary Research and Studies*, 5(2):2043–2050, 2025.
15. Takaaki Fujita. A theoretical investigation of quantum n-superhypergraph states. *Neutrosophic Optimization and Intelligent Systems*, 6:15–25, 2025.
16. Takaaki Fujita and Florentin Smarandache. A concise study of some superhypergraph classes. *Neutrosophic Sets and Systems*, 77:548–593, 2024.
17. Takaaki Fujita and Florentin Smarandache. Superhypergraph neural networks and plithogenic graph neural networks: Theoretical foundations. *Advancing Uncertain Combinatorics through Graphization, Hyperization, and Uncertainization: Fuzzy, Neutrosophic, Soft, Rough, and Beyond*, 2025.
18. Masoud Ghods, Zahra Rostami, and Florentin Smarandache. Introduction to neutrosophic restricted superhypergraphs and neutrosophic restricted superhypertrees and several of their properties. *Neutrosophic Sets and Systems*, 50:480–487, 2022.
19. Georg Gottlob, Gianluigi Greco, Francesco Scarcello, et al. Treewidth and hypertree width. *Tractability: Practical Approaches to Hard Problems*, 1:20, 2014.
20. 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.
21. Georg Gottlob and Reinhard Pichler. Hypergraphs in model checking: Acyclicity and hypertree-width versus clique-width. *SIAM Journal on Computing*, 33(2):351–378, 2004.
22. Martin Grohe. Local tree-width, excluded minors, and approximation algorithms. *arXiv preprint math/0001128*, 2000.
23. 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.
24. Sang il Oum and Paul Seymour. Approximating clique-width and branch-width. *Journal of Combinatorial Theory, Series B*, 96(4):514–528, 2006.
25. 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.
26. TM Nishad, Talal Ali Al-Hawary, and B Mohamed Harif. General fuzzy graphs. *Ratio Mathematica*, 47, 2023.
27. 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.
28. Sang-il Oum. Approximating rank-width and clique-width quickly. *ACM Transactions on Algorithms (TALG)*, 5(1):1–20, 2008.
29. Sang-il Oum. Rank-width: Algorithmic and structural results. *Discrete Applied Mathematics*, 231:15–24, 2017.
30. Sang-il Oum and Paul Seymour. Testing branch-width. *Journal of Combinatorial Theory, Series B*, 97(3):385–393, 2007.
31. Neil Robertson and Paul D. Seymour. Graph minors. i. excluding a forest. *Journal of Combinatorial Theory, Series B*, 35(1):39–61, 1983.
32. Neil Robertson and Paul D. Seymour. Graph minors. ii. algorithmic aspects of tree-width. *Journal of algorithms*, 7(3):309–322, 1986.
33. Florentin Smarandache. n-superhypergraph and plithogenic n-superhypergraph. *Nidus Idearum*, 7:107–113, 2019.
34. Florentin Smarandache. *Extension of HyperGraph to n-SuperHyperGraph and to Plithogenic n-SuperHyperGraph, and Extension of HyperAlgebra to n-ary (Classical-/Neutro-/Anti-) HyperAlgebra*. Infinite Study, 2020.

35. Florentin Smarandache. Foundation of superhyperstructure & neutrosophic superhyperstructure. *Neutrosophic Sets and Systems*, 63(1):21, 2024.

Received: Nov. 10, 2024. Accepted: May 28, 2025